

Człowiek – najlepsza inwestycja

PRACA Z UCZNIEM UZDOLNIONYM INFORMATYCZNIE

Tło psychologiczno-pedagogiczne,
doświadczenia i propozycje zajęć

Praca zbiorowa pod redakcją Macieja M. Sysły

informatyka+

Ponadregionalny program rozwijania
kompetencji uczniów szkół
ponadgimnazjalnych w zakresie technologii
informacyjno-komunikacyjnych (ICT)



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

PRACA Z UCZNIEM UZDOLNIONYM INFORMATYCZNIE

**Tło psychologiczno-pedagogiczne,
doświadczenia i propozycje zajęć**

Praca zbiorowa pod redakcją
Macieja M. Sysły



Publikacja współfinansowana ze środków Unii Europejskiej
w ramach Europejskiego Funduszu Społecznego

Człowiek – najlepsza inwestycja

Praca zbiorowa pod redakcją: prof. dr hab. Macieja M. Sysły

Publikacja opracowana w ramach projektu edukacyjnego
Informatyka+ – ponadregionalny program rozwijania kompetencji
uczniów szkół ponadgimnazjalnych w zakresie technologii
informacyjno-komunikacyjnych (ICT)
www.informatykaplus.edu.pl
kontakt@informatykaplus.edu.pl



Wydawca: Warszawska Wyższa Szkoła Informatyki
ul. Lewartowskiego 17, 00-169 Warszawa
www.wysi.edu.pl
rektorat@wysi.edu.pl

Wydanie pierwsze

Warszawa 2011
Copyright © Warszawska Wyższa Szkoła Informatyki 2010

ISBN 978-83-921270-3-1

Publikacja nie jest przeznaczona do sprzedaży
Nakład: 1100 egz.

Łamanie, druk i oprawa:
PPGK SA Drukarnia KART
01-252 Warszawa, ul. Przyce 20



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

SPIS TREŚCI

Wstęp <i>Maciej M. Sysło</i>	5
Część I – Rozważania ogólne na temat pracy z uczniem uzdolnionym	11
1. <i>Hanna Kiedewicz-Nappi</i> , Wspomaganie rozwoju zdolnych uczniów – refleksje pedagoga szkolnego	13
2. <i>Katarzyna Śliwińska</i> , Uczeń zdolny w szkole	23
Część II – Doświadczenia w pracy z uczniem uzdolnionym informatycznie	35
1. <i>Krzysztof Diks, Jakub Radoszewski, Jakub Łącki, Marek Cygan, Tomasz Idziaszek</i> , Różne podejścia w pracy z uczniami uzdolnionymi informatycznie	37
2. <i>Krzysztof Diks, Jan Madey</i> , Od szkolnych konkursów programistycznych do sukcesów w zawodzie informatyka	45
3. <i>Anna Beata Kwiatkowska</i> , O rozwijaniu zdolności informatycznych uczniów – na podstawie doświadczeń w pracy z uczniami w Liceum Akademickim w Toruniu	53
4. <i>Grzegorz Grochowski</i> , Praca z młodzieżą uzdolnioną informatycznie. Doświadczenia Ryszarda Szubartowskiego i Stowarzyszenia Talent	61
Część III – Przykładowe scenariusze zajęć informatycznych	71
1. Moduł: Algorytmika i programowanie	
<i>Maciej M. Sysło</i> , Rekurencja – jakie to proste!	73
<i>Maciej M. Sysło</i> , Logarytm w informatyce – jego występowanie i znaczenie	83
<i>Błażej Osiński</i> , Gdzie można spotkać drzewa rozpinające	98
<i>Bolesław Kulbabiński</i> , Jak wygrywać w grach	107
2. Moduł: Bazy danych	
<i>Andrzej Ptasznik</i> , Integralność danych – reguły i ograniczenia	116
<i>Andrzej Ptasznik</i> , Wykorzystanie XML w relacyjnych bazach danych	123
<i>Andrzej Ptasznik</i> , Zapytania rekurencyjne z wykorzystaniem wyrażeń CTE w SQL Server	136
3. Moduł: Sieci komputerowe	
<i>Dariusz Chaładyniak</i> , Zarządzanie sieciami LAN	146
<i>Dariusz Chaładyniak</i> , Zarządzanie sieciami WAN	153
<i>Dariusz Chaładyniak</i> , Konfiguracja protokołów routingu statycznego i dynamicznego	159



4. Moduł: Multimedia

<i>Piotr Kopciał</i> , Obróbka i wizualizacja obrazów.....	165
<i>Piotr Kopciał</i> , Tworzenie serwisów internetowych	172
<i>Andrzej Majkowski</i> , Nagrywanie i obróbka filmów.....	178
Biogramy autorów	187



WSTĘP

Niniejsza publikacja powstała w ramach projektu Informatyka +, którego celem jest podniesienie kompetencji informatycznych wśród uczniów szkół ponadgimnazjalnych oraz przybliżenie im akademickich form kształcenia, takich jak wykład i warsztaty. Zajęcia w projekcie są adresowane głównie do uczniów uzdolnionych informatycznie w tym sensie, że przejawiają większe niż przeciętne zainteresowania i umiejętności informatyczne.

Projekt Informatyka + pojawił się w momencie, gdy w wielu krajach, w odpowiedzi na malejące zainteresowanie uczniów karierami w dziedzinach informatycznych i technicznych, powraca do łask wydzielony przedmiot informatyka, który w naszym kraju nigdy nie został usunięty ze szkół. Ten projekt jest zespołem działań, których celem jest *outreach*, czyli wyjście uczelni do szkół, by lepiej przygotować uczniów do przyszłych wyborów kariery w kierunkach informatycznych. Przyjęto przy tym, że uczniowie liceów z klas o profilu matematyczno-informatycznym lub uczniowie techników, zwłaszcza o profilu informatycznych, elektronicznym i elektrycznym, wybierali ten typ szkoły kierując się swoimi większymi zainteresowaniami i zdolnościami, są więc potencjalnymi kandydatami na studia informatyczne.

Książka jest zbiorem artykułów, składających się na trzy części. W pierwszej części są omówione aspekty pedagogiczne i psychologiczne pracy z uczniem uzdolnionym, nie tylko informatycznie. Część druga zawiera doświadczenia zebrane w wybranych ośrodkach, w których kształcą się uczniowie najbardziej uzdolnieni informatycznie, uczestnicy olimpiad i konkursów informatycznych. Natomiast w części trzeciej zawarto propozycje zajęć z uczniami przejawiającymi większe zainteresowania tematami informatycznymi, których zakres wykracza poza podstawę programową,

Książka ma charakter poradnika i jest adresowana do nauczycieli, którzy chętnie podejmą pracę z uczniami przejawiającymi większe zainteresowania tematami informatycznymi. W części pierwszej znajdują rozważania ogólne dotyczące pracy z uczniami zdolnymi, w drugiej – mogą zapoznać się z różnymi formami pracy z uczniami najbardziej uzdolnionymi informatycznie, a w części trzeciej – gotowe propozycje zajęć.

W dalszej części Wstępu, najpierw charakteryzujemy projekt Informatyka +, a następnie krótko omawiamy poszczególne części w tej książce.

1. Projekt Informatyka +

Projekt **Informatyka +**, w ramach którego powstała ta publikacja, jest prowadzony przez Warszawską Wyższą Szkołę Informatyki (WWSI) i w latach 2009-2012 weźmie w nim udział ponad 15 tysięcy uczniów ze szkół ponadgimnazjalnych z pięciu województw¹. Zajęcia odbywają się w WWSI i w pięciu Regionalnych Ośrodkach Projektu (ROP – w Garwolinie, Łodzi, Elblągu, Białymstoku i Lublinie) oraz w wybranych szkołach.

¹ Szczegółowe informacje o projekcie Informatyka + znajdują się na stronie <http://www.informatykaplus.edu.pl/>, a pod adresem <http://mmsyslo.pl/Nauczanie/WWSI> jest zamieszczone podsumowanie przebiegu projektu w pierwszym roku jego realizacji.



Celem tego projektu jest podwyższenie kluczowych kompetencji uczniów szkół ponadgimnazjalnych z zakresu informatyki i jej zastosowań, niezbędnych do dalszego kształcenia się na kierunkach informatycznych i technicznych lub podjęcia zatrudnienia, oraz stworzenie uczniom zdolnym innowacyjnych możliwości rozwijania zainteresowań w tym zakresie. Projekt ten jest formą kształcenia pozalekcyjnego. Zajęcia odbywają się na dwóch poziomach: podstawowym (**Wszechnica Informatyczna**) – dla wszystkich uczniów, bez specjalnego przygotowania informatycznego, i rozszerzonym (**Kuźnia Talentów Informatycznych**) – dla uczniów szczególnie uzdolnionych informatycznie.

Wszystkim zajęciom towarzyszą materiały drukowane, prezentacje, testy, w sumie dla 63 jednostek zajęć. Ukażą się dwa opracowania książkowe, które zostaną rozesłane do szkół: poradnik dla nauczycieli do pracy z uczniem uzdolnionym informatycznie – to opracowanie i „Homo Informaticus – Tendencje w rozwoju informatyki i jej zastosowań”, książka będąca nowoczesnym wprowadzeniem do współczesnej informatyki. Do materiałów tych mają również dostęp na platformie uczniowie i nauczyciele z innych regionów Polski.

Projektowi towarzyszy platforma edukacyjna Fronter, która jest nie tylko repozytorium materiałów dydaktycznych ale również jest wykorzystywana m.in. do przechowywania wyników pracy uczniów na zajęciach warsztatowych, do przeprowadzania testów oraz do zbierania opinii. Dzięki platformie wykładowcy, uczniowie i nauczyciele mają dostęp do materiałów z dowolnego miejsca, gdzie się znajdują, jeśli tylko mają dostęp do Internetu. Platforma Fronter to element chmury edukacyjnej – rozwiązania, które zdobywa coraz większą popularność również w edukacji.

Projekt Informatyka + obejmuje zajęcia w ramach 5 modułów tematycznych: Algorytmika i programowanie, Bazy danych, Grafika, multimedia i techniki Internetu, Sieci komputerowe oraz Tendencje w rozwoju informatyki i jej zastosowań. Ponadto, w ramach projektu są organizowane dodatkowe formy aktywizujące uczniów, takie jak: Algorytmiczna liga zadaniowa, Mistrzostwa WWSI w programowaniu, konkursy tematyczne (konkurs „Bóbr”, „Nasza szkoła w Internecie”, Konkurs grafiki komputerowej), letnie obozy naukowe, udział w kołach naukowych, działających w uczelniach.

Zajęcia w tym projekcie mają także na celu przybliżenie uczniom akademickich form kształcenia – wykład jako wprowadzenie do tematyki zajęć, często na wyższym poziomie i w szerszym zakresie a warsztaty są praktyczną częścią zajęć, bazującą na materiale przekazanym na wykładzie. Drukowane materiały odgrywają przy tym rolę skryptu do zajęć, w którym uczeń może znaleźć szeroki opis przedstawianych zagadnień, a prezentacje do wykładów mogą być przydatne przy powtórce materiału z wykładu.

Zakres i forma zajęć w projekcie Informatyka + służą przybliżeniu uczniom szkół ponadgimnazjalnych możliwych wyborów dróg dalszego kształcenia informatycznego w uczelniach wyższych, a tym samym wyboru przyszłej kariery w zawodach związanych z informatyką i jej zastosowaniami. Uzmysławiamy też uczniom, że poszerza się gama zawodów określanych mianem **IT Profession**, czyli zawodów związanych z profesjonalnym wykorzystywaniem zastosowań informatyki. Pracownicy tych zawodów albo są informatykami z wykształcenia, albo najczęściej nie kończyli studiów informatycznych, jednak muszą profesjonalnie posługiwać się narzędziami informatycznymi. Do IT Profession zalicza się np. specjalistów z zakresu bioinformatyki, informatyki medycznej, telekomunikacji, genetyki itp. – wszyscy oni muszą umieć „programować” swoje narzędzia informatyczne. Informatyki ich w tym nie wyręczy, gdyż nie potrafi. W Stanach Zjednoczonych, do IT Profession zalicza się obecnie ponad 40 zawodów, w których profesjonalnie są wykorzystywane zastosowania informatyki, i ta lista stale się powiększa.

Ten projekt pojawił się w momencie, gdy kształcenie informatyczne przeżywa na całym świecie trudności w pozyskiwaniu nowych studentów i kształceniu informatyków specjalistów, na których jest coraz większe oczekiwanie rynku pracy.

1.1. Regres kształcenia informatycznego i jego powody

Wydzielone zajęcia informatyczne w polskich szkołach były bardzo poważnie traktowane w kolejnych reformach systemu oświaty od drugiej połowy lat 90. XX wieku i nigdy pod żadnym naciskiem nie pozwolono usunąć ze szkół przedmiotu informatyka, chociaż taki przykład płynął z Atlantyku, gdzie od lat 90. XX wieku komputery w szkołach były wykorzystywane głównie do kształcenia umiejętności z zakresu technologii informacyjno-komunikacyjnej.

Obecnie w Stanach Zjednoczonych przywraca się kształcenie w zakresie informatyki, co ma powstrzymać spadek zainteresowania uczniów karierami informatycznymi, wynoszący w ostatnich latach aż 50% (podobnie jest w Wielkiej Brytanii). W tej sytuacji Narodowa Fundacja Nauki USA ogłosiła w 2009 roku „CS/10000 Project”, który ma objąć 10000 szkół i 10000 nauczycieli nowymi programami kształcenia informatycznego. Ponadto, w Senacie USA pojawiła się inicjatywa Computer Science Education Act of 2010, który, podkreślając znaczenie kształcenia informatycznego, ma na celu znalezienie środków na wsparcie działań instytucji stanowych w zakresie podniesienia poziomu kształcenia informatycznego w szkołach.

Powody tej zapaści są wielorakie. Z jednej strony, wiele osób, w tym nauczyciele i rodzice, nie uważa informatyki za niezależną dziedzinę nauki, jak i za szkolny przedmiot. Na ogół myślą oni i utożsamiają informatykę z technologią informacyjno-komunikacyjną i sprowadzają edukację informatyczną do udostępniania uczniom i nauczycielom komputerów i Internetu w szkole i w domu. Nie odróżniają oni stosowania komputerów i sieci Internet od studiowania podstaw informatyki.

Z drugiej zaś strony, jest wiele powodów zmniejszonego zainteresowania samych uczniów informatyką, jako dziedziną kształcenia i przyszłą karierą zawodową. Na początku informatyka była kojarzona z programowaniem komputerów, co wywoływało silny sprzeciw, gdyż niewielu uczniów chciało zostać kiedyś programistami. Na przełomie lat 80. i 90. XX wieku tylko nieliczni uczniowie używali komputerów w szkole lub w domu przed wstąpieniem na uczelnię. Na przełomie XX i XXI wieku główny nacisk w szkołach zmienił się diametralnie – kształcono z zakresu korzystania z aplikacji biurowych i Internetu. Obecnie wielu przyszłych studentów zdobywa pierwsze doświadczenia informatyczne przed wstąpieniem na uczelnię, najczęściej poza szkołą. Co więcej, dostępne oprogramowanie umożliwia tworzenie nawet bardzo złożonych aplikacji komputerowych bez wcześniejszego zaznajomienia się z: logiką, metodami programowania, matematyką dyskretną, które należą do kanonu kształcenia informatycznego. W rezultacie, absolwenci szkół średnich nieźle radzą sobie z wykorzystaniem komputerów do zabawy, poszukiwań w sieci i do komunikowania się, ale znikoma jest ich wiedza na temat informatyki jako dyscypliny oraz o tym, jak funkcjonuje komputer i sieć komputerowa. Dorastając, mają oni na tyle dość styczności z technologią informatyczną, że nie interesuje ich rozwijanie swoich umiejętności w tym zakresie na poziomie uczelni. Potrzebny jest więc sposób motywowania uczniów, aby zainteresowali się tym, co dzieje się poza ekranem komputera, jak zbudowany jest komputer i sieć oraz jak działa oprogramowanie, a w dalszej perspektywie tworzyli własne rozwiązania informatyczne.

Chcąc zmienić tę sytuację, zamiast utwierdzać uczniów w przekonaniu, że ich wiedza i umiejętności w zakresie informatyki są wystarczające, zajęcia w projekcie Informatyka + służą przygotowaniu zdolniejszych uczniów do dalszego kształcenia się w kierunkach związanych z informatyką,

1.2. Trudności w rozwoju kształcenia informatycznego

Wiele czynników powoduje, że przed rozwojem edukacji informatycznej w szkołach piętrzy się wiele trudności, często obiektywnych. Wymieńmy najpoważniejsze z nich:

1. Brak nauczycieli odpowiednio przygotowanych do realizacji zajęć informatycznych. Większość nauczycieli wydziałonych przedmiotów informatycznych ma przygotowanie informatyczne



wyniesione ze studiów podyplomowych, trwających 360 godzin. Niewielu absolwentów kierunków informatycznych podejmuje pracę w szkołach.

2. **Kształcenie informatyczne w szkołach nie nadąża za zmianami w zakresie i metodologii kształcenia informatycznego.** Na ogół sprowadza się ono do alfabetyzacji komputerowej, nauczyciele informatyki rzadko posługują się metodologią rozwiązywania problemów z pomocą komputerów. Do rzadkości należy realizacja algorytmiki w ramach przedmiotu informatyka w gimnazjum.
3. **Kształcenie informatyczne z rzadka odwołuje się do rzeczywistych zastosowań, z którymi uczniowie spotykają się na co dzień, na przykład w zakresie wykorzystania komputerów i Internetu w innych przedmiotach.**
4. **Obserwuje się wśród uczniów regres zainteresowania informatyką (o powodach piszemy powyżej) oraz kształceniem na kierunkach ścisłych i technicznych.**
5. **Większość nauczycieli przedmiotów informatycznych nie jest odpowiednio przygotowanych do pracy z uczniami uzdolnionymi.**
6. **Niewielka jest oferta dodatkowych, pozalekcyjnych zajęć informatycznych w szkołach, a także oferta zajęć pozaszkolnych, oferowanych przez nauczycieli ze szkół.**

Świadomość, że maleje zainteresowanie uczniów studiowaniem na kierunkach technicznych, ścisłych, i przyrodniczych, w tym również na informatyce, powoduje podejmowanie różnych działań zaradczych, takich jak projekt Informatyka +. W Stanach Zjednoczonych powstała nawet specjalna inicjatywa federalna pod nazwą **STEM** (Science, Technology, Engineering, Mathematics), w ramach której jest prowadzonych wiele działań, które mają na celu przynajmniej powrót do sytuacji sprzed 10 lat. Te działania sięgają aż do gimnazjum, by już na tym etapie przygotowywać uczniów do podejmowania w przyszłości właściwych wyborów swojej kariery zawodowej.

U nas w kraju, inicjatywy tego typu można podzielić na dwie grupy, w obu przypadkach są wspierane przez fundusze UE. Z jednej strony, MNiSzW wspiera kierunki deficytowe, które z kolei oferują studentom dość wysokie stypendia (ok. 1000 zł). z drugiej strony – MEN i organy samorządowe prowadzą projekty, których celem – pod hasłem: Człowiek – najlepsza inwestycja – jest wspieranie rozwoju wiedzy i umiejętności w dziedzinach deficytowych. Informatyka + jest jednym z takich projektów.

1.3. Projekt Informatyka + po roku realizacji

Projekt cieszy się olbrzymią popularnością wśród szkół, nauczycieli i uczniów, objętych jego zasięgiem. Wszystkie wskaźniki ilościowe pierwszego roku realizacji zajęć zostały osiągnięte z nadmiarem. Ciekawe są wyniki badań ankietowych wśród uczestników zajęć. Ponad 66% uczestników zajęć jest zainteresowanych studiowaniem na kierunkach informatycznych; 71% spodziewało się, że udział w zajęciach przyczyni się do poprawy ich szkolnych wyników w nauce przedmiotów informatycznych; i najciekawszy dla organizatorów wynik – 64% uczestników zadeklarowało, że ich uczestnictwo w zajęciach projektu może mieć w przyszłości wpływ na wybór zawodu lub kierunku dalszego kształcenia.

Ciekawe okazały się wyniki ankiet wypełnionych na końcu roku szkolnego: 62% absolwentów techników spośród tych, którzy brali udział w zajęciach projektu (278 uczniów), wybrało studia na kierunkach technicznych związanych z informatyką; 45% uczestników zajęć zadeklarowało, że udział w zajęciach projektu przyczynił się do poprawy ich osiągnięć (w szczególności stopni) z przedmiotu informatyka, a 46% – z przedmiotu technologia informacyjna.

Jako koordynator merytoryczny projektu i wykładowca chciałbym potwierdzić olbrzymią celowość i słuszność założeń tego projektu. Edukacja informatyczna w szkołach wymaga profesjonalnego wsparcia nauczycieli akademickich, którzy w ten sposób przyczyniają się do poszerzenia wiedzy

i umiejętności informatycznych uczniów oraz lepszego przygotowania do podejmowania przez nich dalszej nauki i kariery w kierunkach informatycznych, a generalnie – technicznych.

Realizatorzy projektu Informatyka + myślą o kontynuacji projektu w dalszych latach. Po pierwsze szkoda, by zmarnowały się doświadczenia i materiały dydaktyczne wytworzone w tym projekcie. Po drugie, na bazie już zdobytych doświadczeń i opracowanych materiałów, za znacznie mniejsze fundusze można kontynuować ten projekt w pozostałych regionach kraju.

2. Charakterystyka tej książki

2.1. Refleksje pedagogiczne i psychologiczne

Ta część poradnika zawiera dwa artykuły, których autorki blisko współpracują z Gimnazjum i Liceum Akademickim w Toruniu. Rozważania dotyczą ogólnych aspektów pedagogicznych i psychologicznych, wiążących się z rozwojem uczniów uzdolnionych. Lekturę tych artykułów polecamy wszystkim nauczycielom kierując się dewizą, że każdy uczeń jest w jakimś sensie uzdolniony, drzemią w nim różne talenty, nierzadko czekające na odkrycie a później formowanie. Te ogólne rozważania mogą być przydatne również nauczycielom, których najzdolniejsi uczniowie odnoszą już zauważalne sukcesy – praca z uczniem uzdolnionym przechodzi przez wiele etapów, każdy rządzi się właściwymi metodami postępowania.

2.2. Doświadczenia w pracy z uczniami najzdolniejszymi informatycznie

W tej części znalazły się artykuły nauczycieli ze szkół i z uczelni, którzy mają największe doświadczenia w pracy z uczniami najzdolniejszymi informatycznie, potwierdzone sukcesami w konkursach informatycznych w kraju i za granicą. Warto zwrócić uwagę na różne podejścia w pracy z najzdolniejszymi uczniami.

Przegląd największych sukcesów uczniów z polskich szkół w krajowych i międzynarodowych konkursach algorytmicznych i programistycznych, pióra Krzysztofa Diksa i Jana Madeya, opiekunów uczniów i zespołów startujących w zawodach, przynosi również informacje o dalszych losach i karierach olimpijczyków, czempionów na arenie krajowej i międzynarodowej – równie wspaniałych jak ich sukcesy konkursowe. Za młodu szlifowane diamenty lśnią później pełnym blaskiem.

Krzysztof Diks ze swoimi studentami i doktorantami prezentuje przekrój różnych metod pracy z uczniami, głównie przygotowującymi się do startów w konkursach i zawodach algorytmicznych. Ciekawym aspektem tych metod jest ich pochodzenie – są to metody wypracowane i opisane tutaj przez ... byłych uczniów, zawodników w tego typu konkursach. Ich wcześniejsze sukcesy jakby potwierdzają słuszność obranych przez nich metod i sposobów przygotowywania się do zawodów.

Metody pracy i doświadczenia przedstawione przez Annę Beatę Kwiatkowską odnoszą się nie tylko do uczniów zainteresowanych konkursami algorytmicznymi, ale również do tych, którzy przejawiają zaciekawienie innymi obszarami informatyki, mniej algorytmicznymi i konkursowymi, ale również znacznie rozwijającymi ich kompetencje informatyczne i co więcej, jak się okazuje, nie tylko informatyczne. Pod tym względem ta praca wyróżnia się wśród innych prac w tej części. Takie metody pracy, nastawione na rozwój uczniów w różnorodnych kierunkach danej dziedziny, mogą przynieść uczniom, jak i szkołom olbrzymie sukcesy, czego dowodem może być zdecydowane przodownictwo Gimnazjum i Liceum Akademickiego w Toruniu wśród szkół w plebiscycie *Perspektyw i Rzeczypospolitej* w 2010 roku.

Grzegorz Grochowski opisuje metody pracy Ryszarda Szubartowskiego, który doczekał się największej liczby finalistów i laureatów olimpiady informatycznej i innych konkursów informatycznych w kraju i za granicą. W tym przypadku, metody pracy jednego nauczyciela przynoszą tak olśnie-



wające sukcesy uczniów. W tej pracy, perspektywa sukcesu jest często postrzegana przez pryzmat całościowego rozwoju sylwetki ucznia, z jego uwarunkowaniami psychologicznymi, motorycznymi i osobowymi.

Artykuły w tej części stanowią potwierdzenie tezy, że nie ma wyróżnionej metod pracy z uczniami najzdolniejszymi informatycznie. Trudno też jest określić, jaki sposób przygotowywania uczniów do zawodów przesądza o ich sukcesie. Wydaje się, że zarówno metody kształcenia, jak i drogi rozwoju uczniów najzdolniejszych informatycznie są bardzo indywidualne i najlepszy mistrz takich uczniów musi umieć dobrać i dostosować najlepszą odpowiadającą im metodę pracy.

2.3. Propozycje zajęć z uczniami uzdolnionymi informatycznie

Ta część jest zbiorem przykładowych konspektów zajęć informatycznych, zgromadzonych w czterech modułach, odpowiadających czterem głównym grupom tematycznym zajęć prowadzonych w projekcie Informatyka +. Autorami konspektów są prowadzący zajęcia w tym projekcie. Wszystkie konspekty mają jednolitą strukturę². Konspekty te często odwołują się do obszernych materiałów opracowanych do zajęć w tym projekcie. Materiały te są dostępne na stronie projektu.

Zaproponowane scenariusze zajęć swoim zakresem na ogół wykraczają poza podstawę programową, w różnym jednak stopniu. Na przykład, dwa pierwsze scenariusze w module Algorytmika i programowanie stanowią nieznaczne poszerzenie zapisów podstawy programowej matematyki i informatyki, natomiast dwa dalsze – są przeznaczone głównie dla uczniów wybitnie zainteresowanych programowaniem i algorytmicznym podejściem do rozwiązywania problemów.

W konspektach podano punkty z podstawy, do których odnoszą się proponowane zajęcia, lub których punktów są poszerzeniem. Podano również oczekiwane przygotowanie uczniów do zajęć oraz umiejętności, jakie uczniowie nabędą w trakcie tych zajęć. Dalej następuje szczegółowy opis przebiegu zajęć, często rozbitych na kilka lekcji, nie koniecznie kolejnych.

Zaproponowane scenariusze zajęć zostały wcześniej zrealizowane przez autorów, w warunkach szkolnych lub podczas zajęć w projekcie Informatyka +.

Chciałbym na zakończenie podziękować wszystkim autorom artykułów w tym zbiorze za przychylne potraktowanie moje prośby o współudział w przedsięwzięciu napisania tego poradnika oraz za ich wkład.

Maciej M. Sysło
Uniwersytet Wrocławski
Uniwersytet Mikołaja Kopernika w Toruniu
syslo@ii.uni.wroc.pl

² Strukturę konspektu opracowano m.in. na podstawie propozycji zamieszczonej w pracy. E. Cymerys, I. Chojnacka, To nie takie trudne! Kilka uwag o konspekcie, *Gdyński Kwartalnik Oświatowy* 2/2004, str. 4-8.

CZEŚĆ I

**ROZWAŻANIA OGÓLNE NA TEMAT PRACY
Z UCZNIEM UZDOLNIONYM**



WSPOMAGANIE ROZWOJU ZDOLNYCH UCZNIÓW – REFLEKSJE PEDAGOGA SZKOLNEGO

Hanna Kiedewicz-Nappi

ZS UMK Gimnazjum i Liceum Akademickie, Toruń

nappi@gimakad.torun.pl

*W każdym uczniu drzemią jakieś talenty.
Dlatego ważne jest, by w porę je wyłowić.
Duża w tym rola zarówno nauczycieli, jak i rodziców.*

[minister Katarzyna Hall,
Inauguracja Roku Odkrywania Talentów, Warszawa, 30.09. 2010].

Uczniowie zdolni to nie tylko osoby o wysokim poziomie inteligencji, ale także ci, którzy mają szczególne zdolności związane z konkretnymi dziedzinami: matematyczno-fizycznymi, informatycznymi, przyrodniczymi, społecznymi lub artystycznymi. Wykazują szczególną wrażliwość percepcyjną, świetną pamięć oraz dobrze rozwinięte myślenie związane z konkretnym materiałem, typowym dla określonego rodzaju aktywności. Uczniowie szczególnie zdolni mają zdolności twórcze, zauważają problemy oraz rozwiązują je w nowy i oryginalny sposób. Cechuje ich duża plastyczność myślenia, dążenie do zmian, nonkonformizm, wewnętrzna autonomia, nieuleganie naciskom i wpływom społecznym. Osoby twórcze często kwestionują istniejące standardy społeczne, systemy wartości. Charakteryzują się wewnętrzną samokontrolą, pozwalającą im na branie odpowiedzialności za własne działania, sukcesy i porażki. Szybciej i łatwiej przyswajają wiedzę, koncentrują się na zadaniu, wykazują się wytrwałością w dążeniu do celu. Osoby zdolne są bardzo wrażliwe, interesują się problemami społecznymi (Limont, 2010, str. 90-93).

Zdolności, zainteresowania, możliwości poszczególnych uczniów są różne. Uczniowie o nieprzeciętnych zainteresowaniach wymagają indywidualnego traktowania i dostosowania ścieżek edukacyjnych do ich specjalnych potrzeb. Minister edukacji narodowej Katarzyna Hall ogłosiła rok szkolny 2010/2011 Rokiem Odkrywania Talentów i zwróciła uwagę na problem wspierania uczniów zdolnych (<http://www.roktalentow.men.gov.pl>).

Współczesna cywilizacja wymaga od dziecka uczenia się wielu nowych umiejętności. Formy współżycia społecznego stały się tak złożone, że na ich tle rola plastyczności i zmienności zachowania jest bardzo ważna. Nie tylko technologie informacyjne, ale także organizacja społeczna zmienia się w szybkim tempie. Oczekujemy od dzieci przystosowania społecznego, przy czym zapominamy o ćwiczeniu podstawowych sprawności w zakresie komunikacji społecznej. Uczniowie mają problemy z wyrażaniem uczuć i emocji, szczególnie w sytuacji, gdy zapracowani rodzice poświęcają im zbyt mało czasu. Efektywność funkcjonowania w świecie – zwłaszcza w warunkach nadmiaru informacji – jest wyznaczana umiejętnością dokonywania wyborów. Trafne i podjęte w odpowiednim czasie decyzje przesądzają niejednokrotnie o naszych sukcesach życiowych (Ledzińska, 2005, str. 44).



Zalew informacyjny sprawia, że trafność wyborów źródeł i samych informacji staje się umiejętnością priorytetową. Umiejętności selekcjonowania informacji można i należy nauczyć. Jednostki bezkrytyczne, z ograniczoną możliwością wybierania i brakiem ograniczeń w korzystaniu z Internetu stanowią problem nie tylko edukacyjny. Trudności w selekcjonowaniu informacji, nadmierne poświęcanie czasu na gry komputerowe czy poruszanie się w sieci na zasadzie wolnych skojarzeń, mogą zaburzać psychologiczne funkcjonowanie człowieka w jego wszystkich płaszczyznach: poznawczej, emocjonalnej, społecznej. Inteligencja jednostki (zwłaszcza emocjonalna) jest znacząca w warunkach informacyjnego zalewu (Ledzińska, 2005, str. 49).

Uczniowie zdolni cechują się sprawnym przetwarzaniem informacji związanym ze specyficznym ukierunkowaniem poznawczym. Jednak wymagają szczególnej uwagi ze strony nauczycieli i rodziców z powodu zagrożeń uzależnieniem od komputera/Internetu. Pod pretekstem przygotowania prezentacji, poszukiwania wiadomości, wirtualnych znajomych, bądź tworzenia nowego programu, uczniowie zdolni spędzają wiele godzin na korzystaniu z technologii informacyjnych. Często kosztem snu, wypoczynku, realnych kontaktów z rówieśnikami. Niezbędne jest propagowanie zdrowego stylu życia, uświadomienie znaczenia rozwoju fizycznego dla prawidłowego funkcjonowania organizmu, planowanie czasu wolnego i nauki, rozwój zainteresowań i pasji.

W potocznym rozumieniu osoby zdolne szybko i efektywnie uczą się i mają znacznie więcej szans na osiągnięcie sukcesów w życiu. Jest to tylko częściowo prawdziwe, gdyż sukces nie zawsze jest związany z posiadanymi zdolnościami. Częściej występuje z odpowiednią strukturą osobowości i motywacji, pozwalającymi na długotrwały wysiłek i pracę (Limont, Dreszer, Cieślukowska, 2010, str. 5). Osiągnięcia uczniów nie zależą jedynie od poziomu ich inteligencji, tzw. akademickiej, mierzonej testami inteligencji. Problematyka sukcesu i osiągnięć jest przedmiotem zainteresowania psychologów, filozofów, pedagogów i socjologów. Wiąże się ona ze sferą celów, aspektem czasu i potrzebami człowieka (Sękowski, 2010, str. 88). Na uwagę zasługują badania podejmujące zagadnienie mądrości. Psychologiczne koncepcje mądrości podkreślają znaczenie równowagi między procesami emocjonalnymi i poznawczymi. Przejawem mądrości jest wiedza o życiu, a także adekwatna ocena własnych możliwości i ograniczeń (Sękowski, 2010, str. 90). Człowiek żyjąc w otaczającym go środowisku, wykształca w sobie umiejętności i pewne zdolności, które ułatwiają mu efektywne funkcjonowanie. Dla większości dzieci i dorosłych dyspozycje, umiejętności i zachowania bezpośrednio związane z kompetencją społeczną, stanowią istotną podstawę dla osiągania długoterminowych (i doraźnych) sukcesów w sferze życia zawodowego, edukacji, czy w innych dziedzinach. Stąd konieczność uwzględnienia rozwoju kompetencji społecznych w programach edukacyjnych i profilaktycznych (Wołpiuk-Ochocińska, 2010, str. 113).

Rozwój emocjonalny i społeczny dzieci wybitnie zdolnych odbywa się w określonych warunkach społecznych. Otoczenie społeczne wymaga od uczniów umiejętności społecznych, zdolności empatycznych, umiejętności współpracy, komunikowania się, negocjacji, a także tolerancji. Ze względu na zadowolenie i prawidłowy poziom funkcjonowania osobowego, satysfakcja pojawia się wtedy, gdy zaspokojone są potrzeby emocjonalne. Trudności natury społecznej wiążą się z brakiem poczucia komfortu emocjonalnego. Szczególne znaczenie ma środowisko szkolne, w którym uczeń może rozwiązywać problemy poznawcze i rozwijać się społecznie (Sękowski, 2001, str. 93).

Jednak mistrzowskie opanowanie zagadnień technicznych wymaga długich godzin samotnego ślęczenia nad książkami czy przy komputerze. Często, początki sięgają dzieciństwa albo wczesnej młodości, czyli tych okresów życia, w których uczymy się ważnych umiejętności społecznych poprzez kontakty z rówieśnikami. Osoby, które pociągają dziedziny wyróżniające się potrzebnymi dla ich uprawiania znacznym wysiłkiem umysłowym, jak informatyka czy inżynieria, lgną do nich „po części dlatego, że nie muszą się tam zajmować emocjami” (Goleman, 1999, str. 73). Nie znaczy to, że wszyscy naukowcy o wysokim ilorazie inteligencji są społecznie niekompetentni. W pracy nauko-

wej znakomite osiągnięcia uzyskuje się nie tylko dzięki kompetencji technicznej, ale także takim cechom charakteru jak: **wytrwałość**, **samoświadomość** (wiedza o tym, co odczuwamy w danej chwili), **samoregulacja** (panowanie nad emocjami), **motywacja** (kierowanie się swoimi preferencjami w wyznaczaniu celów), **empatia** (wyczuwanie uczuć innych, podtrzymywanie więzi). Istotne są następujące **umiejętności społeczne**: dobre panowanie nad emocjami, dokładne rozpoznawanie sytuacji społecznych, umiejętność pracy zespołowej (Goleman, 1999, str. 440-441). Praca z uczniem zdolnym nie powinna ograniczać się tylko do stymulowania rozwoju poznawczego oraz emocjonalnego czy społecznego. Ważnym zadaniem psychologów i pedagogów jest wychowanie człowieka obdarzonego szczególnymi zdolnościami (Sękowski, 2001, str. 218).

Zadaniem szkoły jest przygotowanie uczniów do bycia aktywnymi obywatelami. Uczeń powinien nie tylko zdobywać wiedzę i umiejętności (rozumiane jako zastosowanie wiedzy w praktyce, komunikacja, rozwiązywanie problemów), ale także kształtować swoją postawę, czyli tożsamość, odpowiedzialność, współpracę (<http://ekspercibolonscy.pl>).

Wychowanie jako zadanie to przede wszystkim (Nakoneczna, 2003, str. 39):

- dialog nauczających i nauczanych,
- stwarzanie uczniom warunków do samodzielnego podejmowania zadań i ponoszenia za nie odpowiedzialności,
- wdrażanie uczniów do pracy nad sobą, do kierowania własnym rozwojem,
- wprowadzanie młodych w świat prawa moralnego i wartości, w świat życia duchowego,
- kształtowanie ludzi mądrych, znających historię i kulturę własnego kraju, Europy i świata,
- przygotowanie uczniów do odpowiedzialnego życia rodzinnego, zawodowego, społecznego, politycznego,
- przygotowanie do życia w demokracji: do tolerancji, odmienności poglądów i przekonań, do odpowiedzialności za siebie, za sprawy kraju i świata.

Praca nauczyciela z uczniami szczególnie zdolnymi wynika z podstawowej tezy, że nauczyciel nie tylko kształci, ale i wychowuje. Jednocześnie jest przewodnikiem po świecie informacji i wiadomości, co zakłada jego ciągle samokształcenie. Osobowość nauczyciela jest bardzo ważnym czynnikiem. Nauczyciel tworzy atmosferę w klasie, pomaga uczniom w pokonywaniu trudności, niepowodzeń, współpracuje z rodzicami i wspiera ich w wychowaniu (Laznibatova, 2005, str. 121). Osoby zajmujące się edukacją uczniów zdolnych powinny pamiętać, że nadwrażliwość może być związana z osobowością jednostek wybitnie zdolnych, a nie jej zaburzeniami emocjonalnymi.

Wychowanie osób szczególnie zdolnych, czyli pomaganie w życiowym rozwoju, ma dwa ważne aspekty: jeden dotyczy odpowiedzialności społeczeństwa za wychowanie jednostki uzdolnionej, drugi natomiast – odpowiedzialności osoby zdolnej za własny rozwój, działalność, twórczość (Limont, 2005, str. 123).

Uczniowie zdolni to uczniowie o specjalnych potrzebach edukacyjnych i szczególnych wymaganiach związanych z opieką i wspieraniem rozwoju przez środowisko szkolne i rodzinne. Dziecko zdolne wymaga specjalnej opieki psychologiczno-pedagogicznej, umożliwiającej rozwiązywanie problemów związanych z funkcjonowaniem społecznym, emocjonalnym i rozwojem osobowości. Edukacja uczniów zdolnych powinna sprzyjać harmonijnemu rozwojowi, osiągnięciu zadowolenia z życia i satysfakcji z twórczej aktywności. Ważna jest współpraca rodziny i szkoły (Limont, 2010, str. 271).

Uczniowie o wysokim potencjale zdolności mogą osiągać niskie wyniki w nauce. Problem ten określany jest jako syndrom nieadekwatnych osiągnięć. Przyczyną może być złe funkcjonowanie w szkole, związane z nauczycielem, uczniem, rówieśnikami lub organizacją kształcenia. Inną przy-



czyną może być nieprawidłowo funkcjonujące środowisko rodzinne. Diagnoza trudności i zastosowanie odpowiedniego programu interwencyjnego mogą pomóc w pokonywaniu trudności (Limont, 2010, str. 262).

Prawa Rimm pomagające odnieść sukces (Rimm, 2000, str. 11):

- Prawo 1:* Szkolne sukcesy dzieci są bardziej prawdopodobne, jeżeli rodzice przekazują im jednolite i jasno określone wymagania oraz w sposób pozytywny wyrażają się o wysiłku wkładanym w naukę.
- Prawo 2:* Dzieci łatwiej przyswajają sobie odpowiednie zachowania, jeśli mają właściwy wzór do naśladowania.
- Prawo 3:* To, co dorośli mówią o dziecku w jego obecności, ma ogromny wpływ na jego zachowanie i na sposób, w jaki postrzega ono siebie.
- Prawo 4:* Przesadne reakcje rodziców na sukcesy i niepowodzenia dziecka prowadzą do odczuwania przez nie silnej presji osiągnięcia sukcesów, bądź rozpaczki i zniechęcenia w przypadku porażki.
- Prawo 5:* Dzieci odczuwają większe napięcie psychiczne, kiedy martwią się o rezultat swojej pracy, niż wtedy, gdy ją wykonują.
- Prawo 6:* Dzieci budują swoją wiarę w siebie poprzez zmaganie się z trudnościami.
- Prawo 7:* Brak i nadmiar często charakteryzują się takimi samymi symptomami.
- Prawo 8:* Pewność siebie i poczucie odpowiedzialności za własne postępowanie dzieci zdobywają wtedy, gdy zakres ich swobód jest poszerzany stopniowo, w miarę ich dojrzewania.
- Prawo 9:* Dzieci przeciwstawiają się dorosłym, gdy jedno z rodziców poszukuje w dziecku sprzymierzeńca przeciwko drugiemu lub nauczycielowi.
- Prawo 10:* Dorosły powinien unikać konfrontacji z dziećmi, gdy nie jest pewny, że potrafi wyegzekwować swoje zakazy.
- Prawo 11:* Dzieci odniosą sukces tylko wtedy, gdy nauczą się funkcjonować w sytuacji współzawodnictwa.
- Prawo 12:* Dzieci nie przestaną dobrze sobie radzić z nauką, jeśli będą zauważały zależność między procesem uczenia się a jego dobrymi rezultatami.

Uczniowie zdają sobie sprawę, że na sukcesy w nauce mają wpływ różnice dotyczące zdolności i umiejętności, stylów nauczania i uczenia się oraz stopień trudności zadań. Jeżeli jednak uczeń wkłada minimalny wysiłek, a rezultaty są wspaniałe, bądź gdy nauka kosztuje je dużo czasu i wyczerpanej pracy, a brak mu sukcesów, wtedy nie ma wpływu na szkolne osiągnięcia i traci poczucie wewnętrznej kontroli (Rimm, 2000, str. 426-431).

Wybrane formy pracy z uczniem zdolnym

- 1. Wolontariat** – nieodpłatne, świadome oraz dobrowolne działanie na rzecz innych, wykraczające poza więzi rodzinno-koleżeńsko-przyjacielskie. Pomimo intensywnej nauki wolontariusze potrafią wykorzystywać w praktyce swoje zainteresowania i pasje. Są niezwykle wrażliwi na potrzeby innych ludzi, odpowiedzialnie realizują podjęte zobowiązania. Dzięki zdobytemu doświadczeniu i wiedzy wzbogacają swoją osobowość i są liderami w grupie rówieśniczej. Mają

satysfakcję z własnej pracy oraz osiągnięć. Zdobywają umiejętność łączenia współczucia i cierpliwości ze sprawnością działania, co może być przydatne w wyborze kierunku studiów i planowaniu kariery zawodowej. W szkole wolontariusze podejmują działania doraźne (np. zbiórki żywności, zabawek, książek, pieniędzy), działają także poza szkołą w instytucjach niosących pomoc potrzebującym: hospicjum, fundacjach, świetlicach środowiskowych, schronisku dla zwierząt. Wolontariat pomaga w rozwoju inteligencji emocjonalnej, uczy radzenia sobie w trudnych sytuacjach. Uczniowie poprzez współdziałanie w grupie, dyscyplinę, obowiązkowość, zdobywają kompetencje społeczne, które będą przydatne w dorosłym życiu. Wolontariat jest formą działalności, która sprzyja osiągnięciu równowagi w rozwoju intelektualnym, emocjonalno-motywacyjnym oraz społecznym uczniów. Pomaga kształtować postawy prospołeczne oparte na życzliwym stosunku do ludzi (Kiedewicz-Nappi, 2005, str. 283-288).

- 2. Edukacja humanitarna (rozwojowa)** – współpraca z Polską Akcją Humanitarną. Celem jest promowanie postawy humanitarnej oraz takich wartości jak: odpowiedzialność, szacunek, tolerancja, sprawiedliwość, prawa człowieka. Poprzez zajęcia z wolontariuszami (studentami), udział w projektach i akcjach PAH, uczniowie poznają różne sposoby działania na rzecz potrzebujących – tych w najbliższym otoczeniu oraz żyjących w odległych częściach świata. Młodzież uczy się rozumienia otaczającego świata oraz poznaje współzależności łączące nas z innymi regionami świata. Tematy zajęć są starannie dobierane do poziomu rozwoju uczniów i zaawansowania w zakresie edukacji humanitarnej. Szkoły współpracujące z Polską Akcją Humanitarną mogą uczestniczyć w realizacji rocznego projektu edukacyjnego **Szkoła Humanitarna**, skierowanego do szkół podstawowych, gimnazjalnych i ponadgimnazjalnych. Jego celem jest zwiększenie wiedzy uczniów z zakresu globalnych współzależności oraz kształtowanie zaangażowanych i odpowiedzialnych postaw względem otaczającego świata. Udział w projekcie daje możliwość realizacji nowej podstawy programowej kształcenia ogólnego w szkole podstawowej i gimnazjum, która wprowadza elementy edukacji globalnej. Praca metodą projektu stwarza okazję zdobycia umiejętności pracy w grupie oraz kształtowania poczucia odpowiedzialności za wykonanie zadań. Polska Akcja Humanitarna oferuje pomoc szkołom realizującym projekty: szkolenia, konsultacje, materiały edukacyjne. Dzięki realizacji projektu Szkoła Humanitarna: uczniowie lepiej rozumieją kluczowe pojęcia z zakresu praw człowieka, czują empatię i solidarność z potrzebującymi wsparcia, posiadają umiejętność pomagania innym z poszanowaniem ich godności. Cała społeczność szkoły jest aktywnie zaangażowana w działania na rzecz poszanowania godności człowieka i jego praw (<http://www.pah.org.pl>).
- 3. Młody Obywatel** – celem programu jest umożliwienie młodym ludziom prowadzenia działań służących wzrostowi zaufania, współpracy między ludźmi, poczucia tożsamości lokalnej i lepszej komunikacji w miejscowości. W ramach programu uczniowie i nauczyciele zdobywają wiedzę i narzędzia pozwalające badać relacje społeczne. Na podstawie wyciągniętych wniosków można podejmować działania obywatelskie służące rozwojowi lokalnemu we wskazanym przez badania zagrożonym obszarze. Nauczyciele biorący udział w projekcie otrzymują wsparcie metodyczne w postaci 8-miesięcznego kursu internetowego (indywidualny *coaching*), na bieżąco są wspierani przez mentora i mają możliwość kontaktów z innymi nauczycielami – uczestnikami kursu. Wszystkie młodzieżowe działania lokalne podejmowane w ramach projektu są przedstawiane na specjalnej stronie internetowej oraz podczas Ogólnopolskiej Prezentacji Projektów Młodzieżowych (<http://www.ceo.org.pl/portal/home>).
- 4. Szkoła Promująca Zdrowie** – to projekt ogólnoszkolny, którego celem jest nauka dbałości o zdrowie własne i innych ludzi. **Edukacja prozdrowotna** daje uczniom możliwość nabycia



kompetencji (umiejętności, wiedza i postawy), które są niezbędne do odpowiedzialnego uczestnictwa w życiu społecznym, ponieważ:

- pomaga odkrywać, jakie wybory służą rozwojowi i lepszej jakości życia,
- pomaga zrozumieć siebie i innych,
- pomaga budować obraz świata, który jest najbliższy codziennym doświadczeniom,
- uczy odpowiedzialności,
- jest pierwszym stopniem profilaktyki różnych zagrożeń i zachowań problemowych.

Niewiele szkół w Polsce może pochwalić się posiadaniem Krajowego Certyfikatu Szkoły Promującej Zdrowie. Jest to zaszczyt dla szkoły, a jednocześnie zobowiązanie do tworzenia warunków, które sprzyjają dobremu samopoczuciu członków społeczności szkolnej, podejmowania działań na rzecz zdrowia własnego oraz innych ludzi oraz tworzenia zdrowego środowiska (<http://www.ore.edu.pl/>, <http://www.cmp3.ore.edu.pl/node/597>).

5. Program profilaktyczny „Gadu – gadu, czyli jak wyrazić, co mnie gryzie” – autorski program profilaktyczny Barbary Śmiarowskiej. Celem programu jest nauczenie rozumienia i nazywania uczuć, nauka asertywnych zachowań, nabycie umiejętności rozumienia emocji, uświadomienie roli uczuć w kształtowaniu poczucia własnej wartości, poznanie znaczenia empatii w lepszym porozumieniu, wzrost poczucia bezpieczeństwa w relacji ja-klasa. Głównym tematem warsztatów jest nabywanie wiedzy o swoich emocjach odczuwanych w relacji z drugim człowiekiem. Zajęcia obejmują też elementy treningu asertywności i wzmacniania poczucia własnej wartości. Struktura programu jest złożona z dwóch części. Pierwsza z nich to zajęcia wprowadzające i integracyjne, druga – zajęcia praktyczne i teoretyczne, podzielone na 5 bloków: emocje, wyrażanie uczuć, radzenie sobie z emocjami, potrzeby, pozytywny obraz samego siebie. Szczegółowy opis tego programu wraz ze scenariuszami zajęć jest zamieszczony na stronie internetowej (http://www.glospedagogiczny.pl/files/Praca_konkursowa.pdf).

6. Jak żyć z ludźmi – program rozwijania umiejętności społecznych, z elementami profilaktyki uzależnień. Głównym jego celem jest rozwinięcie najważniejszych umiejętności interpersonalnych:

- komunikowania się z ludźmi,
- podejmowania decyzji,
- chronienia siebie w sytuacji nacisku grupowego,
- radzenia sobie ze stresem,
- budowania właściwego wizerunku własnej osoby,
- otwartości,
- empatii.

Ponadto, program dostarcza informacji na temat środków uzależniających. Przeznaczony jest do pracy z młodzieżą ze starszych klas szkół podstawowych, gimnazjalnych i ponadgimnazjalnych. Opis programu wydano w formie książkowej, składającej się z trzech części (<http://www.cmp3.ore.edu.pl/node/13685>):

- pierwsza część, teoretyczna, omawia najważniejsze zasady pracy z grupą, cechy dobrego prowadzącego, niezbędne elementy zajęć z młodzieżą,
- druga – podaje propozycje konkretnych zajęć warsztatowych oraz bogaty zestaw ćwiczeń i technik,
- trzecią stanowią materiały pomocnicze: ankiety, kwestionariusze, rysunki, tabele.

Warto skorzystać ze zbioru rekomendowanych programów profilaktycznych. Zostały one skonstruowane w oparciu o współczesną wiedzę dotyczącą czynników ryzyka i sprawdzone modele oddziaływań profilaktycznych (<http://www.cmp3.ore.edu.pl/node/13718>).

7. **Odyseja Umysłu** – program edukacyjny organizowany przez fundację *Odessey of the Mind*. Jego celem jest rozwijanie zdolności twórczych dzieci i młodzieży poprzez rozwiązywanie problemów o charakterze otwartym. Twórca programu dr C. Samuel Micklus stwierdził, że „nie ma czegoś takiego jak zły pomysł – zawsze natomiast istnieje pomysł lepszy”. Zgodnie z tym założeniem kreatywności można się nauczyć. (<http://www.odyseja.org>). Zajęcia prowadzone w ramach programu rozwijają myślenie krytyczne, rozumowanie dedukcyjne i indukcyjne, wyobraźnię oraz inne właściwości ważne dla twórczego funkcjonowania jednostki. Uczestnicy rozwijają swoje zdolności twórcze, budując urządzenia mechaniczne, pisząc wiersze lub opowiadania oraz konfrontując swoje propozycje na spotkaniach konkursowych, odbywających się na poziomie lokalnym i światowym. Problemy i zadania konkursowe są realizowane w pięciu kategoriach: mechaniki, dzieł sztuki, prezentacji, struktur przestrzennych, realizacji technicznych – uczestnicy budują np. robota z cechami człowieka lub wymyślają utwór na nowy, nieznan instrument (Limont, 2010, str. 202-203).

Głównymi wymaganiami stawianymi przed uczestnikami programu są:

- samodzielne tworzenie rozwiązań,
- odpowiedzialność za własne pomysły i proponowane rozwiązania,
- szacunek dla odmiennych osobowości, pomysłów i poglądów, który przekłada się na dobrą współpracę w grupie.

8. **Twórcze rozwiązywanie problemów** – metoda burzy mózgów. Jest wykorzystywana do zwiększania możliwości poszukiwania rozwiązań problemu oraz do stymulowania myślenia twórczego. W procedurze burzy mózgów wyróżnia się trzy etapy: rozumienie problemu, generowanie pomysłów oraz planowanie działania. Podstawową zasadą jest unikanie krytykowania pojawiających się pomysłów, odrzucanie wartościowania i oceny wygenerowanych pomysłów oraz troska o ich jakość (Limont, 2010, str. 203).

Zdolności, myślenie twórcze i postawy twórcze są ważnymi elementami w funkcjonowaniu zdolnych uczniów. Programy przewyciężające bariery i blokady psychiczne mają charakter treningów, których celem jest wykształcenie umiejętności pokonywania wewnętrznych przeszkód blokujących twórcze działanie. Programy te kształtują postawę otwartości uczniów, stymulują rozwój ciekawości poznawczej, wspierają niezależność i nonkonformizm, rozwijają zdolność dostrzegania problemów. Niezależnie od założeń, programy związane z kształceniem zdolności twórczych sprzyjają wprowadzaniu innowacji pedagogicznych na różnych poziomach edukacji (Limont, 2010, str. 208-209).

Podsumowanie

Uczniowie zdolni powinni rozwijać swoje zainteresowania, korzystając z indywidualnych toków/programów nauczania, uczestniczyć w zajęciach dostosowanych do ich poziomu zdolności i zainteresowań. Jednak nie należy wyłączać tych uczniów z życia klasy i społeczności szkolnej. Cenne są wspólne wyjazdy na obozy (nie tylko naukowe), wycieczki, udział w wymianach młodzieży, projektach edukacyjnych (np. MEP – Młodzieżowy Parlament Państw Nadbałtyckich, Szkoła Humanitarna, Szkoła Globalna), redagowanie gazetki szkolnej, zajęcia sportowe. Należy zachęcać młodzież do ak-



tywności społecznej, poprzez pracę w Samorządzie Uczniowskim, kole Caritas, klubie wolontariusza, w instytucjach pomagających potrzebującym na swoim terenie.

Podjmując się pracy z uczniem zdolnym, nauczyciele muszą tak organizować proces nauczania i wychowania, aby w pełni rozwijać osobowość ucznia, a nie tylko koncentrować się na przyspieszonym rozwoju intelektualnym i/lub uzdolnieniach kierunkowych. Organizacja pracy szkoły powinna uwzględniać indywidualne potrzeby uczniów, ale też sprzyjać rozwojowi tych kompetencji, które będą przydatne w dalszej edukacji i karierze naukowej (zawodowej). Warto włączyć do współpracy pedagoga, psychologa, doradcę zawodowego, którzy wspomagają proces dydaktyczno-wychowawczy w szkole. Poprzez systematyczne zajęcia warsztatowe w klasie, powinni zapoznawać uczniów z technikami uczenia się i pamięci, radzenia sobie ze stresem, rozwijać umiejętności interpersonalne, myślenie twórcze i wyobraźnię, propagować zdrowy styl życia, uczyć technik relaksacji i autoprezentacji. Zdarza się, że młodzież niechętnie bierze udział w takich zajęciach „dodatkowych”, zwłaszcza jeśli nie są one ujęte w stałym planie zajęć. Jednak z czasem uczniowie przekonują się, że warto było spróbować i sami proponują kolejne tematy zajęć lub spotkania ze specjalistami. Ankiety ewaluacyjne, rozmowy z nauczycielami, uczniami i rodzicami, wskazują na potrzebę takich zajęć – zwłaszcza w klasach pierwszych gimnazjum i liceum. Na każdym etapie edukacyjnym warto rozwijać myślenie twórcze uczniów. Wychowawca powinien zachęcać do poszukiwania ciekawych, różnorodnych i nietypowych sposobów rozwiązywania problemów. Wskazane jest kształtowanie konstruktywnego podejścia do błędów, gdyż perfekcjonizm może utrudniać przyjmowanie krytyki i uwag. Korzystne jest bazowanie na mocnych stronach uczniów oraz pozytywne wzmacnianie. Pogłębianie zainteresowań i pasji może sprzyjać rozwijaniu motywacji wewnętrznej, która jest niezbędna w pracach długoterminowych (np. projektach edukacyjnych, wolontariacie).

Bibliografia

1. Covington M. V., Teel K. M. (2004), *Motywacja do nauki*, GWP, Gdańsk.
2. Goleman D. (1999), *Inteligencja emocjonalna w praktyce*, Media Rodzina, Poznań.
3. Kiedewicz-Nappi H. (2005), *Znaczenie wolontariatu w uspołecznieniu uczniów „Gimnazjum i Liceum Akademickiego”*, w: [7], str. 283-288.
4. Laznibatova J., (2005), *Uczeń zdolny na Słowacji – koncepcja opieki na podstawie projektu alternatywnej opieki na uzdolnionych dziećmi*, w: [7], str. 107-122.
5. Ledzińska M. (2005), *Znaczenie inteligencji jednostki w dobie informacyjnego zalewu*, w: [7] str. 41-50.
6. Limont W. (2010), *Uczeń zdolny. Jak go rozpoznać i jak z nim pracować*, GWP, Gdańsk.
7. Limont W. (red.) (2004), *Teoria i praktyka edukacji uczniów zdolnych*, Impuls, Kraków.
8. Limont W., Cieslikowska J. (red.) (2005), *Wybrane zagadnienia edukacji uczniów zdolnych*, T. 2: Uczeń – Nauczyciel – Edukacja, Impuls, Kraków.
9. Limont W., Dreszer J., Cieślikowska J. (red.) (2010), *Osobowościowe i środowiskowe uwarunkowania rozwoju ucznia zdolnego*, WN UMK, Toruń.
10. Nakoneczna D. (2003), *Wychowanie w szkole gimnazjalnej, 20-lecie Towarzystwa Szkół Twórczych 1983-2003*, MH, Warszawa.
11. Rimm S. (2000), *Dlaczego zdolne dzieci nie radzą sobie w szkole. Jak temu przeciwdziałać*, Moderski i S-ka, Poznań.

12. Sękowski A. E. (2001), *Osiągnięcia uczniów zdolnych*, TN KUL, Lublin.
13. Sękowski A. E. (red.) (2004), *Psychologia zdolności. Współczesne kierunki badań*, WN PWN, Warszawa.
14. Sękowski A. E., Klinkosz W. (red.) (2010), *Zdolności człowieka w ujęciu współczesnej psychologii*, TN KUL, Lublin.
15. Wołpiuk-Ochocińska A. (2010), *Kompetencja społeczna człowieka jako wymiar funkcjonowania społecznego jednostki*. w: [13], str. 93-117.
16. <http://www.ceo.org.pl/portal/home>
17. <http://www.cmp3.ore.edu.pl>
18. <http://ekspercibolonscy.pl/prezentacje>
19. http://www.glospedagogiczny.pl/files/Praca_konkursowa.pdf
20. <http://www.gimakad.torun.pl>
21. <http://www.mep-poland.eu>
22. <http://www.odyseja.org>
23. <http://www.pah.org.pl>
24. <http://www.roktalentow.men.gov.pl>



UCZEŃ ZDOLNY W SZKOLE

Katarzyna Śliwińska

ZS UMK Gimnazjum i Liceum Akademickie, Toruń

yen23@wp.pl

W literaturze przedmiotu rozróżnia się trzy podstawowe ujęcia zdolności: pedagogiczne, pedagogiczno-psychologiczne i psychologiczne.

Podejście pedagogiczne opisuje ucznia wybitnie zdolnego jako tego, który osiąga celujące i bardzo dobre wyniki w nauce, oraz jest laureatem olimpiad i konkursów. Koncepcje pedagogiczno-psychologiczne przypisują ponadto wybitnemu uczniowi określone cechy psychiczne związane z motywacją (aktywność, wytrwałość, otwartość) i dobrym funkcjonowaniem społecznym. Psychologiczne rozumienie wybitnych zdolności opiera się natomiast na diagnozie cech osobowości, intelektu i temperamentu (Ćwiok, 2000, str. 227-244).

Współczesne koncepcje zdolności: systemowe (Koncepcja Josepha S. Renzulliego, Koncepcja zdolności Roberta J. Sternberga WISC, Koncepcja Kazimierza Dąbrowskiego) i rozwojowe (Wieloczynnikowy Model Zdolności Franza J. Mönksa, Model Abrahama J. Tannenbauma, Piramida Rozwoju Talentu Jane Piirto, Zróżnicowany Model Zdolności i Talentu Francoys Gagnę) rozpatrują potencjał, jakim dysponują utalentowane jednostki posługując się szeroką perspektywą. Obejmuje funkcje poznawcze, osobowościowe, oraz wpływy środowiskowe (Limont, 2010, str. 39-76).

Ten ostatni czynnik wydaje się o tyle ważny, że nie sposób jest zrozumieć postępowania jednostki bez uwzględnienia kontekstu. A kontekst tworzy środowisko. Środowiskiem ucznia jest rodzina, szkoła i rówieśnicy. W przypadku dzieci i młodzieży w wieku szkolnym wpływy środowiskowe nie tylko tworzą tło działań, ale w sposób szczególny decydują o losach zdolnych jednostek.

W tym kontekście warto zwrócić szczególną uwagę, że uczeń zdolny nie zawsze spełnia oczekiwania o „idealnym uczniu”. Niekiedy jego osiągnięcia nie są adekwatne do posiadanych możliwości, może sprawiać problemy wychowawcze, przeżywać specyficzne trudności związane z tymi samymi cechami, które w sprzyjających okolicznościach stanowią motor rozwoju.

Wiele współczesnych modeli zdolności podkreśla rolę nauczyciela, który właściwie diagnozując potrzeby ucznia zdolnego i oferując mu adekwatną pomoc może odegrać kluczową rolę w urzeczywistnieniu talentu.

Aby skutecznie wspierać rozwój zdolności trzeba przede wszystkim właściwie zidentyfikować jednostki uzdolnione. Nie zawsze jest to proste. Artykuł jest poświęcony wybranym charakterystykom ucznia zdolnego, które pozwalają wejść w jego świat, zobaczyć otaczającą nas rzeczywistość jego oczami. Przede wszystkim jednak, lepiej zrozumieć zachowania, które nie zawsze są zgodne z naszymi oczekiwaniami, dostrzec w nich załączek twórczych inicjatyw, wydobyć go, docenić, a tym samym wspomóc rozwój dojrzałego talentu.

1. Kim jest uczeń zdolny?

W literaturze przedmiotu znajdujemy wiele badań dotyczących ucznia zdolnego. Wymieniane przez autorów cechy można podzielić na kilka grup: pierwsza obejmuje zdolności

poznawcze i kierunkowe, druga – zdolności twórcze, trzecia – cechy osobowości (Limont, 2010, str. 90).

W kontekście cech poznawczych badacze zwracają szczególną uwagę na zdolność szybkiego i efektywnego uczenia się, oraz wybitne osiągnięcia w wartościowej dziedzinie aktywności (Tyszkowa, 1990, str. 166). Charakterystyczna dla uczniów uzdolnionych jest autentyczna satysfakcja z pokonywania trudności i uczenia się rzeczy nowych. Czynności poznawcze (myślenie, uczenie się) u osób uzdolnionych przebiegają sprawniej, są lepiej zorganizowane i osiągają większą efektywność (Tyszkowa, 1990, str. 113).

Z ujęciem, utożsamiającym ucznia zdolnego z wysokimi osiągnięciami szkolnymi, koresponduje potoczna definicja, funkcjonująca w środowisku szkolnym. Z badań prowadzonych w polskich szkołach wynika, że głównymi wskaźnikami identyfikacji ucznia zdolnego podawanymi przez nauczycieli, są ponadprzeciętne osiągnięcia w nauce. Wzorowy uczeń, to taki, który odnosi sukcesy w szkole, jest dobrze dostosowany do jej wymagań i środowiska, wykazuje pozytywny stosunek do obowiązków szkolnych i nie sprawia kłopotów. Cechuje go otwartość na nowość, pracowitość i niezależność myślenia (w granicach wyznaczonych przez nauczyciela) (Tokarz, Słabosz, 2001, str. 37).

Kompleksową charakterystykę ucznia zdolnego przedstawiła Wiesława Limont. Dzieci szczególnie zdolne bardzo wcześnie ujawniają wysoki poziom rozwoju zdolności intelektualnych i/lub kierunkowych. Manifestują aktywność w określonej dziedzinie, szybsze tempo uczenia się i pasję w nabywaniu wiedzy z interesującego zakresu. Wcześniej niż przeciętnie uzdolnione dzieci zaczynają chodzić i mówić, często przed pójściem do szkoły potrafią czytać. Cechuje je duża wiedza, dobra pamięć, bogate słownictwo, zainteresowanie liczbami. Są bardziej zainteresowane wiedzą abstrakcyjną, niż zajęciami praktycznymi. Posiadają także samowiedzę i metawiedzę z zakresu procesów myślenia (Limont, 2010, str. 90-91).

Uczniowie zdolni cechują się nie tylko wysokim poziomem inteligencji, ale też zdolnościami związanymi z konkretnymi dziedzinami: matematyczno-fizycznymi, przyrodniczymi, społecznymi, artystycznymi. Posiadają także zdolności twórcze, wrażliwość na problemy i umiejętność rozwiązywania ich w oryginalny i nowy sposób. Są niezależni w myśleniu, otwarci na nowość i wytrwali w działaniu. Obok zdolności twórczych istotne są także cechy osobowości osób zdolnych. Od najwcześniejszych lat dzieci zdolne są ruchliwe, pełne energii. Specyficzna jest androgynia, która przejawia się w preferencji zabaw typowych dla płci przeciwnej. Osoby zdolne charakteryzują się dużą ciekawością, interesują się nowymi problemami, mają szerokie zainteresowania. Badania wskazują także na adekwatną ocenę swoich możliwości. Od najwcześniejszych lat traktują naukę jako przyjemność. Chętnie dzielą się informacjami z kolegami, są gotowi do współpracy, mają rozwiniętą empatię, poczucie sprawiedliwości oraz uczciwości społecznej i intelektualnej. Osoby zdolne są wrażliwe, mają intuicyjne rozumienie dobra i zła, interesują się problemami społecznymi, cechują się wysokim poziomem rozwoju moralnego, dobrze rozumieją problemy etyczne i moralne (Limont, 2010, str. 93).

Badania pokazują, że uczniowie zdolni są dobrze przystosowani społecznie i lubiani przez swoich kolegów, często wybiera się ich na przyjaciół (French, 1977, str. 463, za: Limont, 2010, str. 93).

Rówieśnicy opisując uzdolnionych kolegów wskazują głównie na funkcjonowanie poznawcze: możliwości przyswajania nowych informacji, logiczne myślenie, dobrą pamięć, wysoką inteligencję, umiejętność koncentracji na zadaniu. Jednocześnie jednak uważają, że ich zdolni koledzy nie są egoistami, nie przeszkadzają w prowadzeniu zajęć, nie wchodzą w konflikty z otoczeniem, nie mają trudności w podporządkowaniu się zasadom, nie czują się osamotnieni (Cieślakowska, Limont, 2008).

Limont zwraca uwagę, że z posiadaniem przez dziecko wybitnych zdolności wiąże się asynchronia rozwojowa związana z nierównomiernym rozwojem różnych funkcji psychicznych. Przykładowo

dziecko zdolne w wieku 5 lat potrafi płynnie czytać, ale ze względu na niedostateczny rozwój kinestetyczno-motoryczny nie potrafi pisać (Limont, 2010, str. 88).

Asynchronia może dotyczyć także sfery emocjonalnej. Dojrzałość intelektualna uwrażliwia na wiele problemów niedostrzeganych przez rówieśników, ale nie pomaga radzić sobie z emocjami doświadczanymi przez dziecko. Wiele aspektów życia nie podlega racjonalizacji i może zostać zrozumianych jedynie dzięki nagromadzonym doświadczeniom (Webb, 1993, str. 529).

Wrażliwość emocjonalna, nadmierny perfekcjonizm, zbyt wysokie i nierealistyczne oczekiwania wobec siebie mogą prowadzić do różnego rodzaju trudności doświadczanych przez osoby zdolne. Maria Ledzińska podkreśla, że za wieloma pozytywnymi charakterystykami funkcjonowania poznawczego, nie zawsze idą wysokie wyniki w nauczaniu. Uczniów zdolnych często cechują przeciętne postępy, zbliżone do ocen uzyskiwanych przez mniej zdolnych kolegów (Ledzińska, 1996, str. 244).

Nauczyciele uzdolnionych uczniów z niskimi osiągnięciami szkolnymi, zidentyfikowali następujące charakterystyczne zachowania dzieci z trudnościami w nauce (Butler-Por, 1993, str. 659):

- czytają bardzo dużo w domu i w klasie, często pod ławką i niekoniecznie to, co wyznaczył nauczyciel,
- prezentują rozległą wiedzę ogólną, ale nie przykładają się do zadań obowiązkowych i prac domowych,
- doskonale radzą sobie z ekspresją werbalną, ale znacznie gorzej w wypowiedziach pisanych,
- wykazują bardzo wysoki poziom kompetencji w dziedzinach, które ich interesują, a jednocześnie nie są związane z programem szkolnym,
- prezentują zachowania unikające, przejawiające się brakiem uczestnictwa w życiu klasy, wysokim wskaźnikiem absencji,
- okazują niezadowolenie z siebie i własnych osiągnięć,
- zachowują się agresywnie i destrukcyjnie w klasie, chociaż doskonale współpracują z nauczycielem podczas sesji indywidualnych,
- stosują manipulacje,
- wykazują braki w kompetencjach społecznych i relacjach z rówieśnikami,
- uzdolnione uczennice z niskimi osiągnięciami charakteryzują się niższymi oczekiwaniami akademickimi i wysokim poziomem lęku przed porażką.

Badania porównawcze uczniów zdolnych z osiągnięciami i ich brakiem przeprowadzone w kilku szkołach nie wykazały istnienia znaczących różnic w charakterystykach osobowościowych (motywacja osiągnięć, twórczość, perfekcjonizm, pobudliwość psychiczna)¹. Wydaje się zatem, że należy postawić pytania dotyczące znaczenia wpływów środowiska na sposób ekspresji danej cechy i tym samym na poziom osiągnięć. Próba odpowiedzi na te pytania, w oparciu o dane z literatury przedmiotu, może stanowić przydatne wskazówki w pracy z uczniem zdolnym.

2. Motywacja osiągnięć

Motywacja jest jednym z najistotniejszych warunków urzeczywistnienia procesu rozwoju zdolności. Motywacja do osiągania sukcesów kształtuje się u ludzi w procesie socjalizacji. Definiuje się ją jako dążenie do doskonałości, odczuwanie pozytywnych emocji w sytuacjach zadaniowych, spotrzeganych jako wyzwanie (Łukaszewski, 2004, str. 461).

¹ Badania sfinansowane z grantu nr 1H01F09430, pt. *Spoleczne zagrożenia uczniów szczególnie zdolnych* (kierownik grantu: prof. dr hab. Wiesława Limonit).

Badania porównawcze uczniów zdolnych z osiągnięciami szkolnymi i nie osiągających sukcesów, nie ujawniły istotnych różnic w poziomie motywacji osiągnięć w obu grupach. Ujawniła się jednak tendencja wskazująca, że w kontekście osiągnięć szkolnych najbardziej optymalna jest motywacja nieco przekraczająca przeciętny poziom (Śliwińska, 2007).

Otrzymane wyniki można interpretować w kontekście klasycznej teorii Yerkesa-Dodsona. Zakłada, że w miarę wzrostu intensywności pobudzenia czynność wykonywana jest do określonego poziomu krytycznego coraz lepiej, po czym zaczyna pogarszać się, aż do całkowitej dezorganizacji. Zbyt silne pobudzenie może prowadzić do ograniczenia możliwości przetwarzania informacji i często odczuwane jest przez jednostkę jako lęk, natomiast zbyt niskie pobudzenie wywołuje znużenie. (za: Łukaszewski, Doliński, 2004, str. 443)

Tokarz zwraca uwagę, że mówiąc o motywacji, należy wspomnieć o jej dwóch rodzajach: wewnętrznej (autonomiczna) i zewnętrznej (instrumentalna) (Tokarz, 2005, str. 43). Próbując zilustrować te różne rodzaje motywacji w funkcjonowaniu dorosłych ludzi, przewagę regulacji autonomicznej odnaleźlibyśmy u osoby pracującej z pasją, lubiącej swój zawód, nastawionej na zmianę i działanie, poszukującej, otwartej, a także bardziej elastycznej, niezależnej od otoczenia, wewnątrzsterownej. Przewagę regulacji instrumentalnej wykazywałaby zaś osoba równie skutecznie wykonująca swoje podstawowe zadania zawodowe, ale nastawiona na ściśle określone cele, unikająca zmian, innowacji, konserwatywna, sztywna w sposobie myślenia i funkcjonowania (Tokarz, 2005, str. 43).

Czynności motywowane autonomicznie i instrumentalnie wywołują odmienne uczucia i emocje, odmiennie wpływają też na nastrój. Czynności motywowane autonomicznie wiążą się z zdziwieniem, zaciekawieniem, przyjemnością, radością, czy wreszcie dumą i satysfakcją. Z satysfakcją wiążą się też czynności motywowane instrumentalnie. Wynika ona z poczucia skuteczności, kontroli nad działaniem i odpowiedniej, zewnętrznej nagrody. Tokarz podkreśla jednak, że jeżeli taka aktywność będzie wiązać się z poznawczo interesującymi rezultatami, w tle może pojawić się również zaciekawienie, zdziwienie, satysfakcja poznawcza, a aktywność może nawet zmienić swój charakter (Tokarz, 2005, str. 45).

Zawiadujący regulacją autonomiczną mechanizm daje pozytywne sprzężenie zwrotne, samonapędzanie się aktywności, a regulacja instrumentalna wywołuje sprzężenie negatywne (zadanie wykonane, aktywność zakończona). Motywacja wewnętrzna skierowana jest więc na aktywność, a motywacja zewnętrzna na rezultat. Biorąc pod uwagę wymienione właściwości nie ma wątpliwości, który z rodzajów motywacji stymuluje rozwój zdolności.

W praktyce oznacza to znaczącą rolę nauczycieli, rodziców w podkreślaniu roli dziecka w osiągnięciu sukcesu, a także znaczenia sukcesu przede wszystkim dla samego dziecka. Mówmy zatem dziecku nie tylko „jesteśmy z ciebie dumni”, ale też „powinnaś/powinieneś być z siebie dumny”. Żeby pochwała sprawiła satysfakcję opiszmy, co nam się konkretnie podoba (Faber, Mazlish, 2006, str. 145). Dziecko widzi wówczas nasze rzeczywiste zainteresowanie, ale też jednocześnie uczy się, co robi dobrze, rozwija swoje zdolności.

Inne ujęcie problematyki motywacji zakłada, że postępowaniem człowieka kierują dwa zasadnicze mechanizmy: motywacja do osiągnięcia sukcesu i motywacja uniknięcia porażki (Łukaszewski, 2004, str. 461). W większości sytuacji ludzie motywowani są przez obie wymienione tendencje, ale u niektórych motyw osiągnięcia sukcesu jest silniejszy od motywu unikania porażki, a u innych jest odwrotnie. Osoby z pozytywną motywacją będą preferowały zadania o średnim stopniu trudności, a dla osób z negatywnym wzorcem motywacji będą one najmniej atrakcyjne. Człowiek zorientowany na unikanie niepowodzeń powinien chętnie podejmować się zadań bardzo łatwych (bo ryzyko porażki jest minimalne) oraz bardzo trudnych (bo choć porażka jest bardzo prawdopodobna, nie przynosi ujemny) (Łukaszewski, 2004, str. 463).

Unikanie podejmowania ryzyka może wystąpić wówczas, gdy młodzież uzdolniona początkowo doznaje jakiejś porażki, zwykle wtedy, gdy przechodzi ze szkoły niższego do wyższego stopnia. Tego typu doświadczenia są często związane z zasadami rekrutacji, mało twórczymi i wymagającymi przeciętnych kwalifikacji i umiejętności (Sękowski, 2000, str. 73).

Brak umiejętności skutecznego radzenia sobie z niepowodzeniem może wynikać z nowości tego doświadczenia dla dziecka zdolnego, które dotychczas odnosiło wyłącznie sukcesy. Jest też sprzeczne z oczekiwaniami otoczenia, które wyrażanym zaskoczeniem często nieświadomie wzmacnia poczucie zagubienia i „nienormalności” tej sytuacji. W konsekwencji uczeń może postrzegać naturalne w procesie kształcenia zjawisko porażki jako zagrożenie dla obrazu siebie, samooceny i w konsekwencji wykształcić negatywną motywację. Unikanie porażek oznacza jednocześnie rezygnację z wyzwań i podejmowania ryzyka, które są niezbędnym elementem rozwoju.

Stąd ważnym elementem pracy z uczniem zdolnym jest uczenie umiejętności dostrzegania pozytywnych aspektów niepowodzeń, traktowania ich jako źródła cennych informacji. Użyteczna jest w tym kontekście praca z biografiami znanych osób, będących autorytetami dla dzieci i młodzieży. Uczniowie zachęceni są do poznania drogi do sukcesu, jaką przechodzą osoby sławne. Zawsze okazuje się, że talent jest wypadkową zdolności, wysiłku i wytrwałości mimo, a czasem dzięki, pierwszym niepowodzeniom. Na przykład nauczyciel Beethovena powiedział o nim kiedyś, że jako „kompozytor jest beznadziejny”. Redaktor gazety wyrzucił z pracy Walta Disneya, bo podobno nie miał żadnych dobrych pomysłów. Pierwsza firma Henry’ego Forda, Detroit Automobile Company, zbankrutowała w ciągu 2 lat. Jego druga firma samochodowa także zbankrutowała. Jednak trzecia, Ford Motor Company sprawiła, że stał się jednym z najbogatszych ludzi w Ameryce. Koszykarz wszechczasów, Michael Jordan, został wyrzucony z licealnej drużyny koszykówki (Robbins, 2005, str. 169). Skuteczność prezentowanej metody wynika z tego, że wierzymy w to, co ludzie robią, a nie mówią.

3. Twórczość

W wielu koncepcjach twórczość jest jednym z podstawowych kryteriów identyfikacji utalentowanych jednostek. Osoby uzdolnione twórczo posiadają umiejętność dostrzegania problemów oraz niestandardowych, nowych rozwiązań. Z łatwością rozpoznają ukryte wzorce w nowych, złożonych strukturach, łączą odległe zagadnienia i informacje w poszukiwaniu rozwiązań. Są ciekawe, otwarte, niezależne w myśleniu (Limont, 2010, str. 93).

Aby potencjał przerodził się w talent musi towarzyszyć mu twórczość. Tymczasem Limont zwraca uwagę na istnienie momentów kryzysowych w rozwoju twórczości. W momencie rozpoczęcia edukacji szkolnej mają związek z koniecznością akceptacji autorytetów i zasad dorosłych, dostosowaniem się do reguł panujących w grupie rówieśniczej, w późniejszym okresie w dążeniu do akceptacji płci przeciwnej, a w końcu z poszukiwaniem aprobaty u osób znaczących i dostosowywaniem się do dorosłego życia (Limont, 2010, str. 89). Wsparcie ze strony otoczenia, w tym szkoły, jest niezwykle ważne w rozwoju twórczości, aby tak było nauczyciele muszą trafnie rozpoznawać i wspierać nie tylko zdolności twórcze, ale też osobowościowe komponenty towarzyszące twórczym jednostkom.

Strzałecki wyodrębnił style zachowania charakterystyczne dla osób twórczych. Proces twórczego rozwiązywania problemów obejmuje określone operacje umysłowe, wewnętrzną reprezentację problemu i system wiedzy jednostki, ale wszystkie wymienione aspekty funkcjonowania poznawczego zostają „przefiltrowane” przez osobowość twórcy (Strzałecki, 1989, str. 186).

Kształcenie w szkole opiera się najczęściej na przekazywaniu i egzekwowaniu wiedzy, sposoby oceniania premiją zachowania konwencjonalne, konwergencyjny sposób rozwiązywania problemów, preferuje się wiedzę pamięciową i encyklopedyczną (Wiechnik, 2000, str. 257). Wyniki badań prezentowanych w literaturze przedmiotu wskazują, że na obraz ucznia idealnego sformułowanego



przez nauczycieli składa się umysł sprawny, skuteczny, otwarty, ale konwencjonalny (Tokarz, 2004, str. 77). Na liście cech sporządzonych przez nauczycieli, które ich zdaniem powinny charakteryzować ucznia wzorowego nie znalazła się ani jedna cecha wskazana jako kreatywna (Wiechnik, 2000, str. 262).

Nonkonformizm uczniów mających wysoki poziom zdolności twórczych powoduje, że sprawiają więcej problemów wychowawczych, mają większe trudności w adaptacji do warunków szkolnych, a to prowadzi do osiągnięć niższych niż spodziewane. Zdolności twórcze nie zawsze są więc cechą oczekiwaną przez nauczycieli i użyteczną w procesie kształcenia (Sękowski, 2000, str. 31).

Badania porównawcze uczniów zdolnych z osiągnięciami i bez sukcesów w kontekście stylów twórczego zachowania nie przekroczyły progu istotności statystycznej (Śliwińska, 2007). Takie wyniki mogą świadczyć o pozytywnym kierunku zmian w nastawieniu nauczycieli wobec twórczych cech uczniów. Wielu nauczycieli uczących w szkołach, w których prowadzono badania, posiada pogłębioną wiedzę w zakresie pracy z uczniem zdolnym, twórczym, co niweluje, przewidywane na podstawie cytowanych wyżej doniesień z literatury przedmiotu, negatywne konsekwencje. Analiza danych jednostkowych, wskazała jednak na pewne tendencje, które mogą stanowić wskazówki w pracy z uczniem twórczym.

Wyższy poziom niezależności sądów i opinii raczej nie przynosi negatywnych konsekwencji w przypadku uczniów z osiągnięciami szkolnymi. Może mieć to związek z faktem, że zainteresowania koncentrują się bardziej na zadaniach szkolnych niż zainteresowaniach z innych dziedzin. Tym samym mieszczą się w granicach niezależności uznanej przez nauczycieli za korzystną w procesie nauczania. Uczniowie z wysokimi osiągnięciami dysponują także podwyższonym stopniem elastyczności w stosowaniu strategii rozwiązywania problemów, łączenia pojęć z innych dziedzin, są bardziej skoncentrowani na podejmowanych działaniach, bardziej pewni siebie i gotowi do pogodnego przyjmowania niepowodzeń, bez uczucia zniechęcenia wobec realizowanych celów. Ich świat wydaje się spójny, klarowny i zmierza w wybranym kierunku aprobowanym przez otoczenie.

Uczniowie z niskimi osiągnięciami mogą natomiast częściej skupiać swoją energię na zainteresowaniach pozaszkolnych, którym poświęcają czas i uwagę kosztem zobowiązań szkolnych. Nonkonformizm w ich wydaniu wydaje się być kojarzony z postawą negacji i wyniosłością, podważaniem autorytetu wykładowcy, dotyczy bowiem treści wykraczających poza granice wyznaczone przez program szkolny. Potrafią elastycznie korzystać ze strategii rozwiązywania problemów, łączyć informacje z różnych dziedzin. Wykorzystują jednak często te umiejętności w realizacji celów niezwiązanych ze szkołą. Mimo odważnie prezentowanych poglądów nie cechuje ich zbyt duża pewność siebie, w razie porażki rezygnują i wycofują się z działania (Śliwińska, 2007).

Rozwiązaniem może być przeniesienie zainteresowań pozaszkolnych na grunt szkolny, danie możliwości zaprezentowania ich w szkole i wyrażenie uznania dla wiedzy i zaangażowania ucznia, co wzmocni jego samoocenę. Wspólne poszukiwanie i akcentowanie znaczenia wiedzy zdobywanej w szkole, a jednocześnie użytecznej w dążeniu do celów wybranych przez samego ucznia, wzmacnia motywację wewnętrzną ucznia do realizacji zadań szkolnych. Limont prezentuje wiele cennych wskazówek dla nauczycieli w pracy z uczniem zdolnym. Wśród nich docenianie twórczych inicjatyw, zwiększanie wrażliwości dzieci, uczenie tolerancji, unikanie schematów, konstruktywny krytycyzm, który rozwija, a nie rani (Limont, 2010, str. 185-187).

4. Pobudliwość psychiczna

Pojęcie „pobudliwości psychicznej” w kontekście zagadnienia jednostek wybitnie zdolnych, zastosował po raz pierwszy Kazimierz Dąbrowski (1937). Na podstawie wnikliwej analizy biografii słynnych twórców (np. Kafki, Michała Anioła i innych – Dąbrowski, 1979, str. 85-105) wyodrębnił

charakterystyczny zespół czynników psychicznych, które określił mianem form psychicznej pobudliwości. Autor w swoich pracach używał pojęcia *potencjał rozwojowy*, na który składają się trzy główne komponenty: szczególne zdolności i talenty, pięć rodzajów pobudliwości i silna, wewnętrzna potrzeba osiągnięć. Pojęcie stało się podstawą opracowanej przez Dąbrowskiego teorii pozytywnej dezintegracji (1979). Badacz oparł swoją koncepcję na założeniu, że proces rozwoju związany jest często z załamywaniem się dotychczasowych struktur psychicznych (życia emocjonalnego, systemu wartości, wzorców zachowań). Według Dąbrowskiego wynikiem tych kryzysów bywają zaburzenia psychiczne (zwłaszcza silny lęk i depresja). Prawidłowy proces rozwojowy prowadzi do konstruktywnego rozwiązania konfliktów wewnętrznych i wytworzenia silnych, unikatowych, autonomicznych, autentycznych wartości i spójnej dojrzałej osobowości (Dąbrowski, 1979, str. 78-85).

Jednym z kluczowych pojęć teorii pozytywnej dezintegracji jest pobudliwość. Autor użył tego terminu, ponieważ trafniej niż pojęcie wrażliwość oddaje ponadprzeciętną intensywność przeżywania, która jest istotą zjawiska.

Teorię pobudliwości jako zespołu cech charakteryzujących osoby wybitnie zdolne rozwinął i upowszechnił Michael Piechowski. Zdaniem autora wzbogacają, nadają moc i rozwijają talent. Podwyższony poziom pobudliwości towarzyszy zatem przeważnie jednostkom uzdolnionym intelektualnie, a najsilniej zaznacza się u osób twórczych (Piechowski, 1997, str. 1).

Dąbrowski wyróżnił pięć form pobudliwości: psychomotoryczną, sensoryczną, intelektualną, wyobrazeniową i emocjonalną (Dąbrowski, 1979, str. 63-65):

1. **Pobudliwość psychomotoryczna** jest stosunkowo łatwa do zaobserwowania przez otoczenie. Można wyróżnić dwie zasadnicze drogi ekspresji tego rodzaju pobudliwości. Jedna z nich dotyczy nadwyżki energii charakteryzującej uzdolnione i kreatywne jednostki. Przejawia się w szybkiej mowie, gwałtownej gestykulacji, intensywnych ćwiczeniach fizycznych itp. Druga przejawia się w pewnej nerwowości w zachowaniu i działaniach destrukcyjnych, uleganiu impulsom, nawykom, będącym ekspresją napięcia emocjonalnego, np. obgryzanie paznokci (Piechowski, 1997, str. 4).
2. **Pobudliwość sensoryczna** oznacza ponadprzeciętną intensywność odbioru wrażeń zmysłowych. Osoba wydaje się być zanurzona w świecie szczególnych doznań, niedostępnych przeciętnym jednostkom. Bez tej niezwykłej wrażliwości niemożliwa byłaby aktywność na wielu polach artystycznej działalności. Manifestacją pobudliwości sensorycznej może być również niezwykle sposób ubierania, zamiłowanie do ozdób i biżuterii (Piechowski, 1997, str. 5).
3. **Pobudliwość intelektualna** obejmuje intensywną aktywność umysłową, głód wiedzy, ciekawość, ale także zdolność koncentracji i długotrwałego zaangażowania. Identyfikacji osób z pobudliwością intelektualną może służyć charakterystyczna wielość stawianych pytań. Dyktowane są pragnieniem zrozumienia istoty problemu i dotarcia do prawdy, choć otoczenie odbiera je często jako próbę podważenia kompetencji rozmówcy. Innym źródłem trudności w relacjach interpersonalnych jest niezależność, która przejawia się w krytyce docierających informacji i negacji powszechnie uznanych autorytetów. Jednostki z tym rodzajem pobudliwości odznaczają się refleksyjnością, umiejętnością wglądu we własne procesy myślowe, odczuwaniem autentycznej przyjemności w analizowaniu problemów i odnajdywaniu rozwiązań (Piechowski, 1997, str. 6).
4. **Pobudliwość wyobrazeniowa** przejawia się w bogatej wyobraźni, tendencji do marzeń na jawie, dostrzeganiu niezwykłych związków, tworzeniu niespotykanych metafor. Charakteryzuje ją także zdolność do wizualizacji, skłonność do animistycznego i magicznego myślenia. Prywatny świat marzeń jest tak bogaty, że wydaje się przenikać rzeczywistość. Nie oznacza to jednak,



że osoby z pobudliwością wyobraźniową nie potrafią odróżnić fikcji od prawdy. Przeciwnie, różnica jest dla nich bardzo wyraźna, a realność wyobrażeń polega przede wszystkim na odczuwaniu intensywnych emocji związanych z tematyką marzenia. Jednostki z pobudliwością wyobraźniową mają trudności w funkcjonowaniu w sytuacjach wymagających wykonywania rutynowych, powtarzających się czynności. Bardzo źle znoszą monotonne zajęcia (Piechowski, 1997, str. 7-8).

5. **Pobudliwość emocjonalną** można określić jako wyższy niż przeciętny poziom uczuciowości, oraz umiejętność odczytywania stanów emocjonalnych u innych osób. Zarówno pozytywne jak i negatywne emocje są doświadczane z bardzo dużą intensywnością. W kulturze zachodnioeuropejskiej ujawnianie przeżywanych emocji jest niepopularne, socjalizacja zmierza raczej w kierunku tłumienia uczuć niż ich swobodnej ekspresji. U dzieci z tym rodzajem pobudliwości powoduje poczucie zagubienia i nieszczęścia. Niezwykła wrażliwość emocjonalna często wiąże się z tendencją do osądzania siebie, co pogłębia uczucie osamotnienia. Osoby z tym rodzajem pobudliwości przywiązują się do ludzi, miejsc, rzeczy. Duże trudności sprawia im konieczność adaptacji do nowego środowiska (Piechowski, 1997, str. 9-10).

Wzmoczona pobudliwość psychiczna, wrażliwość w odbiorze i przeżywaniu bodźców w sferze sensorycznej, intelektualnej, emocjonalnej, psychomotorycznej i wyobraźniowej wymaga odpowiednich strategii wychowawczych, terapeutycznych i edukacyjnych, które powinny być stosowane przez rodziców, nauczycieli i opiekunów w celu stworzenia optymalnych warunków rozwoju (Limont, 2010, str. 89).

Badania porównawcze uczniów zdolnych osiągających sukcesy szkolne i bez sukcesów nie wykazały istotnych statystycznie różnic, analiza wyników jednostkowych ukazała pewne ciekawe tendencje. W próbie ich wyjaśnienia ujęto rolę relacji uczeń-nauczyciel wielokrotnie podkreślaną w literaturze przedmiotu. Wagę nastawienia nauczyciela pokazuje klasyczny już eksperyment Roberta Rosenthala i Eleonory Jacobson (1968) prezentujący tzw. efekt Pigmaliona. W połowie lat 60. XX wieku przeprowadzili oni w jednej ze szkół podstawowych badanie. Każdego ucznia przebadano testem inteligencji i powiedziano nauczycielom, że niektórzy uczniowie otrzymali wyniki tak dobre, że w nadchodzącym roku szkolnym niewątpliwie nastąpi rozkwit ich osiągnięć. W rzeczywistości uczniowie wskazani przez badaczy jako „talenty” zostali wybrani losowo, nie przewyższali swoich rówieśników w szansach na osiągnięcie sukcesu.

Po wywołaniu u nauczycieli wspomnianych oczekiwań, badacze nie ingerowali w dalsze wydarzenia. Pod koniec roku szkolnego okazało się, że dzieci wskazane jako „talenty” znacząco podniosły swoje wyniki. Warto wspomnieć, że nauczyciele nie poświęcili całej uwagi jedynie wyznaczonym uczniom. Stawiali im natomiast wyższe wymagania, zakładając, że mają wyższe zdolności, dostarczali więcej informacji zwrotnych dotyczących ich pracy i częściej dawali szansę na wypowiedzenie się na forum klasy. Te zachowania sprawiły, że uczniowie bardziej się starali, zwiększyła się ich pewność siebie i samoocena (Rosenthal, 1974, za: Aronson, Wilsom, Alert, 1997, str. 143-144). Zatem okazało się, że oczekiwania nauczycieli dotyczące określonego funkcjonowania uczniów działają jak samospełniająca się przepowiednia i wpływają na wyniki uczniów zgodnie z ich nastawieniem.

Wyniki badań nad pobudliwością pokazały, że uczniowie bez sukcesów charakteryzują się w nieco wyższym stopniu ruchliwością, ekspansywnością, nerwowością. U chłopców jednak wymienione cechy wydają się wywierać mniej negatywny skutek na osiągnięcia szkolne, niż w przypadku ich równie uzdolnionych koleżanek. Być może dzieje się tak dlatego, że otoczenie w większym stopniu akceptuje tego typu zachowania u chłopców, gdyż są zgodne z powszechnie przyjętym stereotypem (Bedyńska, Dreszer, 2005, str. 75-87).

Podobne zależności można zaobserwować w przypadku pobudliwości sensorycznej, intelektualnej i emocjonalnej. Pobudliwość sensoryczna manifestuje się często w niekonwencjonalnym, oryginalnym stylu bycia (niezwykły sposób ubierania się, zamiłowanie do ozdób i biżuterii) (Piechowski, 1997, str. 5). Natomiast wizerunek „wzorowego ucznia” jest raczej konwencjonalny, a jego zachowanie i wygląd mieszczą się w granicach obowiązujących norm (Wiechnik, 2000; Tokarz, Słabosz, 2001). Można zatem podejrzewać, że pobudliwość sensoryczna będzie miała negatywny wpływ na osiągnięcia. Tymczasem okazało się, że rzeczywiście może tak być, ale raczej w przypadku chłopców niż dziewcząt. Dlaczego? Powodem może być odmienna percepcja ucznia i uczennicy, która wpływa na zróżnicowane interakcje w układzie nauczyciel-uczeń (Turska, 2006, str. 29). Być może wrażliwość na piękno, poszukiwanie doznań estetycznych, zamiłowanie do ozdób i oryginalnych strojów nie kolidują tak bardzo z obrazem „idealnej uczennicy”, jak w przypadku ucznia wyposażonego w te cechy.

Z analogiczną sytuacją mamy do czynienia w przypadku pobudliwości intelektualnej. Wszyscy badani uczniowie wydają się charakteryzować ponadprzeciętnym poziomem głodu wiedzy, ciekawością, zdolnością koncentracji i innymi cechami, które powinny wspierać wysoki potencjał intelektualny w osiągnięciu sukcesu. Okazuje się jednak, że dziewczęta dysponując podobnym natężeniem badanej cechy co chłopcy, odnoszą częściej sukces szkolny niż oni.

Być może także w tym przypadku odpowiedź można znaleźć w znanych przekonaniach na temat uczniów. Powszechnie uważa się, bowiem dziewczęta za bardziej sumienne i wychodzące naprzeciw oczekiwaniom nauczycieli (pilne). Od typowej uczennicy nauczyciel oczekuje wielu zalet ułatwiających osiągnięcie sukcesu szkolnego: wytrwałości, staranności, dużej koncentracji na zadaniach szkolnych. Oczekiwania takie powodują uwrażliwienie pedagoga na wystąpienie przewidywanych zachowań, które spotykają się z pochwałą i skutkują wieloma dobrymi ocenami. Zatem na drodze warunkowania instrumentalnego, wzmacniana jest u dziewcząt pożądana z perspektywy wymagań szkolnych ekspresja wymiaru pobudliwości intelektualnej (Turska, 2006, str. 1980). Chłopcy zdają się być bardziej narażeni na uczucie nudy i zniechęcenia stereotypowymi zadaniami szkolnymi, a co za tym idzie, skierowaniem uwagi na pozaszkolne zainteresowania. Funkcjonują wówczas według innego szkolnego schematu „zdolny, ale leniwy”, o negatywnym wydzwiekiu, ale uznanym przez nauczycieli. Konsekwencją mogą być obniżone osiągnięcia szkolne.

W przypadku chłopców wysoki i podwyższony poziom pobudliwości emocjonalnej częściej towarzyszy niskim osiągnięciom w nauce. Wydaje się, że chłopcy mogą być narażeni nie tylko na negatywne konsekwencje związane wprost z wysoką wrażliwością emocjonalną, ale także z brakiem zrozumienia ze strony otoczenia ze względu na niezgodność z funkcjonującym stereotypem. Chłopcy postrzegani są jako ci, którzy powinni radzić sobie z uczuciami smutku, przygnębienia, a najlepiej w ogóle ich nie ujawniać (każdy nastolatek wie, że *chłopaki nie płaczą*).

Dziewczęta charakteryzujące się związaną z pobudliwością emocjonalną wrażliwością, mogą otrzymywać ze strony otoczenia więcej wsparcia niwelującego negatywne konsekwencje tych stanów, niż chłopcy (dziewczynce bardziej wypada melancholia). Być może dlatego nie różnią się od siebie pod względem tego wymiaru, bez względu na osiągnięte wyniki szkolne.

W przypadku pobudliwości wyobraźniowej spodziewano się, że ze względu na często towarzyszącą jej trudności w koncentracji na zadaniu, wysoki i podwyższony poziom będzie negatywnie wpływać na osiągnięcia szkolne. Wydaje się, że w przypadku chłopców podwyższone natężenie omawianej cechy rzeczywiście wywiera negatywny wpływ na osiągnięcia szkolne. Z drugiej strony w przypadku dziewcząt zależność jest odwrotna. Być może potrafią lepiej niż chłopcy wykorzystać bogatą wyobraźnię, dostrzeganie niezwykłych związków, tworzenie niespotykanych metafor. Nie rezygnując z prywatnego świata marzeń wywiązują się z wykonywania zadań rutynowych, charakterystycznych dla wymagań szkolnych, które stanowią duży problem dla ich kolegów z podwyższonym poziomem tego rodzaju pobudliwości (Śliwińska, 2007).



Mimo, że przedstawione powyżej rozważania mają charakter hipotetyczny i wymagają weryfikacji w toku dalszych badań, podkreślają rolę nauczyciela we wspomaganiu rozwoju zdolności. Każdy z nas postrzega rzeczywistość przez pryzmat wypracowanych w toku doświadczenia przekonań, oczekiwań, schematów. W większości przypadków służą one usprawnianiu procesu podejmowania decyzji i są pomocne w pracy. Ważna jest świadomość posiadanych schematów poznawczych, wówczas stają się użytecznym narzędziem w naszym ręku i jednocześnie pozwalają dostrzegać to, co nieschematyczne, wyjątkowe. Tak jak wyjątkowe, niekiedy w nieoczywisty sposób, są zdolne dzieci.

5. Perfekcjonizm

Perfekcjonizm jest definiowany jako kombinacja myśli i zachowań związanych z wysokimi standardami lub oczekiwaniami, co do własnych osiągnięć. Jest cechą charakteryzującą utalentowane osoby (Hamachek, 1978, za: Schuler, Ferbezer, O'Leary i inn., 2003, str. 67).

Zdrowy perfekcjonizm może stanowić użyteczne narzędzie w dążeniu do doskonałości, funkcjonowania na optymalnym poziomie i osiągnięcia pełnej kontroli nad sobą. Część badaczy zwraca jednak uwagę na destrukcyjną rolę perfekcjonizmu. Osoby charakteryzujące się tzw. neurotycznym perfekcjonizmem, podlegają ciągłemu stresowi towarzyszącemu nierealistycznym planom. Perfekcjonści tego rodzaju oceniają swoją wartość przez pryzmat osiągnięć i produktywność w realizacji niemożliwych celów (Schuler, Ferbezer, O'Leary i inn., 2003, str. 67).

Zapał i zaangażowanie w podejmowaną aktywność w kombinacji z nierealistycznymi celami, pochłaniają dużą ilość czasu i energii, nierzadko bez widocznych rezultatów. W konsekwencji perfekcjonizm u ok. 15-20% wysoko uzdolnionych uczniów w pewnym momencie edukacji staje się istotnym powodem trudności w nauce (Webb, 1993, str. 529).

Niskie osiągnięcia, mimo wysokiego potencjału, spowodowane są napięciem, jakie towarzyszy osobom z neurotycznym perfekcjonizmem. Żadne rozwiązanie nie wydaje się wystarczająco dobre, aby można było je uznać za zadowalające. W związku z tym perfekcjonści mają trudności z zakończeniem podjętego zadania. Prowadzi to do lęku przed porażką i unikania sytuacji postrzeganych jako ryzykowne. Ostatecznie możemy mieć do czynienia z ukształtowaniem negatywnego obrazu siebie, związanego z nadmierną samokrytyką i prowadzącego do pogłębiania się problemu (Śliwińska, Limont, Dreszer, 2008, str. 132).

Badania wskazują, że dobra organizacja, staranność i uporządkowanie są ściśle związane z tzw. zdrowym perfekcjonizmem i wspierają rozwój zdolności (za: Schuler, Ferbezer, O'Leary i inn., 2003, str. 68).

Wyniki badań sugerują istnienie pewnej zależności między skłonnością do koncentracji na błędach, a obniżonymi osiągnięciami. Generalizacja porażek na wszelkie aspekty funkcjonowania jednostki może zaowocować brakiem wiary w możliwość sprostania szkolnym wymaganiom tym samym stać się przyczyną niskich wyników w nauce (za: Butler Por, 1993, str. 658, Śliwińska, 2007).

Uczniowie zdolni z niskimi osiągnięciami w nauce znacznie dotkliwiej odczuwają presję związaną z wymaganiami i krytyką, niż ich równie zdolni koledzy z wysokimi ocenami. Zbyt wysokie oczekiwania, powodują, że dzieci czują się bezsilne w próbach sprostania im i nie są zdolne do czerpania satysfakcji z osiągniętych sukcesów. Stała presja w kierunku wysokich osiągnięć powoduje frustrację i brak poczucia spełnienia. W konsekwencji może zrodzić się obawa przed porażką i krytyką, co prowadzi do unikania nowych wyzwań (Śliwińska, Limont, Dreszer, 2008, str. 112).

W pracy z uczniem zdolnym powinno zmierzać się do wzmacniania zachowań związanych z perfekcjonizmem zdrowym. Obok omówionej wcześniej pracy nad akceptacją błędów, pomyłek, własnej niedoskonałości, istotną rolę odgrywa rozwijanie umiejętności stawiania sobie celów krótko-

i długoterminowych, planowania, zarządzania czasem, w tym czasem wolnym. Właściwy odpoczynek jest niezbędny, aby dzieci zdolne mogły czerpać satysfakcję i radość z nauki.

Ważną rolę odgrywa również konstruktywna krytyka, która obejmuje akcentowanie pozytywnych elementów wykonanego zadania, wskazuje konkretne błędy i dostarcza wskazówek jak je poprawić. Wówczas dziecko odbiera ją jako element nauki, rozwoju, a nie niepowodzenie.

6. Podsumowanie

Uczeń zdolny, dziecko zdolne nosi w sobie obietnicę. Zadaniem szkoły jest wspieranie procesu urzeczywistniania zdolności w talent. W świetle przedstawionych rozważań założenie, że uczeń zdolny nie wymaga uwagi i szczególnego zainteresowania wydaje się być pozbawione podstaw. Opracowanie skutecznych metod pracy z uczniem zdolnym, programów wspierających rozwój zdolności wymaga znajomości specyfiki funkcjonowania dziecka zdolnego.

Bardzo ważna jest rola nauczyciela. Jak podkreśla Wiesława Limont wychowawcom i pedagogom pracującym ze zdolnymi uczniami stawia się wysokie wymagania. Mają być ekspertami w swoich dziedzinach, umiejętnie nauczać uczniów o bardzo zróżnicowanych możliwościach, wymaganiach i potrzebach. Nauczyciel jest mistrzem i mentorem. Pracując z pasją, rozumiejąc swoich uczniów stajemy się doskonałym wzorem do naśladowania, bo uczymy się przede wszystkim obserwując.

Mistrz i uczeń dziwią się razem.

Bibliografia

1. Aronson E., Wilson T. D., Alert R. M., *Psychologia społeczna. Serce i umysł*, Wydawnictwo Zysk i S-ka, Poznań 1997.
2. Bedyńska S., Dreszer J., *Stracone talenty. Negatywny stereotyp jako czynnik odpowiedzialny za niskie osiągnięcia szkolne uczniów zdolnych*, [w:] Limont W., Cieślukowska J. (red.), *Wybrane zagadnienia edukacji uczniów zdolnych*, Tom I, Impuls, Kraków 2005.
3. Butler-Por N., *Underachieving Gifted Students*, [w:] Heller K. A., Monks F. J., Passow A. H. (red.), *International Handbook of Research and Development of Giftedness and Talent*, 1993.
4. Cieślukowska J., Limont W., *Competencies and qualities of teachers of specialty gifted students in the light of popular opinions held by students*, V Międzynarodowa Konferencja Od rozwijania nadania k rozwijaniu osobnosti nadanych, Bratysława, 4-5.04.2008.
5. Czerwińska-Jasiewicz M., *Postawy nauczycieli i rówieśników w stosunku do uczniów wyróżniających się inteligencją i zdolnościami twórczymi*, *Psychologia wychowawcza*, 1984, 4.
6. Ćwiok E., *Motywy uczenia się młodzieży wybitnie uzdolnionej i przeciętnie zdolnej*, *Przegląd Psychologiczny*, 2000, 41(1/2).
7. Dąbrowski K., *Dezintegracja pozytywna*, WSiP, Warszawa 1979.
8. Faber A., Mazlish E., *Jak mówić żeby dzieci się uczyły. W domu i szkole*, Media Rodzina, Poznań 2006.
9. French J. L., *Dzieci szczególnie uzdolnione – badania i metody kształcenia*, [w:] Haring N. G., Schiefelbusch R. L. (red.), *Metody pedagogiki specjalnej*, PWN, Warszawa 1977.
10. Hamachek D. E., *Psychodynamics of norma and naurotic perfectionism*, *Psychology* 1978, 15.



11. Ledzińska M., *Przetwarzanie informacji przez uczniów o zróżnicowanym poziomie zdolności, a ich postępy szkolne*, WP UW, Warszawa 1996.
12. Limont W., *Uczeń zdolny. Jak go rozpoznać i jak z nim pracować*, GWP, Gdańsk 2010.
13. Limont W., Śliwińska K., Dreszer J., Bedyńska S., *Wzmoczona pobudliwość psychiczna, a perfekcjonizm uczniów zdolnych*, *Psychologia. Edukacja i społeczeństwo*, Tom 1, nr 1 i 2, 2008.
14. Łukaszewski W., *Motywacja w najważniejszych systemach teoretycznych*, [w:] Strelau J. (red.), *Psychologia. Podręcznik akademicki*, GWP, Gdańsk 2004.
15. Piechowski M. M., *Emotional giftedness: The measure of intrapersonal intelligence*, [w:] Colangelo N., Davis G.A. (red.), *Handbook of gifted education*, Allyn and Bacon, Needham MA 1997.
16. Piechowski M. M., *Mellow Out They Say. If I Only Could. Intensities and Sensitivities of the Young and Bright*, Yunasa Books, Madison Wisconsin 2006.
17. Robbins S. P., *Skuteczne podejmowanie decyzji*, PWE, Warszawa 2005.
18. Schuler P. A., Ferbezer I., O'Leary i inn., *Perfectionism; International Case Studies, Gifted and Talented International*, vol. XVIII, No. 2, 2003.
19. Sękowski A. E., *Osiągnięcia uczniów zdolnych*, Towarzystwo Naukowe KUL, Lublin 2000.
20. Strzalecki A., *Twórczość, a style rozwiązywania problemów praktycznych. Ujęcie prakseologiczne*, Ossolineum, Wrocław-Warszawa-Kraków 1989.
21. Śliwińska K., *Osobowościowe uwarunkowania niskich wyników Osiągnięć nauczaniu uczniów szczególnie zdolnych (Syndrom Nieadekwatnych Osiągnięć)*, niepublikowana praca magisterska, 2007.
22. Śliwińska K., Limont W., Dreszer J., *Perfekcjonizm, a osiągnięcia szkolne uczniów zdolnych*, [w:] Limont W., Cieślukowska J., Dreszer J. (red.), *Zdolności, talent, twórczość*, Wydawnictwo Naukowe Mikołaja Kopernika, Toruń 2008.
23. Tokarz A., *Procesy motywacyjne, a dyspozycje do wybitnych osiągnięć w kontekście rozwoju*, [w:] Limont W., Cieślukowska J. (red.), *Wybrane zagadnienia edukacji uczniów zdolnych*, Tom II, Impuls, Kraków 2005.
24. Tokarz A., Słabosz A., *Cechy uczniów preferowane przez nauczycieli jako wymiar aktywności twórczej w szkole*, Cz. II, *Edukacja. Studia. Badania. Innowacja*, 2001, 3 (75).
25. Turska D., *Skuteczność ucznia*, UMCS, Lublin 2006.
26. Tyszkowa M., *Zdolności, osobowość i działalność uczniów*, PWN, Warszawa 1990.
27. Webb J. T., *Nurturing Social-Emotional Development of Gifted Children*, [w:] Heller K. A., Monks F. J., Passow A. H. (red.), *International Handbook of Research and Development of Giftedness and Talent*, Pergamon Press, Oxford 1993.
28. Wiechnik R., *Obraz ucznia idealnego w percepcji nauczycieli szkół podstawowych*, *Psychologia Rozwojowa* 2000, 5, 3-4.



CZEŚĆ II

**DOŚWIADCZENIA W PRACY
Z UCZNIEM UZDOLNIONYM INFORMATYCZNIE**



RÓŻNE PODEJŚCIA W PRACY Z UCZNIAMI UZDOLNIONYMI INFORMATYCZNIE

*Krzysztof Diks
Jakub Radoszewski, Jakub Łącki
Marek Cygan, Tomasz Idziaszek*

Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

{diks, jrad, cygan, j.lacki, idziaszek}@mimuw.edu.pl

W tym artykule zamieszczamy przemyślenia osób związanych z Olimpiadą Informatyczną na temat pracy z uczniem uzdolnionym informatycznie. Przemyślenia są osobiste. Czasami zawierają różne spojrzenie na ten sam problem. Ich największą zaletą jest to, że przedstawiają je osoby, które niewątpliwie są utalentowane, osiągnęły sukces i starają się teraz same docierać do młodzieży uzdolnionej informatycznie i pracować z nią systematycznie.

1. Cztery elementy w pracy z uczniem zdolnym (Krzysztof Diks)

Moje przemyślenia w pracy z uczniem zdolnym, to przemyślenia nauczyciela akademickiego, który nigdy nie uczył w szkole i którego kontakt z uczniami uzdolnionymi informatycznie ma miejsce tylko poprzez Olimpiadę Informatyczną. Z drugiej strony jako nauczyciel akademicki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego miałem do czynienia z kilkuset studentami, o których śmiało można powiedzieć, że stanowili (i stanowią) śmietankę polskiej informatyki. Moje przemyślenia sformułuję w czterech punktach.

1. Talenty rodzą się wszędzie, zarówno w wielkich miastach, jak i małych osadach

Talent to diament, który należy dostrzec, wydobyć i oszlifować. W początkach tego procesu nic nie zastąpi rodziców i nauczycieli. W okresie wczesno szkolnym szkodliwa jest wszelkiego rodzaju specjalizacja. Dzieci przejawiające zdolności w naukach ścisłych należy zachęcać do czytania popularnych książek z matematyki, fizyki, chemii itp. Należy też zachęcać je do rozwiązywania zagadek logicznych i nauki gry w warcaby, szachy, Go. Niezmiernie ważna jest już w tym okresie możliwość szlachetnej rywalizacji. Służą temu różnego rodzaju konkursy, takie jak Kangur i Bóbr. Udział w konkursach umożliwia kontakty z innymi uczniami o podobnych zainteresowaniach i zdolnościach i tworzeniu grup rówieśniczych, w których wybitni uczniowie nie czują się wyobcowani.

2. Rola nauczyciela (rodzica) zmienia się wraz z rozwojem talentu ucznia

W początkowym etapie należy ucznia wprowadzić w dziedzinę. Przekazać mu minimum wiedzy i umiejętności niezbędnych w dalszym rozwoju. W przypadku informatyki polega to głównie na nauczaniu programowania w małej skali i podstaw algorytmiki. Według mnie programowania nie należy uczyć w oderwaniu od algorytmiki. Program, to przecież formalny zapis algorytmu. Początkujący programista powinien poznać tylko podstawowe instrukcje sterujące (if, while), proste typy danych i tablice oraz umieć korzystać z procedur i funkcji. Uczenie pełnego języka programowania, ze wszystkimi jego zawiłościami, jest na tym etapie zupełnie niepotrzebne. W późniejszym okresie



pracy z uczniem rola opiekuna powinna się ograniczyć do przyjaznego stymulowania, podsuwaniu stosownej lektury oraz problemów do rozwiązywania. Najlepsi są w stanie pracować samodzielnie i szybko się uczyć, ale potrzebują odpowiednich bodźców – życzliwego zainteresowania, docenienia ich pracy.

3. Konkursy informatyczne motorem rozwoju talentu

Wspomniałem już o korzyściach płynących z udziału w różnego rodzaju konkursach. Według mnie olimpiady przedmiotowe są jednym z najważniejszych elementów rozwoju uczniów utalentowanych. W tym roku szkolnym (2010/2011) odbywa się XVIII edycja Olimpiady Informatycznej. We wszystkich edycjach Olimpiady wzięło udział kilkanaście tysięcy uczniów. Zmieniają oni oblicze naszego kraju. Dla przykładu portal społecznościowy Nasza Klasa stworzyli byli olimpijczycy, podobnie coraz popularniejszy portal Codility.com jest też dziełem olimpijczyków.

Dlaczego udział w Olimpiadzie jest ważny? Żeby osiągnąć w niej sukces należy wykształcić cechy, które są niezbędne w osiągnięciu sukcesu w życiu szkolnym, naukowym, czy też zawodowym. Należą do nich między innymi: pracowitość, wytrwałość, samodzielność, dociekliwość, chęć rywalizacji, otwartość. Ludzie zdolni chcieliby rozwiązywać zadania (a w przyszłości problemy) na miarę ich zdolności i umiejętności. Rozwiązanie zadania powinno sprawiać satysfakcję, a jego nierozwiązanie pobudzać do dalszej nauki i rozwijania talentu. Zadania stawiane uczniom powinny być tak układowe, żeby pozwalały uczniom zrobić kolejny krok na drodze ich rozwoju – poznać i zrozumieć nową technikę, odkryć nowy algorytm.

Olimpiada to też miejsce tworzenia się wspólnoty zainteresowań. Niezmiernie ważną rolę w Olimpiadzie Informatycznej odgrywa forum dostępne w Systemie Informatycznym Olimpiady. Jest to miejsce dyskusji nad zadaniami, wymiany doświadczeń, dzielenia się testami, nawiązywania znajomości. Udział w Olimpiadzie uczy też profesjonalizmu. Organizatorzy starają się przygotować profesjonalny serwis i wymagają profesjonalizmu od uczestników, w szczególności etycznego zachowania.

4. Rola autorytetów

Młodzi ludzie, a szczególnie ludzie utalentowani potrzebują autorytetów, które mogą stanowić wzorzec dla rozwoju ich karier. Dlatego z młodymi ludźmi powinni pracować starsi koledzy, którzy osiągnęli sukces i mogą być wzorem do naśladowania. W Olimpiadzie Informatycznej wiele prac wykonują (układają zadania, piszą wzorcowe rozwiązania, odpowiadają za oprogramowanie, prowadzą warsztaty) byli olimpijczycy, którzy osiągnęli sukcesy na skalę światową. Są to zwycięzcy Olimpiady Informatycznej oraz medaliści prestiżowych konkursów międzynarodowych, najlepsi studenci, doktoranci i cenieni pracownicy polskich uczelni. Ich wiedza i umiejętności są doceniane w skali globalnej. Największe firmy branży IT starają się przyciągnąć ich do siebie. Wielu byłych olimpijczyków osiąga też sukcesy naukowe w skali świata. Talent wymaga docenienia. Rozumieją to znani naukowcy-informatycy, którzy z pasją uczestniczą w pracach Olimpiady i dzielą się swoją wiedzą z olimpijczykami.

2. Wyławianie talentów informatycznych i budzenie zainteresowań uczniów informatyką, w szczególności algorytmiką (Jakub Radoszewski)

Istnieją różne drogi, którymi można młodzież zainteresować informatyką, w szczególności programowaniem i algorytmiką. Bodaj najprostszym pomysłem jest systematyczne prowadzenie podopiecznych: najpierw przez bardziej zaawansowane korzystanie z komputera (można z dużą dozą pewności założyć u młodzieży podstawową znajomość komputera), dalej przez składnię wybranego

języka programowania (względnie także przez prostszy język pośredni), by wreszcie dotrzeć do świata ciekawszych zagadnień algorytmicznych. Jest to w dużej mierze metoda małych kroków, dobra dla podopiecznych lubiących odkrycia natury technicznej, stymulująca wizualnie. Niestety, używając tej metody, może być trudniej zachęcić do informatyki osoby wybitnie utalentowane, lubiące przede wszystkim rozwiązywać ambitne łamigłówki, a nie przedzierać się przez gąszcz zagadnień technicznych. Takie osoby mogły już wcześniej odkryć swoje zainteresowania innymi przedmiotami ścisłymi.

Dobrą próbą wzbudzenia zainteresowania utalentowanych uczniów informatyką może być pokazanie im ciekawych zagadnień algorytmicznych z pominięciem uprzedniego wprowadzania całej warstwy technicznej. Dzięki temu podejściu uczniowie mogą antycypować to, z jakimi problemami będą mogli zmierzyć się po przyswojeniu sobie niezbędnego materiału natury technicznej. Innymi słowy, można ich zainteresować programowaniem, zainteresowawszy ich uprzednio algorytmiką. W tym duchu można, na przykład, zacząć od wprowadzenia schematów blokowych i na nich prezentować pierwsze przykłady algorytmów. Jest to ciekawy pomysł, choć takie schematy mogą wydawać się uczniom nieco sztuczne i przez to mniej interesujące.

Nasz pomysł¹ polega na przedstawieniu uczniom łamigłówek sformułowanych w języku matematyki (z użyciem bardzo prostych pojęć), których rozwiązania wymagają jedynie użycia kartki i długopisu, ale zawierają istotne elementy myślenia algorytmicznego. Przykłady takich zadań można znaleźć w moich artykułach, zamieszczonych w czasopiśmie *Delta*: Zadanka (nie)informatyczne, *Delta* 8/2009 i Liczba liczb, *Delta* 6/2010². Zadania wspomnianego typu można tworzyć na podstawie konkursowych zadań informatycznych, biorąc konkretne dane wejściowe, których rozmiar nie może być zbyt mały (trywialne zadanie), ani zbyt duży (zadanie zbyt męczące, przez to niezbyt interesujące). Nie każdy problem algorytmiczny można sformułować w ten sposób – trzeba wziąć pod uwagę to, że rozwiązanie natury algorytmicznej, które chcemy wymusić, powinno być optymalne wśród możliwych rozwiązań pod względem łącznego czasu jego wymyślenia, powiększonego o czas jego ręcznego wykonywania. W przeciwnym przypadku rozwiązujący chętniej wybiorą, albo mniej ciekawą metodę siłową, pozwalającą wyznaczyć wynik za pomocą większej, choć wciąż rozsądnej liczby operacji, albo metody heurystyczne, które działają dokładnie dla wskazanych danych. Innymi słowy, w przypadku takich zadań stosujemy inną definicję złożoności obliczeniowej rozwiązania.

Podobne do powyższego pomysły występują w niektórych konkursach informatycznych. W Polsce najbardziej znany jest konkurs Bóbr (<http://www.bobr.edu.pl/>), w którym uczestnicy rozwiązują test złożony z krótkich i stosunkowo prostych zadań-łamigłówek. Innymi przykładami są: Olimpiada Informatyczna w RPA³ oraz Australijski Konkurs Informatyczny.⁴ Jednym z głównych celów tych konkursów jest popularyzacja informatyki, w czym korzystają one z paradygmatu *competitive learning*. Nasz pomysł różni się od nich tym, że celuje głównie w popularyzację informatyki wśród uczniów zainteresowanych naukami ścisłymi lub szczególnie utalentowanych w tym kierunku. Ponadto, rozwiązane przez nas zadania nie mają charakteru konkursowego: mogą być bardziej złożone, wymagać dłuższego kombinowania, zawierać wskazówki. Rezygnacja z formatu konkursowego jest celowa – opieramy się na naturalnym zainteresowaniu uczniów zagadkami i łamigłówkami (na przykład ostatnio coraz bardziej popularnymi łamigłówkami Sudoku i obrazkami logicznymi).

¹ Kubica M., Radoszewski J., Algorithms without Programming, *Olympiads in Informatics* 4, 2010, str. 52-66.

² Polecamy portal *Deltę* pod adresem <http://www.deltami.edu.pl/>, zawierający bardzo ciekawe materiały dla uczniów szukających ciekawych zadań i ich rozwiązań, jak również prezentacji zagadnień z obszaru ciekawej matematyki i informatyki.

³ Merry B., Gallotta M., Hultquist C., Challenges in Running a Computer Olympiad in South Africa, *Olympiads in Informatics* 2, 2008, str. 105-114.

⁴ Burton B. A., Encouraging Algorithmic Thinking Without a Computer, *Olympiads in Informatics* 4, 2010, str. 3-14.



3. Uczenie algorytmiki i programowania od podstaw (*Jakub Łącki*)

Choć pytanie o metody pracy z uczniami wybitnie zdolnymi wydaje się bardzo naturalne, trudno sformułować kategoryczne opinie na ten temat. Po pierwsze, uczniowie wybitnie zdolni, są, niejako z definicji, grupą nieliczną i wszelkie wnioski wyciągane na podstawie doświadczeń w ich nauczaniu są wielce niepewne. Po drugie, uczenie młodzieży algorytmiki rozpoczęło się stosunkowo niedawno i nawet najstarsi nauczyciele mają za sobą zaledwie kilkanaście lat doświadczeń. Wydaje się zatem, że wszelkie metody można zaproponować jedynie na podstawie niewielkiego doświadczenia, popartego intuicją i zdrowym rozsądkiem. Jednak i to może nie być skuteczną metodą. Mimo wszystko spróbuję zaprezentować swój punkt widzenia w tej sprawie, zastrzegam jednak, że będzie to zaledwie efekt moich przemyśleń i skromnych doświadczeń. Oprę się głównie na swoich doświadczeniach z czasów, gdy byłem uczniem. Wprawdzie prowadziłem także zajęcia z młodzieżą w trakcie studiów, jednak były to raczej krótkie obozy lub pojedyncze zajęcia i nie potrafię wyciągnąć zbyt wielu wniosków z kilku takich spotkań.

Zainteresowanie zdolnych uczniów algorytmiką nie jest trudne. Młodzi i utalentowani uczniowie są z reguły chętni do nauki. Programowanie i algorytmika stoją dodatkowo na uprzywilejowanej pozycji, bo stanowią nie tylko wyzwanie naukowe, lecz również poszerzają bardzo praktyczne umiejętności uczniów. Programowanie może być dla uczniów narzędziem, które umożliwi im realizowanie ich kreatywności. Za to algorytmika z pewnością spodoba się zarówno teoretykom, jak i praktykom, bo, szczególnie na podstawowym poziomie, pokazuje, jak w prosty sposób można rozwiązać stosunkowo złożone problemy przy użyciu komputera. Nie sądzę więc, by jakiegokolwiek wysublimowane metody były konieczne do zachęcenia uczniów do nauki informatyki. Poważniejszą kwestią wydaje się umożliwienie nauki większej liczbie uczniów. Obecnie taką szansę dostają oni tylko w nielicznych szkołach, dość powiedzieć, że każdego roku w finałach Olimpiady Informatycznej reprezentowane są szkoły z zaledwie około 13 województw, za to z wielu szkół jest po kilku finalistów. W grupie zdolnych uczniów nie powinno być więc problemów ze znalezieniem chętnych do nauki.

Zanim rozpocznie się uczenie algorytmiki, uczniowie muszą opanować umiejętność programowania.⁵ Dla utalentowanych uczniów nie powinno być to zbyt trudne. Nauka algorytmiki jest jednak trudniejsza i na niej chciałbym się skoncentrować.

Książki o tematyce algorytmicznej są pisane zazwyczaj z myślą o studentach i zawierają mnóstwo wiedzy teoretycznej. Prawdopodobnie nie trafią do wielu młodszych czytelników i mogą wręcz wytworzyć wrażenie, że cała dziedzina jest bardzo trudna. Zajęcia dla młodzieży powinny rozpocząć się od zupełnie innych tematów niż podręczniki akademickie. Może to być po prostu kontynuacja zajęć o programowaniu, na których po zapoznaniu uczniów z językiem programowania przejść można do rozwiązywania prostych problemów za pomocą komputera. Podręczniki akademickie mogą zaś służyć jako źródło encyklopedyczne.

Początkowa część zajęć będzie prawdopodobnie dla uczestników szczególnie wciągająca. Uczniowie powinni zainteresować wyjątkowo proste rozwiązania problemów, których nawet najlepsi z nich nie będą zapewne w stanie rozwiązać samodzielnie. Jako przykład mogą posłużyć: problem plecakowy oraz zadanie znajdowania fragmentu tablicy o maksymalnej sumie. Nie oznacza to, że polecam te tematy na pierwsze zajęcia z algorytmiki. Lepiej rozpocząć od metod, które wszyscy uczniowie w zasadzie znają, czyli np. dodawania pisemnego wielocyfrowych liczb albo prostych sposobów sortowania.

⁵ Nie sądzę, by uczenie algorytmiki bez wcześniejszej nauki programowania było skuteczne, bo tym sposobem pomniejszy się grupę potencjalnie zainteresowanych o tych, którzy nastawiają się na praktyczne zastosowania. Jeśli ktoś interesuje się naukami ścisłymi, to niemal na pewno umiejętność programowania będzie mu w życiu przydatna.

Dobierając tematy na zajęcia warto kierować się kryterium praktyczności omawianych metod. Do takich z pewnością należą metody przeszukiwania grafów, których liczne zastosowania można bez trudu dostrzec. Dużo trudniej jest przekonać uczniów o praktyczności stosowania wyrafinowanych struktur danych, choć mogą one być pomocne w praktycznym zrozumieniu złożoności obliczeniowej algorytmu.⁶

Podczas nauki zdolnych uczniów należy unikać podawania im gotowych rozwiązań, warto wyrabiać w nich samodzielność. Przed pokazaniem rozwiązania problemu, można dać uczniom czas na pomyślenie tak, by sami zrozumieli na czym polega trudność, a po zapoznaniu się z rozwiązaniem – by wiedzieli, w jaki sposób „zaatakować” problem. Same algorytmy również lepiej przedstawiać na przykładach, pozostawiając ich implementację jako pożyteczne ćwiczenie. W szczególności złą praktyką jest pokazywanie gotowych kodów źródłowych. Kod źródłowy to po prostu zapis metody w formalnym języku i nie ułatwia on zrozumienia zawartego w nim pomysłu.

W trakcie rozwoju młodych talentów trzeba też dbać o to, by w miarę poznawania nowych porcji wiedzy, poszerzali oni również umiejętności logicznego myślenia i samodzielnego rozwiązywania problemów. Bardzo ważną rolę odgrywa tu nauczyciel. Mimo, że w Internecie znaleźć można wiele serwisów z zadaniami programistycznymi, początkujący uczeń nie ma możliwości znalezienia czegoś na odpowiednim dla siebie poziomie. Co gorsza, zadania rozwijające pomysłowość giną wśród dziesiątek problemów, które wymagają przede wszystkim dobrych umiejętności programistycznych. Bardzo ważne więc, by nauczyciel zadbał o odpowiedni dobór zadań dla swoich podopiecznych, a najlepiej prowadził własny serwis z zadaniami.

Znalezienie chętnych na zajęcia z informatyki z reguły nie stwarza trudności, jeśli tylko takie zajęcia dobrze rozreklamujemy. Jednak w miarę upływu czasu, każdy następny etap rozwoju wymaga od ucznia dużego wysiłku i wielu godzin poświęconych na naukę. Trzeba o tym pamiętać i zwracać uwagę na motywację uczniów. Ponownie, w przypadku informatyki nie powinno to być trudne. Istnieje całe mnóstwo konkursów programistycznych i algorytmicznych, począwszy od konkursów internetowych, a skończywszy na Olimpiadzie Informatycznej, w której bodaj najważniejszą dla uczniów nagrodą, poza osobistą satysfakcją, jest wolny wstęp na dowolną uczelnię w kraju. Niektóre zawody odbywają się co kilka tygodni i osiągnięcie w nich coraz lepszych rezultatów to dobry cel dla początkujących programistów i algorytmików. Poza tym, nauka informatyki stwarza świetne szanse zawodowe, które nie są zarezerwowane jedynie dla małej grupy wybitnie uzdolnionych. Z naszego kraju każdego roku dziesiątki osób dostają się na praktyki wakacyjne do renomowanych, światowych firm. Z samego Uniwersytetu Warszawskiego na praktyki do czołowych firm w Stanach Zjednoczonych pojechało w roku 2010 ponad 20 osób. Wiele z nich to byli olimpijczycy. Takie perspektywy z pewnością działają motywująco na młodych ludzi.

4. Doskonalenie umiejętności (Marek Cygan)

Bazą do poniższych przemyśleń było przygotowywanie studentów UW do konkursów programistycznych, jak również prowadzenie kółka informatycznego w VI LO w Bydgoszczy.

Po osiągnięciu pewnego poziomu wiedzy i umiejętności programistycznych zdecydowanie zwiększają się możliwości samodzielnego rozwoju ucznia. Z uwagi na rozwój technologiczny wiedza wcześniej niedostępna, lub też trudno dostępna w postaci literatury anglojęzycznej, staje się coraz bardziej powszechna za sprawą Internetu. W sieci znaleźć można bardzo wiele serwisów z zadaniami bądź też materiałami, podzielonymi na sekcje tematyczne dotyczące poszczególnych zagadnień algorytmicznych lub ogólniej informatycznych. Należy jednak zaznaczyć, iż większość materiałów,

⁶ Uwaga o strukturach danych pochodzi od Krzysztofa Diksa.

które można znaleźć w Internecie, jest dostępna w języku angielskim. Obecnie narzędzia do automatycznego tłumaczenia, pomimo powszechnej dostępności, nie rozwiązują problemu bariery językowej, gdyż automatycznie generowane tłumaczenia nie są wystarczająco dobre przy tłumaczeniu bardziej zaawansowanych tekstów matematycznych i informatycznych. Dlatego warto zachęcić uczniów do nauki języka obcego, co z pewnością będzie przydatną umiejętnością w ich życiu, a jednocześnie umożliwi dostęp do zdecydowanie większej ilości materiałów, które mogą ich zainteresować.

Przy samodzielnej pracy ucznia ważną kwestią jest możliwość uzyskania pomocy w rozwiązywaniu zadań, których uczeń nie umie samodzielnie rozwiązać. Być może zadania te wymagają przysposobienia nieznannej partii materiału, a może uczeń nie zauważył kluczowej w danym zadaniu obserwacji. W przypadku zbioru zadań bez rozwiązań i bez podpowiedzi trudno będzie się uczniowi rozwijać. W takim wypadku ważna jest pomoc nauczyciela lub rozmowa z innymi uczniami, co zwiększa zaangażowanie grupy w dążeniu do wspólnego celu doskonalenia swoich umiejętności.

Kolejnym istotnym elementem nauki jest możliwość sprawdzenia się w konfrontacji z presją oraz innymi uczniami. Rozwiązywanie zadania w warunkach konkursowych istotnie różni się od spokojnego analizowania problemu, dlatego należy umożliwić uczniom sprawdzenie się i nabycie doświadczenia w warunkach zbliżonych do konkursowych. Może się to odbyć za pomocą uczestnictwa w zawodach internetowych bądź poprzez przygotowanie przez nauczyciela zestawu zadań, które uczniowie będą rozwiązywać na czas. Bardzo często zdarza się, że uczniowie z niewielkim (bądź z żadnym) doświadczeniem konkursowym popełniają podstawowe błędy w warunkach konkursowych – zaczynają implementację nieprzemysłanych rozwiązań, nie potrafią kontrolować czasu, przez co zamiast jednego dobrze rozwiązane zadania, kończą konkurs z dwoma prawie rozwiązanymi zadaniami, co oznacza bardzo małą (bądź też zerową) liczbę punktów.

Praca z grupą uczniów różni się od nauczania indywidualnego. W przypadku pracy z grupą uczniów warto rozważyć różne metody zachęcające uczniów do współpracy – wymiany wiedzy i doświadczeń. Jedną z takich metod jest praca w parach nad wspólnym rozwiązaniem zadania oraz jego implementacją. Jeśli uczeń musi wytłumaczyć swój nieczytelny program komputerowy, to najprawdopodobniej następnym razem będzie starał się poprawić strukturę swojego programu oraz wyeliminować złe nawyki, co pozytywnie wpłynie na jego umiejętności programistyczne. Jednakże należy dokonywać podziału na podgrupy w sposób przemyślany. Warto, aby uczeń słabszy miał możliwość współpracy z osobami dużo lepszymi, co pozwoli mu zauważyć swoje braki, jednakże częsta współpraca z uczniami dużo lepszymi może być zniechęcająca dla obu stron. Ponadto w przypadku pracy z grupą uczniów o zróżnicowanym poziomie lub też wieku należy zadbać o zróżnicowany stopień trudności materiału tak, aby mniej doświadczeni uczniowie mieli szansę zrozumieć podstawowe zagadnienia, natomiast najlepsi powinni mieć też możliwość rozwiązywania zadań dodatkowych.

5. Prosty język maszynowy pierwszym językiem programowania? (Tomasz Idziaszek)

Obecnie większość konkursów algorytmicznych wymaga od uczestników nie tylko umiejętności projektowania efektywnych algorytmów, ale również programowania. Co więcej, biegłe programowanie jest punktem wyjścia do nauki algorytmiki – o ile bowiem sprawne zapisanie mniej efektywnego algorytmu jest zwykle premiowane częściowymi punktami, to nieumiejętność zapisania optymalnego algorytmu już nie. Niestety dość często zakłada się, że skoro uczniowie potrafią napisać kompilujący się program, to jest to równoznaczne z tym, że umieją programować. Nie jest to prawdą w ogólności.

Nauczyciel chcący wykształcić umiejętność programowania u uczniów stoi przed niełatwym wyborem pierwszego języka programowania, z którym uczniowie zetkną się na jego zajęciach.

Poniżej przedstawimy wybór, jakiego dokonał Andrzej Gąsienica-Samek, kiedy prowadził olimpijskie kółko informatyczne w roku 2001, i wymienimy zalety takiego podejścia. Wybór Andrzeja był dosyć niestandardowy, gdyż językiem, którego używał na zajęciach do wprowadzenia młodych adeptów w świat informatyki, był język maszynowy.

Warto zauważyć, że takie podejście ma pewne uzasadnienie historyczne. Zanim pojawiły się kompilatory języków wyższego poziomu, programowanie siłą rzeczy musiało odbywać się w wewnętrznym języku maszyny liczącej. Maszyna taka była w gruncie rzeczy dosyć prosta, tak więc i jej język udostępniał nieduży zbiór prostych operacji. Powodowało to, że złożone obliczenia wymagały długich programów, które były trudne w utrzymaniu. Naturalna skłonność człowieka do myślenia abstrakcyjnego spowodowała opracowanie różnych technik i paradygmatów programowania, które pozwalały mu zwięźle wyrażać swoje myśli, oraz narzędzi przekształcających programy wyższego poziomu na język maszynowy. Jednak zawsze fundament, na którym opierały się konstrukcje wyższego poziomu, był niezmienny.

Język używany na kółku olimpijskim był wewnętrznym językiem prostej maszyny zaprojektowanej przez Andrzeja. Miała ona niewielką pamięć złożoną z kolejnych komórek, w których można było zapisać liczby całkowite. Kod programu był zapisany w osobnym miejscu. Maszyna udostępniała raptem pięć instrukcji (ustawianie zawartości komórek pamięci, skoki warunkowe i bezwarunkowe, operacje czytania liczb z wejścia i wypisywania liczb na wyjście). Jedynymi udogodnieniami były: możliwość pisania wyrażeń arytmetycznych w postaci znanej uczniom z lekcji matematyki (z dodatkowym operatorem dostępu do komórki pamięci) oraz możliwość definiowania nazwanych stałych (dzięki temu dostęp do zmiennych mógł odbywać się poprzez podanie ich nazwy zamiast numeru komórki pamięci).

Jedną z podstawowych zalet tego języka była jego prostota. Zdecydowana większość współczesnych języków programowania nie ma tej zalety. Chcąc napisać choćby najprostszy program w języku C lub Java, jesteśmy zmuszeni do tłumaczenia dość zawilej składni i technicznych elementów związanych z dołączaniem potrzebnych bibliotek czy nieintuicyjną obsługą wejścia/wyjścia. Nawet w przypadku języka Pascal musimy wyjaśnić znaczenie typów przy deklaracji zmiennych i niefortunne zasady używania średnika. Próba prześlizgnięcia się nad tymi problemami (mówiąc, że to są rzeczy, których nie trzeba jeszcze rozumieć) może spowodować u uczniów frustrację.

Drugą zaletą było osvajanie uczniów z architekturą komputera. Dzięki temu nie traktowali komputera jako magicznej czarnej skrzynki, ale jako narzędzie, które rozumieją, jak działa. Przy okazji pisania bardziej złożonych programów, uczniowie sami dostrzegali ograniczenia języka maszynowego i to był właściwy moment na wspólne wprowadzenie mechanizmów, które pozwalały im ułatwić sobie pracę. Jeden z takich mechanizmów, którego zrozumienie jest kluczowe w nauce programowania, to rekurencja. Wiele osób rozpoczynających naukę od języka wysokiego poziomu ma kłopot ze zrozumieniem idei rekurencji, która wydaje się im jakąś czarną magią. Na kółku uczniowie pod kierunkiem nauczyciela od podstaw budowali mechanizm stosu używanego do pamiętania informacji związanych z wywoływanyymi podprogramami. Dało im to pełne zrozumienie idei tego mechanizmu, który bez kłopotu odkrywali w językach wyższego poziomu.

Dopiero gdy uczestnicy kółka nabrali biegłości w posługiwaniu się językiem maszynowym i poznali implementację podstawowych konstrukcji programistycznych, które udostępniają języki wyższego poziomu, mogli zacząć naukę algorytmiki. Ta była już prowadzona z użyciem języka Pascal, który był jednym z dostępnych języków na Olimpiadzie Informatycznej.

Może nasunąć się pytanie, czy rzeczywiście do sprawnego programowania w języku Pascal jest potrzebna znajomość języka maszynowego? Z przykrością można stwierdzić, że spora liczba studentów informatyki ma mglistą wiedzę na temat działania komputera na poziomie niskim (procesor, programowanie w języku maszynowym), czy nawet średnim (implementacja mechanizmów



dostarczanych przez języki wysokiego poziomu, tj. odśmiecanie pamięci, mechanizmy języków obiektowych). Jednym z efektów tej niewiedzy jest pisanie programów, które są nieefektywne czasowo lub pamięciowo. Studenci bowiem wiedzą, że dany mechanizm działa, ale nie wiedzą jak, wskutek czego nie są w stanie oszacować, jak długo działa i ile pamięci komputera będzie potrzebować. W przypadku uczniów i studentów, którzy startują w konkursach programistycznych może to być poważna przeszkoda. We współczesnym komputerze istnieje bardzo wiele czynników, które mogą zaważyć na czasie wykonania programu, który skądinąd może być oparty na teoretycznie optymalnym algorytmie (wymienić można chociażby koszt różnych mechanizmów wejścia/wyjścia, dostęp do pamięci uwzględniający wpływ pamięci podręcznej i dyskowej, koszt instrukcji skoku w procesorze czy też koszt używanych funkcji bibliotecznych). Aby być w stanie w pełni zrozumieć te zagadnienia, jest potrzebny czas na ich poznanie i oswojenie się z nimi w praktyce. Uczniowie, którzy nie znają podstaw działania komputera, a rekurencję traktują jak „czarną skrzynkę”, będą mieć z tym kłopoty. Zwłaszcza uczniów, którzy w przyszłości chcieliby sięgnąć po laury w Olimpiadzie Informatycznej, czy innych konkursach programistycznych, należy od początku rozwijać w kierunku dogłębnego zrozumienia zasad działania komputera.

Materiały z kółka olimpijskiego, o którym piszemy w ostatniej części, są dostępne na stronie <http://web.archive.org>:

<http://web.archive.org/web/20020610104156/kolko.ofek.waw.pl/kasmv1/podstawy.html>

<http://web.archive.org/web/20030916184656/kolko.ofek.waw.pl/kasmv1/spec.html>

<http://web.archive.org/web/20031123071251/kolko.ofek.w>



OD SZKOLNYCH KONKURSÓW PROGRAMISTYCZNYCH DO SUKCESÓW W ZAWODZIE INFORMATYKA

Krzysztof Diks, Jan Madey

Instytut Informatyki, Uniwersytet Warszawski

{diks, madey}@mimuw.edu.pl

Streszczenie. W tym opracowaniu przedstawiamy historię sukcesów polskiej informatyki młodzieżowej w ostatnich kilkunastu latach, wskazujemy podstawy tych sukcesów oraz przedstawiamy warunki, które powinny być spełnione, żeby sukcesy młodych informatyków przekuć w sukcesy w ich życiu zawodowym, na miarę ich możliwości i ambicji. Wydaje się, że zdobyte doświadczenia są na tyle uniwersalne, że mogą być pomocne w wyławianiu uczniów uzdolnionych informatycznie oraz w pracy z nimi, w kraju i zagranicą.

1. Wstęp

W roku szkolnym 1993/1994 wystartowała Polska Olimpiada Informatyczna [8] – ogólnopolski konkurs informatyczny dla uczniów szkół średnich. Olimpiada Informatyczna stała się jedną z oficjalnych ogólnopolskich olimpiad przedmiotowych uzyskując tym samym wsparcie organizacyjne i finansowe Ministerstwa Edukacji Narodowej. We wszystkich dotychczasowych zawodach Olimpiady wzięło udział około 13000 uczniów. Od roku szkolnego 1995/1996 polscy uczniowie co roku przywożą złote medale z Międzynarodowej Olimpiady Informatycznej [5]. Największe sukcesy polskich olimpijczyków, to zwycięstwa w Międzynarodowej Olimpiadzie Informatycznej w roku 2006 (Filip Wolski pokonał 281 uczestników z 74 krajów) i w roku 2007 (Tomasz Kulczyński był najlepszy wśród 283 uczestników z 77 krajów). O tym, jak dobrzy są uczestnicy Polskiej Olimpiady Informatycznej świadczą akademickie sukcesy byłych olimpijczyków. Do największych należą zwycięstwa zespołów Uniwersytetu Warszawskiego w konkursie The ACM International Collegiate Programming Contest [3] w latach 2003 (zespół w składzie Tomasz Czajka, Andrzej Gąsienica-Samek, Krzysztof Onak zwyciężył w Hollywood, pokonując w finale 69 zespołów wyłonionych w eliminacjach spośród 3850 drużyn, pochodzących z 1329 uczelni z 68 krajów) i 2007 (zespół w składzie Marek Cygan, Marcin Pilipczuk, Filip Wolski zwyciężył w Tokio, pokonując w finale 88 zespołów wyłonionych w eliminacjach spośród 6099 drużyn, pochodzących z 1756 uczelni z 82 krajów). Polacy odnosili także sukcesy w innych światowych konkursach: TopCoder Open [9], Google Code Jam 2005 [1], Microsoft's Imagine Cup [2], czy też w The IEEE Computer Society Annual International Design Competition [10].

Konkursy pełnią rolę nie do przecenienia w wyławianiu i kształceniu uczniów i studentów wyjątkowo uzdolnionych. Wymagają one wiedzy i umiejętności wybiegających daleko poza to, co jest uczzone w szkole i na studiach. Minimalnych wymagań stawianych uczestnikom Międzynarodowej Olimpiady Informatycznej [11] nie spełniają często studenci wielu uczelni informatycznych w kraju. Dobry konkurs dotyka zawsze podstaw dziedziny, której dotyczy, a wiedza i umiejętności w nim zdobyte nie są ulotne i dają niezbędne podwaliny dalszego dziedzinowego rozwoju. Jeszcze ważniejszym jest, żeby konkurs kształcił w młodych ludziach umiejętności, które są niezbędne w ich późniejszej aktywności zawodowej: pracowitość, systematyczność, samodyscyplina, dociekliwość, samodoskonalenie, praca



w zespole, uczciwość, ambicja, chęć konkurowania, dążenie do sukcesu. Startowanie w dobrym konkursie powinno być wyzwaniem intelektualnym dla młodego człowieka, a sukces nobilitować. Z drugiej strony organizatorzy konkursów powinni zadbać o to, żeby ich uczestnicy mieli okazję poznać się i nawiązać bliskie kontakty, które później mogą zaowocować w ich życiu zawodowym. Wszystkie konkursy, o których była mowa powyżej, posiadają te cechy.

Żaden konkurs nie powinien być celem samym w sobie. Odkryte talenty trzeba szlifować, co oznacza zapewnienia uzdolnionym młodym ludziom możliwości kształcenia na najwyższym światowym poziomie, kontaktów z najlepszymi badaczami z dziedziny, staży i praktyk w czołowych firmach decydujących o rozwoju dziedziny, wyzywających zadań do rozwiązania i wreszcie zapewnienia warunków finansowych umożliwiających pełne zaangażowanie się im w pracę zawodową, która powinna być na miarę ich ambicji, wiedzy i umiejętności.

W tym artykule przedstawiamy przegląd najważniejszych konkursów informatycznych, polskich i światowych. Dokonujemy ich krótkich charakterystyk oraz przedstawiamy najważniejsze sukcesy Polaków w tych konkursach. Następnie próbujemy scharakteryzować główne czynniki, które są podstawą sukcesów młodych polskich informatyków. Na koniec zastanawiamy się, co zrobić, żeby sukcesy młodzieżowe przekuć na sukcesy zawodowe w dorosłym życiu.

2. Konkursy Informatyczne

Najważniejsze konkursy informatyczne dla uczniów i studentów, to konkursy algorytmiczno-programistyczne. W tego typu konkursach uczestnicy dostają pewną liczbę zadań do rozwiązania, z których każde składa się z krótkiej historyjki przedstawiającej sytuację problemową. Rozwiązaniem zadania jest zazwyczaj algorytm zapisany w postaci programu w wybranym przez zawodnika (algorytmicznym) języku programowania. Najczęściej używanymi językami programowania są C i C++. W ostatnich latach obserwuje się coraz mniejsze zainteresowanie językiem Pascal i wzrost popularności języka Java. Poprawnie kompilujące się programy są następnie uruchamiane na nieznanym dla zawodników testach przygotowanych przez organizatorów. Testy są tak dobrane, żeby wykrywały programy niepoprawne i różnicowały rozwiązania o różnej złożoności obliczeniowej, przy czym głównie chodzi o złożoność czasową, a złożoność pamięciowa jest wymuszana przez podane explicite ograniczenia na wielkość wykorzystywanej przez program pamięci. W zależności od konkursu ocena rozwiązania jest binarna (zaliczone/niezaliczone wszystkie testy) lub rozwiązanie uzyskuje liczbę punktów zależną od jego jakości. Żeby przybliżyć czytelnikom styl prezentacji zadań konkursowych przedstawiamy jedno z zadań Polskiej Olimpiady Informatycznej.

Zadanie KOD

Każdą permutację $A = (a_1, \dots, a_n)$ liczb $1, \dots, n$ można zakodować za pomocą ciągu $B = (b_1, \dots, b_n)$, w którym B_i jest równe liczbie wszystkich a_j takich, że $j < i$ oraz $a_j > a_i$ dla każdego $i = 1, \dots, n$.

Przykład

Kodem permutacji $A = (1, 5, 2, 6, 4, 7, 3)$ jest ciąg $B = (0, 0, 1, 0, 2, 0, 4)$.

Zadanie

Napisz program, który:

- wczytuje ze standardowego wejścia długość n i kolejne wyrazy ciągu liczb B ,
- sprawdza, czy jest on kodem jakiejś permutacji liczb $1, \dots, n$,
- jeżeli tak, to znajduje tę permutację i wypisuje ją na standardowe wyjście,
- w przeciwnym przypadku wypisuje jedno słowo NIE.

Wejście

- W pierwszym wierszu pliku jest zapisana dodatnia liczba całkowita $n \leq 30000$. Jest to liczba wyrazów ciągu B .
- W każdym z kolejnych n wierszy jest zapisana jedna liczba całkowita nieujemna nie większa niż 30000. Są to kolejne wyrazy ciągu B .

Wyjście

- W każdym z kolejnych n wierszy należy zapisać jeden wyraz permutacji A , której kodem jest dany ciąg B ,
albo
- jedno słowo NIE, jeśli ciąg B nie jest kodem żadnej permutacji.

Przykłady

Dane:	Wynik:
6	1
0	6
0	4
1	2
2	5
1	3
3	

Dane:	Wynik:
2	NIE
1	
0	

Osoby interesujące się kombinatoryką od razu zauważą, że w zadaniu KOD chodzi o odtworzenie permutacji znając jej wektor inwersji. Rozwiązanie tego zadania wymaga odkrycia wzajemnie jednoznacznego odwzorowania pomiędzy permutacjami, a wektorami inwersji. Najprostsze rozumowanie prowadzące do tego odkrycia (należy odtwarzać permutację analizując wektor inwersji od końca) daje natychmiast algorytm działający w czasie kwadratowym ze względu na n . Istnieją jednakże algorytmy działające w czasie $O(n \log n)$ i takie były najwyżej punktowane. Wymagają one jednak znajomości drzew poszukiwań binarnych lub rekurencji i technik związanych z sortowaniem kubełkowym. Trzeba jeszcze pamiętać, że uczestnik konkursu nie tylko musi ułożyć stosowny algorytm, ale także poprawnie zaprogramować go w wybranym przez siebie języku programowania. Wiemy, że nawet wytrawni programiści nie są w stanie od ręki napisać poprawnie działającego programu. W drodze do tego celu walczą najpierw z błędami syntaktycznymi (kompilacji), następnie przeprowadzają szereg testów pozwalających wyeliminować błędy logiczne, a na koniec badają jeszcze wydajność swoich rozwiązań. Taką drogę przechodzi też każdy uczestnik konkursu algorytmiczno-programistycznego. W małej skali uczestnik konkursu pokonuje etapy realizacji rzeczywistych projektów programistycznych.

Już na tym prostym przykładzie widać, że aby z sukcesami startować w konkursach algorytmiczno-programistycznych niezbędne są następujące umiejętności:

- zdolność precyzyjnej analizy zadań algorytmicznych (znajomość matematyki i logicznego rozumowania są w tym niezmiernie pomocne),

- sprawność w programowaniu w co najmniej jednym języku programowania wysokiego poziomu,
- znajomość co najmniej jednego środowiska programistycznego oraz umiejętność kompilowania, śledzenia i wykonywania programów w tym środowisku,
- znajomość podstawowych technik projektowania algorytmów i struktur danych.

Zwycięzcy prestiżowych konkursów mają opanowane te umiejętności na poziomie najlepszych zawodowców.

Scharakteryzujemy teraz pokrótce 5 najbardziej prestiżowych konkursów informatycznych dla uczniów szkół średnich i studentów oraz przedstawimy najważniejsze osiągnięcia Polaków w tych konkursach.

2.1 Olimpiady informatyczne

Olimpiada Informatyczna [8] jest konkursem przeznaczonym dla uczniów szkół średnich. Zwycięzcy (pierwsza czwórka) krajowych konkursów programistycznych reprezentują swoje kraje w zawodach Międzynarodowej Olimpiady Informatycznej (IOI) [5]. Polska Olimpiada Informatyczna składa się z trzech etapów. Pierwszy etap jest etapem szkolnym, rozgrywanym na przełomie października i listopada i gromadzi około tysiąca uczestników. Uczestnicy mają do rozwiązania zazwyczaj pięć zadań i pracują nad nimi w domu. Wyniki swojej pracy przesyłają organizatorom do oceny przez Internet. Do drugiego etapu awansuje około 360 najlepszych zawodników z etapu pierwszego. Drugi etap jest organizowany w ośmiu ośrodkach regionalnych współpracujących ściśle z najlepszymi uczelniami informatycznym w kraju i trwa trzy dni. Pierwszy dzień jest poświęcony na zapoznanie się z warunkami rozgrywania zawodów. W każdym z następujących dwóch dni uczestnicy mają do samodzielnego rozwiązania 2-3 zadania w trakcie pięciogodzinnej, kontrolowanej sesji. Rozwiązania z całej Polski są zbierane centralnie i wszystkie oceniane w takim samym środowisku i na tych samych testach. Około 80 najlepszych uczestników drugiego etapu awansuje do finału Olimpiady. Finał jest rozgrywany w jednym miejscu, w Sopocie, i trwa pięć dni. Trzy dni są przeznaczone na same zawody, a dwa dni na rekreację i turystykę. Sposób rozgrywania finału jest podobny do tego z etapu drugiego. Czterej najlepsi zawodnicy z finałów reprezentują Polskę na międzynarodowych zawodach informatycznych, w tym na Międzynarodowej Olimpiadzie Informatycznej. Międzynarodowa Olimpiada Informatyczna jest rozgrywana latem każdego roku i gromadzi najlepszych w świecie młodych informatyków – uczniów szkół średnich. Pierwsza Międzynarodowa Olimpiada Informatyczna miała miejsce w roku 1989. Od tego czasu Polacy wywalczyli 29 medali złotych, 26 medali srebrnych i 22 medale brązowe. W latach 2006 i 2007 odpowiednio Filip Wolski i Tomek Kulczyński zostali absolutnymi zwycięzcami Międzynarodowej Olimpiady Informatycznej.

2.2 Mistrzostwa świata w programowaniu zespołowym

Konkurs ICPC (The ACM International Collegiate Programming Contest) [3] jest najstarszym i najbardziej prestiżowym konkursem informatycznym w świecie i jest uznawany jako mistrzostwa świata w programowaniu zespołowym. Konkurs ten jest przeznaczony dla studentów i jest zespołowy. Każdy zespół składa się z trzech studentów reprezentujących tę samą uczelnię. Konkurs ICPC jest dwuetapowy. Etap pierwszy to eliminacje regionalne. Takich eliminacji jest kilkadziesiąt i są rozgrywane na wszystkich zamieszkałych kontynentach. Najlepsze drużyny z eliminacji (zawsze zwycięzcy oraz kilka następnych drużyn, zależnie od siły regionu i liczby startujących drużyn) awansują do finałów. Zarówno zawody eliminacyjne, jak i zawody finałowe są rozgrywane w ten sam sposób. Każda trzyosobowa

drużyna ma do dyspozycji jeden komputer, pięć godzin czasu i od 8 do 12 zadań do rozwiązania. Rozwiązania poszczególnych zadań są sprawdzane w czasie rzeczywistym, a drużyny są informowane o wyniku sprawdzenia zadania: zaakceptowane, niepoprawne wyniki, przekroczenie limitu czasu na rozwiązanie, błąd wykonania, niepoprawna prezentacja wyników. Poszczególne zadania są oceniane binarnie (zaliczone/odrzucone). Zawody wygrywa drużyna, która zaliczy najwięcej zadań. W przypadku takiej samej liczby zaliczonych zadań przez więcej niż jedną drużynę, o ich kolejności w rankingu decyduje krótszy, łączny czas przeznaczony na rozwiązanie wszystkich zaliczonych zadań, przy czym za każde wcześniej odrzucone zgłoszenie zaliczonego zadania nalicza się 20 minut kary.

Historia konkursu ICPC sięga roku 1977 i nierozzerwalnie jest związana z osobą profesora Billa Pouchera, pomysłodawcy i animatora konkursu. Początkowo w konkursie startowały tylko zespoły ze Stanów Zjednoczonych (kilkanaście). W roku akademickim 2009/2010 wystartowało 7000 drużyn z 1931 szkół wyższych z 82 krajów, spośród których do finałów zakwalifikowano 103 zespoły. Polacy, a ściślej drużyna Uniwersytetu Warszawskiego, wystartowali po raz pierwszy w konkursie ICPC w roku 1994/1995, od razu awansując do finału i zajmując w nim 11 miejsce. Od tego czasu zespoły Uniwersytetu Warszawskiego nieprzerwanie biorą udział w finałach konkursu. Tylko jedna uczelnia w świecie może się pochwalić większą liczbą startów w finałach, kanadyjski uniwersytet z Waterloo. Największe sukcesy Uniwersytetu Warszawskiego (i Polski) to zwycięstwa w finałach konkursu w latach 2003 i 2007. W roku 2003 zwyciężyła drużyna w składzie Tomasz Czajka (zwycięzca Polskiej Olimpiady Informatycznej w roku 1999/2000), Andrzej Gąsienica-Samek (zwycięzca Polskiej Olimpiady Informatycznej w latach 1994/1995, 1996/1997 i 1997/1998), Krzysztof Onak (II miejsce w Polskiej Olimpiadzie Informatycznej w latach 1997/1998 i 1999/2000). W roku 2007 akademickimi mistrzami świata w programowaniu zespołowym zostali: Marek Cygan (uczestnik II etapu Polskiej Olimpiady Informatycznej w latach 2001/2002 i 2002/2003), Marcin Pilipczuk (piąte miejsce w Polskiej Olimpiadzie Informatycznej w roku 2001/2002, trzynaste miejsce w roku 2002/2003), Filip Wolski (zwycięzca Polskiej Olimpiady Informatycznej w latach 2004/2005 i 2005/2006, zwycięzca Międzynarodowej Olimpiady Informatycznej w roku 2006). Inne znaczące osiągnięcia w tym konkursie to drugie miejsce zespołu z Uniwersytet Jagiellońskiego (Arkadiusz Pawlik, Bartłomiej Walczak, Paweł Walter) w roku 2006, piąte miejsce zespołu z Uniwersytetu Wrocławskiego w roku 2005 (Paweł Gawrychowski, Jakub Łopuszański, Tomasz Wawrzyniak), szóste, siódme, dziesiąte, dziewiąte i ósme miejsce drużyn z Uniwersytetu Warszawskiego odpowiednio w latach 2001, 2004, 2006, 2009 i 2010. Wszyscy członkowie tych drużyn brali udział w szkole średniej w Polskiej Olimpiadzie Informatycznej.

2.3 Konkurs TopCoder

Konkurs TopCoder [9] jest jednym z najmłodszych konkursów w świecie, ale niezwykle popularnym. Liczba zarejestrowanych zawodników przekracza 270000 i reprezentują oni ponad 200 krajów. TopCoder jest konkursem indywidualnym, rozgrywanym początkowo w 3 kategoriach: Algorytm, Projektowanie i Wytwarzanie. Kategoria Algorytm jest najbardziej popularna. W tej kategorii uczestnicy rozgrywają w Internecie programistyczne mecze, które mają miejsce średnio co tydzień. Każdy mecz trwa około półtorej godziny. Mecz jest podzielony na dwie fazy: fazę kodowania i fazę sprawdzania. W fazie kodowania zawodnicy mają do rozwiązania trzy zadania: łatwe, średnio trudne i trudne, odpowiednio za 250, 500 i 1000 punktów. Żeby otrzymać dodatnią liczbę punktów za zadanie musi być ono rozwiązane poprawnie. Liczba punktów za poprawnie rozwiązane zadanie zależy jednak od czasu, który został poświęcony na jego rozwiązanie. W fazie sprawdzania zawodnicy widzą programy innych uczestników meczu i mogą kwestionować ich poprawność. Za każde wskazanie niepoprawnego rozwiązania innego uczestnika meczu można otrzymać dodatkowe punkty,



natomiast błędne wskazania są karane odjęciem punktów. Zwycięzcą meczu zostaje uczestnik, który uzyska największą liczbę punktów. Na podstawie wyników dotychczasowych meczów tworzy się ranking wszystkich zawodników. Punkty rankingowe studentów są podstawą rankingu uczelni oraz krajów. Aktualnie Uniwersytet Warszawski zajmuje trzecie miejsce w światowym rankingu (będąc na czele przez ponad 4 lata), a Polska trzecie miejsce wśród krajów.

Raz do roku odbywa się turniej stacjonarny pod nazwą TopCoder Open poprzedzony kilkoma rundami eliminacji. Do turnieju stacjonarnego awansuje od 24 do 48 zawodników. Zwycięzca turnieju TopCoder Open to jak mistrz Wimbledonu w tenisie. Polacy trzykrotnie zwyciężali w finałach TopCoder Open w latach 2003 – 2005, dwa razy Tomek Czajka i raz Eryk Kopczyński.

2.4 Imagine Cup

Imagine Cup [2] jest konkursem organizowanym przez firmę Microsoft. Jest on adresowany do studentów. Imagine Cup jest rozgrywany w różnych kategoriach, które zmieniają się z roku na rok. Konkurs jest skierowany zarówno do twórców narzędzi technologii informacyjnych, jak i użytkowników tych narzędzi. Celem konkursu jest wyzwolenie kreatywności jego uczestników. Polska także w tym konkursie odnosi spektakularne sukcesy. Na przykład w roku 2007 Polacy wygrali w kategoriach Algorytm (Przemysław Dębiak), Fotografia (Iwona Bielecka, Małgorzata Łopaciuk), Krótki film (Julia Górniewicz, Jacek Barcikowski).

2.5 Projektowanie oprogramowania

Poszukiwaniem innej niż ICPC formuły współzawodnictwa informatycznych zespołów uczelnianych zajęło się kilka lat temu towarzystwo naukowe IEEE Computer Society i w rezultacie w 2000 roku został powołany konkurs CSIDC (Computer Society International Design Competition) na projekt programistyczny [10]. W ramach konkursu należy zaprojektować i wykonać w ramach zajęć studenckich urządzenia sterowane komputerowo. Uczestniczące uczelnie są losowane spośród chętnych (aby z góry była określona liczba uczestników, zazwyczaj około 200 na cały świat) i otrzymują od organizatorów odpowiednie zestawy sprzętowe oraz pakiety programistyczne.

Każda z uczelni przedstawia po kilku miesiącach raport, oceniany przez ekspertów z przemysłu oraz akademii. Dziesięć najlepszych projektów jest kwalifikowanych do finałów. W finałach każdy zespół ma szansę na żywo zaprezentować wyniki swojej pracy. Ekspertki po dyskusji wyłaniają zwycięzcę.

Okazuje się, że i w tym konkursie polscy informatycy odnoszą duże sukcesy. Zespoły Politechniki Poznańskiej sześciokrotnie awansowały do finałów, zwyciężając w drugiej i czwartej edycji CSIDC (lata 2001, 2004), zajmując drugie miejsce w latach 2002 i 2005 oraz trzecie miejsce w roku 2000. Niestety z powodów finansowych ten konkurs nie był rozgrywany w latach 2007 – 2009. W roku 2010 odbyła się kolejna jego edycja.

3. Anatomia sukcesu

W tym rozdziale próbujemy odpowiedzieć na pytanie, co jest tajemnicą sukcesów młodych polskich informatyków. Niewątpliwie liczne sukcesy młodzieży, potwierdzane co roku, nie są przypadkowe. Nie mogły też się wydarzyć z dnia na dzień. Wydaje się, że najważniejszym, długofalowym przedsięwzięciem, które miało olbrzymi wpływ na rozwój młodzieżowej informatyki w Polsce było powołanie w roku 1993 Olimpiady Informatycznej. Olimpiada to instytucja, która w sposób zorganizowany, przy współpracy Ministerstwa Edukacji Narodowej, najlepszych uczelni w kraju (Uniwersytet Warszawski, Uniwersytet Jagielloński, Uniwersytet Wrocławski, Uniwersytet im. Mikołaja



Kopernika w Toruniu, Politechnika Poznańska, Akademia Górniczo-Hutnicza, Politechnika Śląska), nauczycieli i przemysłu IT zajmuje się wyławianiem uczniów utalentowanych informatycznie i rozwojem ich talentów. Cele te są realizowane poprzez umożliwienie uczniom szlachetnej rywalizacji w rozwiązywaniu niebanalnych zadań informatycznych. Zadania są przygotowywane zarówno przez naukowców-informatyków o światowej renomie, jak i byłych uczestników Olimpiady, osiągających sukcesy w konkursach studenckich. Co więcej, byli olimpijczycy aktywnie uczestniczą w pracach Olimpiady przygotowując rozwiązania wzorcowe zadań olimpijskich i są autorami wyrafinowanego oprogramowania olimpijskiego służącego do automatyzacji prac w Olimpiadzie, w szczególności automatycznego sprawdzania rozwiązań zawodników.

Olimpiada prowadzi intensywną działalność edukacyjną. Co roku są wydawane materiały popolimpijskie zawierające szczegółową dyskusję rozwiązań zadań oraz wzorcowe programy. Byli olimpijczycy prowadzą portal edukacyjny dla początkujących adeptów programowania i algorytmiki [6], dzięki któremu nawet uczniowie z małych miasteczek i wiosek mają szansę poznawać tajniki „prawdziwej” informatyki. Finaliści Olimpiady mają co roku okazję uczestniczyć w wakacyjnych obozach wypoczynkowo-naukowych, na których wysłuchują wykładów przygotowywanych przez pracowników naukowych i starszych kolegów. Mogą też doskonalić swoje umiejętności algorytmiczno-programistyczne biorąc udział w praktycznych warsztatach programistycznych.

Ważną rolę w rozwoju najlepszych odgrywa też Krajowy Fundusz na rzecz Dzieci [4] – organizacja, która statutowo zajmuje się młodzieżą szczególnie uzdolnioną (nie tylko informatycznie). Na warsztatach i obozach Funduszu uczniowie mają szansę poznawać dziedziny informatyki, z którymi niekoniecznie muszą zetknąć się startując w konkursach.

Współpracownicy Olimpiady są często autorami lub tłumaczami najważniejszych podręczników informatycznych, które umożliwiają naukę młodym ludziom w języku ojczystym.

Pierwszą osobą, która może dostrzec talent ucznia i skierować go na właściwe drogi rozwoju jest jego nauczyciel. Dlatego w ramach Olimpiady są organizowane warsztaty olimpijskie dla nauczycieli, na których mogą oni zapoznać się praktycznie ze specyfiką konkursów informatycznych.

Tak szeroka działalność nie byłaby możliwa bez wsparcia finansowego. Działalność Olimpiady Informatycznej jest finansowana zarówno ze środków publicznych (dotacja Ministerstwa Edukacji Narodowej), jak i środków prywatnych pochodzących od czołowych firm informatycznych w Polsce. Tutaj na szczególne wyróżnienie zasługuje integrator systemów informatycznych firma PROKOM SOFTWARE S.A., która współorganizowała Olimpiadę w latach 1997–2006. W roku 2007 rolę biznesowego współorganizatora Olimpiady przejęła firma Asseco.

Nie mniej ważnym jest to, że najlepsi olimpijczycy, zwycięzcy konkursów programistycznych, tworzą elitę i są wzorcem dla następnych pokoleń. Bycie w elicie nobilituje. Sprzyja temu też polityka czołowych uczelni w kraju, które przyjmują finalistów Olimpiady na studia bez postępowania kwalifikacyjnego. Tylko Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego przyjmuje corocznie około 50 olimpijczyków, którzy należą do najlepszych studentów. Ta grupa pozwala prowadzić zajęcia na najwyższym poziomie i stawiać studentom bardzo wysokie wymagania.

Podsumowując kluczami do sukcesy są więc:

- pasjonaci (naukowcy, nauczyciele, studenci, uczniowie),
- wspierające instytucje (uczelnie wyższe, Ministerstwo Edukacji Narodowej, szkoły, Ośrodek Edukacji i Zastosowań Komputerów, Ogólnopolska Fundacja Edukacji Komputerowej, Krajowy Fundusz na rzecz Dzieci),
- bezpieczeństwo finansowe (MEN, Asseco, ATM),
- wysoka jakość (zarówno jakość przygotowywanych zadań, jak i jakość organizacyjna).



4. Co dalej?

Żeby osiągnąć sukces na poziomie szkoły średniej i uczelni wystarczy grono zaangażowanych osób, dobra organizacja, wsparcie urzędowe i finansowe. Najlepsi młodzi ludzie trafiają na czołowe uczelnie w kraju, które gwarantują bardzo wysoki poziom ich wykształcenia. Osiągnięcia w konkursach gwarantują wsparcie finansowe (stypendia) oraz możliwość odbycia staży w najlepszych firmach informatycznych świata. Corocznie kilkudziesięciu studentów Uniwersytetu Warszawskiego odbywa staże w największych i najlepszych firmach takich, jak Google, Nvidia, Microsoft, IBM. Jeszcze na studiach znajdują bez problemu zatrudnienie. Powstaje tylko pytanie, czy to jest praca na miarę ich możliwości i ambicji. Czasami tak. Nie jest tajemnicą, że jednym z argumentów, które stanęły za tym, że Google otworzył centrum badawczo-rozwojowe w Polsce były sukcesy Polaków w konkursach programistycznych polskich studentów. Reguła 20% polegająca na możliwości poświęcenia 20% czasu pracy na realizację własnych pomysłów jest dobrym sposobem na wyzwolenie kreatywności i spożytkowaniu wiedzy i umiejętności utalentowanych młodych ludzi. Należy zachęcać wszystkie wielkie firmy, żeby poszły tym śladem i zaufali wiedzy i umiejętnościom talentów, z pożytkiem dla ich własnego i ogólnego rozwoju.

Naturalnym miejscem zatrudniania najzdolniejszych są uczelnie. Na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego pracuje wielu byłych olimpijczyków. Są oni autorami wybitnych wyników naukowych. Dla przykładu praca byłych olimpijczyków Marcina Muchy i Piotra Sankowskiego o obliczaniu najliczniejszego skojarzenia [7] zawierała wyniki przełomowe w tej dziedzinie. Takich prac mogłoby być znacznie więcej. Barię jednak są warunki finansowe. **Płaca oferowana przez uczelnie nie pozwala skoncentrować się młodym badaczom w pełni na nauce.** Jest wyzwaniem dla wielu krajów, w szczególności nowych krajów Unii Europejskiej, żeby zapewnić warunki finansowe i organizacyjne na „dorosły” rozwój zwycięzców konkursów i olimpiad. Ich „młodzieżowe” wyniki udowodniły, że dobra organizacja i wsparcie finansowe przynosi sukcesy rozpoznawane w świecie. Zainwestowanie w ich dorosłe życie przyniesie sukcesy, które przeniosą Polskę do czołówki krajów ery IT.

Odnośniki

1. Google Code Jam 2005, <http://code.google.com/codejam>
2. Imagine Cup, <http://imaginecup.com/>
3. International Collegiate Programming Contest, <http://icpc.baylor.edu/icpc/>
4. Krajowy Fundusz na rzecz Dzieci, <http://www.fundusz.org/>
5. Międzynarodowa Olimpiada Informatyczna, <http://www.ioinformatics.org/>
6. Młodzieżowa Akademia Informatyczna, <http://www.main.edu.pl>
7. Mucha M., Sankowski P., Maximum Matchings via Gaussian Elimination, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2004.
8. Polska Olimpiada Informatyczna, <http://www.oi.edu.pl/>
9. TopCoder, <http://www.topcoder.com>
10. The IEEE Computer Society Annual International Design Competition, <http://www.computer.org/portal/web/education/Competitions-and-Awards>
11. Verhoeff T., Horváth G., Diks K., Cormack G., A Proposal for an IOI Syllabus, *Teaching Mathematics and Computer Science* Vol. IV, Issue 1, str. 193-216.



O ROZWIJANIU ZDOLNOŚCI INFORMATYCZNYCH UCZNIÓW – NA PODSTAWIE DOŚWIADCZEŃ W PRACY Z UCZNIAMI W LICEUM AKADEMICKIM W TORUNIU

Anna Beata Kwiatkowska

ZS UMK Gimnazjum i Liceum Akademickie, Toruń
Zakład Metodyki Nauczania Informatyki i Technologii Informatycznej, WMil, UMK
aba@mat.uni.torun.pl

Szkoła

Gimnazjum i Liceum Akademickie¹ rozpoczęło swoją działalność w 1998 roku jako eksperyment kontynuujący przejęte przez Uniwersytet Mikołaja Kopernika wileńskie tradycje tworzenia szkół akademickich. Ma charakter ponadregionalny i jest przeznaczony dla młodzieży szczególnie uzdolnionej. Szkoła utworzona jako pięcioletnie gimnazjum połączone ze szkołą średnią, w którym nauka była wieńczona egzaminem maturalnym, po zakończeniu pięcioletniego eksperymentu została przekształcona w zespół szkół, który stanowią trzyletnie gimnazjum i trzyletnie liceum.

Gdy w trakcie tworzenia szkoły za namową wspianego metodyka nauczania informatyki, doktora Bolesława Wojdyły przystąpiłam do rozmów dotyczących podjęcia pracy w szkole z ówczesnym jej pierwszym dyrektorem doktorem Jerzym Wieczorkiem, miałam ze sobą багаż wielu lat doświadczeń wynikających z pracy w charakterze nauczyciela informatyki, opiekuna olimpijczyków, w najlepszym w regionie IV Liceum Ogólnokształcącym im. Tadeusza Kościuszki², które nie do końca słusznie, głównie za sprawą wybitnych osiągnięć olimpijskich uczniów, miało opinię szkoły kujonów. Usłyszałam wtedy bardzo wyraźnie, że w eksperymencie nie jest celem trenowanie uczniów do olimpiad przedmiotowych lecz przede wszystkim diagnozowanie i rozwijanie zdolności uczniów oraz wspomaganie ich twórczości. Nieco zdziwiona, widząc, że tylko wysokie wyniki w olimpiadach przedmiotowych pozwalają zauważyć szkołę, przysporzyć kandydatów i przez to utrzymać wysoką pozycję, dość sceptycznie podeszłam do takiej koncepcji. Słowa te jednak bardzo mocno zaciążyły na sposobie mojej pracy z uczniami, a później również ze studentami i po latach muszę przyznać, że te pierwsze obawy nie były uzasadnione. Dziś jestem zdecydowanym przeciwnikiem postawienia w nauczaniu uczniów zdolnych z informatyki tylko na specjalne trenowanie ich do udziału w Olimpiadzie Informatycznej. Uważam, że drogę do sukcesu szkoły³ i sukcesu uczniów można przemierzać inaczej, kierując się bardziej humanistycznymi metodami związanymi z daleko posuniętą indywidualizacją pracy, z większą korzyścią dla uczniów i ich przyszłych wyborów. Z racji swojej profesji skupię się w poniższych rozważaniach na przedmiocie informatyka, ale uważam że

¹ Strona WWW Gimnazjum i Liceum Akademickiego dostępna jest pod adresem <http://www.gimakad.torun.pl>.

² IV LO im. Tadeusza Kościuszki w Toruniu zajęło w roku 1997 drugie miejsce w kraju w Rankingu Szkół Laureatów Olimpiad Przedmiotowych UNESCO i obecnie zajmuje wysokie pozycje w rankingu Rzeczypospolitej i Perspektyw. Strona liceum jest dostępna pod adresem <http://www.loiv.torun.pl>.

³ W roku szkolnym 2010/2011 Liceum Akademickie zajęło pierwsze miejsce w rankingu szkół prowadzonym przez Rzeczypospolitą i Perspektywy.



podobnie można rozważać inne przedmioty. Nie bez znaczenia jest również moje przekonanie o tym, że spotkanie z informatyką od najmłodszych lat ma duży i pozytywny wpływ na sukcesy uczniów w innych dziedzinach, nie wykluczając dziedzin humanistycznych.

Klasy informatyczne

Opisywana powyżej koncepcja pracy szkoły, polegająca na braku nastawienia na trenowanie do olimpiad grupy uczniów uzdolnionych, przyczyniła się do tego, że początkowo w Gimnazjum i Liceum Akademickim nie były tworzone klasy o konkretnych profilach. Uczniowie uczęszczali do klas ogólnych, wybierając dodatkowe zajęcia fakultatywne rozszerzające ich wiedzę i umiejętności w danej dziedzinie, rozwijające ich zainteresowania. Mieli również możliwość zmiany tych zainteresowań wybierając każdego roku dwie z całego spektrum dziedzin nauki. Po zakończeniu eksperymentu i restrukturyzacji szkoły, pozostawiono taką organizację pracy w gimnazjum, jednak w liceum zdecydowano się na utworzenie klas profilowanych, ze względu na bardziej czytelny odbiór społeczny oferty szkoły. Nie zmieniono jednak założonej koncepcji pracy z uczniami. W ten sposób powstały m.in. klasy o profilu matematyczno-informatycznym, których założenia miałam ogromną przyjemność tworzyć, a teraz mogę realizować. W klasach tych lekcje informatyki prowadzone są w wymiarze 2 godzin tygodniowo i dodatkową ofertą dla uczniów jest seminarium informatyczne w tym samym wymiarze. SeminaRIA są podzielone na trzy rodzaje: rozszerzające wiadomości i umiejętności z lekcji, przygotowujące do olimpiady i przygotowujące do egzaminu maturalnego i mogą w nich uczestniczyć uczniowie z każdej klasy, niezależnie od rocznika. Doświadczenie wielu lat pracy mówi mi, że są to zupełnie różne grupy uczniów i uważam, że tak przyjęty podział seminariów to pierwszy krok do indywidualizacji pracy z uczniem. Krótko opiszę te formy zajęć.

Lekcje informatyki

Uważam, że istotnym elementem w nauczaniu informatyki jest jak najwcześniejsze uczenie myślenia algorytmicznego. Potwierdzają to najwyższe wyniki w Olimpiadach tych liceów, przy których istnieją gimnazja z rozszerzoną informatyką⁴ i ich sukcesy nie dotyczą tylko Olimpiady Informatycznej lecz zawodów o różnej tematyce, również tych z grupy humanistycznej. Jeśli jednak szkoła nie ma gimnazjum, opisany niżej schemat postępowania można przyjąć w klasie pierwszej liceum. Pierwszym krokiem jest dyskusja nad możliwymi algorytmami rozwiązywania problemów i zapisywanie wybranych algorytmów w postaci schematów blokowych. Omawiane problemy powinny być tak dobrane, aby w schemacie nie pojawił się zapis bardziej złożony niż jeden poziom pętli z blokami warunkowymi. Zbyt skomplikowane schematy zazwyczaj nudzą uczniów, a część z nich gubi się podczas ich rysowania. Lepiej przesunąć takie problemy na późniejszą lekcję, podczas których możemy zapisać rozwiązanie w postaci w kodu formalnego języka programowania. Takie postępowanie jest niezwykle istotne – uniezależniamy się od stopnia znajomości przez uczniów formalnego języka programowania i bardzo szybko zaczynają oni rozumieć powtarzalność i podejmowanie decyzji w algorytmach. To dobry moment na ocenę możliwości uczniów i ich predyspozycji. Istotnym jest również sposób wprowadzania języka programowania. Zdecydowanie uważam, że jest to jedynie narzędzie do zapisania rozwiązań i jego niezbędne elementy powinno się wprowadzać przy okazji omawiania kolejnych problemów i zapisywania ich algorytmów. Uznając walory dydaktyczne

⁴ Przykładem liceum, którego uczniowie mają okazję uczyć się informatyki na poziomie rozszerzonym już w gimnazjum jest obok GiLA, VI LO z Zespołu Szkół Ogólnokształcących nr 6 w Bydgoszczy, którego wychowankiem jest m.in. Tomek Kulczyński, <http://vilo.bydgoszcz.pl/>.

języka Pascal, uważam jednak, że optymalnym wyborem do zapisu algorytmów podczas lekcji jest język programowania C++, ze względu na jego popularność, obiektowość i korzenie w języku C.

Ważny jest również sposób przedstawienia innych zagadnień z programu nauczania informatyki. Na przykład istnieje pewna nonszalancja uczniów, ale też i dorosłych informatyków, w odniesieniu do zagadnień związanych z aplikacjami biurowymi, szczególnie edytorami tekstów. Wśród uczniów klas informatycznych panuje przekonanie, że dokument w komputerze jest w stanie napisać każdy i poświęcanie na to lekcji informatyki jest stratą czasu. Staram się przekonać moich uczniów, że dobry styl tworzenia dokumentów jest jednym z elementów ogólnego wykształcenia, podobnie jak np. prawidłowe budowanie zdań w języku polskim. W tym duchu, na lekcjach mówię o uniwersalnych zasadach tworzenia dokumentów, a znajomość funkcji programu ograniczam do tych niezbędnych, ale dobrze organizujących pracę z dokumentem. Takie podejście powoduje, że uczniowie poważnie traktują lekcje dotyczące edytorów tekstu. Trudno opisać w tej pracy sposób postępowania dla wszystkich zagadnień podstawy programowej, być może staną się one treścią szerszego opracowania.

Pomocniczo, do pracy podczas lekcji doradzam książki [10] i [11] oraz podręczniki [2] i [4]. Pierwsze dwie książki polecam ze względu na bogaty i przemyślany zestaw omawianych algorytmów, trzecią – ze względu na bardzo przystępny sposób wprowadzania języka programowania w języku C++, a czwartą – ponieważ wplecione w jej treść zadania maturalne pokazują, że omawiane na lekcji zagadnienia, również z technologii informacyjnej, przydają się do rozwiązywania tych zadań i zachęcają do wybierania informatyki jako przedmiotu maturalnego.

Rozszerzanie wiadomości i umiejętności z lekcji

To seminarium ma na celu rozszerzenie wiadomości dotyczących algorytmiki i programowania z lekcji. Wszystkich uczniów klas informatycznych, a często uczniów z innych klas, bardzo ciekawia zagadnienia dotyczące teorii grafów, elementów kombinatoryki i algorytmy graficzne. Do przygotowania tych zajęć polecam opracowania [1], [6] i [9]. Podczas seminarium rozwiązujemy wspólnie łatwiejsze zadania olimpijskie, a do zaprogramowania ich rozwiązań są wprowadzane elementy standardowej biblioteki C++ STL, które znacznie ułatwiają zapis najbardziej złożonych struktur danych i umożliwiają korzystanie z wielu gotowych funkcji. Nie jest poruszana tu tematyka sieci komputerowych, gdyż rozszerzenie zagadnień sieciowych jest realizowane w ścieżce indywidualnej, dzięki istniejącej w szkole Lokalnej Akademii Cisco⁵.

Przygotowanie uczniów do olimpiady

Przed powstaniem klas informatycznych szkoła miała pojedyncze sukcesy w Olimpiadzie Informatycznej. Przygotowanie do tej olimpiady wymaga spędzenia dużo czasu przy komputerze i rozważania zagadnień, które dla uczniów o innych zainteresowaniach są zbyt abstrakcyjne. Często prowadzi to do czasowego wyłączenia ze społeczności uczniów – uczeń przygotowujący się do zawodów zostaje sam w swojej trudnej pracy, co nie sprzyja osiągnięciu wysokich wyników. Dopiero utworzenie

⁵ Gimnazjum i Liceum Akademickie jest jedną z Lokalnych Akademii Cisco Akademii Regionalnej, funkcjonującej przy Regionalnym Studium Edukacji Informatycznej WMił UMK. Jest to jedyna szkoła w Polsce, w której uczniowie za darmo mogą korzystać z materiałów szkoleniowych Akademii i otrzymują certyfikat ukończenia poszczególnych semestrów nauki, których tematyka dotyczy funkcjonowania sieci komputerowych. W szkole nie są prowadzone regularne zajęcia stacjonarne Akademii. Do programu Cisco uczniowie przystępują indywidualnie i w wolnych chwilach samodzielnie przygotowują się do egzaminów. Egzaminy są zdawane w obecności nauczyciela podczas lekcji informatyki lub seminarium. Program Akademii jest dostępny pod adresem <http://cisco.netacad.net>.



klas informatycznych pokazało, że do odnoszenia sukcesów w tej dziedzinie, oprócz zajęć w szkole i pracy indywidualnej, jest potrzebna odpowiednia atmosfera, którą w naturalny sposób wytwarzają między sobą przede wszystkim uczniowie zainteresowani informatyką. Nie do przecenienia są ich dyskusje dotyczące pomysłów na rozwiązywanie problemów, wzajemne uczenie się. Istotnym czynnikiem w dobrym przygotowaniu do olimpiady jest też wprowadzanie elementów rywalizacji i dobrze jest wskazać miejsca, gdzie mogą podjąć taką rywalizację. Polecam stronę Młodzieżowej Akademii Informatycznej⁶, gdzie oprócz poznania wielu cennych rad mogą przesyłać swoje rozwiązania zadań z minionych olimpiad informatycznych i sprawdzać liczbę punktów, jaką otrzymaliby na olimpiadzie. Bardzo cennym źródłem wiedzy są popularnie zwane niebieskie książeczki [5], zawierające m.in. treści zadań olimpijskich i wzorcowe opracowania ich rozwiązań, choć ostrzegam, że ta lektura dla początkujących olimpijczyków może być zbyt trudna i trochę ich zniechęcać. Dlatego zadania powinny być umiejętnie dobierane na zajęcia, a ich rozwiązania powinny być omawiana wspólnie z nauczycielem podczas seminariów. Literaturę obowiązkową dla olimpijczyków powinny stanowić również książki [3], [8] i [9].

Przygotowanie uczniów do matury z informatyki

Przystąpienie do matury z informatyki nie wymaga specjalnych zdolności informatycznych. Często bardzo dobrze zdają maturę uczniowie, którzy nie osiągnęli sukcesów w olimpiadzie czy konkursach. Wystarczy, że regularnie uczestniczyli w lekcjach i seminariach informatycznych, na których rozwiązywane są zadania maturalne z ubiegłych lat⁷. Poruszane na maturze problemy algorytmiczne są standardowe, algorytmy zazwyczaj nie muszą być optymalne, równie nieskomplikowane są zadania dotyczące obliczeń w arkuszu kalkulacyjnym i projektowania bazy danych. Matura ma jednak jedną wspólną cechę z Olimpiadą Informatyczną, która ma bardzo duży wpływ na jej wynik. Oprócz wiedzy, jak rozwiązać dany problem uczeń musi biegle zapisywać programy w wybranym języku programowania i swobodnie pracować z daną aplikacją. Krótki czas na rozwiązanie zadań sprawia, że uczeń nie wykazujący tych umiejętności, nawet znając rozwiązanie problemu, nie zdąży wygenerować odpowiedniego programu, zaprojektować bazy danych lub arkusza. Nie bez znaczenia jest też duży stres towarzyszący uczniom podczas egzaminu, który nie jest sprzymierzeńcem poszukiwań i testowania poszczególnych funkcji już na samej maturze. Uczniowie tacy, pomimo wysokich ocen na lekcjach, podczas których jest znacznie więcej czasu na rozwiązanie problemu, osiągają wyniki maturalne w granicach około 75%. Takie niedokładne przygotowanie do matury, często polegające na nie utrwaleniu składni języka programowania, czy czynności rutynowych w aplikacji, jest częstym i nie uświadamianym sobie błędem nauczycieli i uczniów.

Wspomaganie zajęć informatycznych pracą na platformie elektronicznej

Materiały do wszystkich zajęć informatycznych w Liceum Akademickim są dostępne dla uczniów na platformie elektronicznego wspomaganie Moodle⁸. Za pośrednictwem Moodle są również zbierane rozwiązania uczniów i przesyłane informacje o uzyskanej ocenie wraz z komentarzem. Taka organizacja pracy bardzo dobrze rozwiązuje sytuację, gdy uczeń z różnych powodów, takich jak na przy-

⁶ Strona Młodzieżowej Akademii Informatycznej jest wyposażona w „sprawdzączkę olimpijską”. zawiera również opracowania wybranych zagadnień informatycznych, wprowadzenie do języków programowania na bardzo przystępnym dla ucznia poziomie. Jest dostępna pod adresem <http://www.main.edu.pl>,

⁷ Treści zadań maturalnych z poprzednich lat wraz z kluczami rozwiązań są dostępne na stronie Centralnej Komisji Egzaminacyjnej, <http://www.cke.edu.pl>.

⁸ Moodle GiLA jest dostępny pod adresem <http://moodle.umk.pl/GiLA>.

kład wyjazd na zawody lub choroba, nie może być obecny na zajęciach. Zawsze może zapoznać się z materiałem omawianym na lekcji i rozwiązać zadania indywidualnie. Platforma dobrze służy również do zorganizowania ścieżki indywidualnej uczniów i kontaktu z nimi za pośrednictwem forum lub czata. Ponieważ wykorzystujemy tylko podstawowe funkcje platformy, które ma każdy rodzaj tego typu oprogramowania, praktycznie nie ma znaczenia, którą platformę szkoła wybierze, ale zachęcam do stosowania tego typu rozwiązania.

Zdolności informatyczne uczniów

Niewielkie grupy uczniów (do 12 w grupie) pozwalają bardzo dokładnie zdiagnozować zdolności informatyczne ucznia i praca z tymi uczniami nauczyła mnie, że w tak dobranej grupie uczniów można wyróżnić wiele rodzajów zdolności informatycznych, których nie można zgubić błędnym podejściem uznawania zdolności informatycznych jedynie jako zdolności do algorytmicznego myślenia i programowania, choć trzeba te zdolności uznać za ważne i znaczące dla dalszych sukcesów.

Mimo podziału na grupy seminaryjne, każdego ucznia trzeba dokładnie poznać i traktować indywidualnie ze względu na jego zdolności. Poniżej odwołuję się do konkretnych postaci uczniów, których postawy powtarzają się wśród uczniów klas informatycznych.

Uczeń olimpijczyk

Michał przyszedł do pierwszej klasy licealnej z innej szkoły mając już na swoim koncie tytuł finalisty Olimpiady Informatycznej Gimnazjalistów. Był uczniem cichym i zamyślonym. Szybko można było zauważyć, że nudziła go regularna tematyka lekcji i potrzebuje indywidualnego traktowania. Był jedyną osobą w swojej klasie o tak wysoko zaawansowanym stopniu znajomości algorytmiki i programowania. Przygotowywałam dla niego dodatkowe zadania na poziomie olimpijskim do rozwiązania na lekcji i seminarium. W pierwszej klasie został finalistą OI zajmując wysoka pozycję na liście. Zyskał tym duży szacunek u swoich rówieśników, ale jednocześnie odrzucenie z powodu jego skrytości i osiągnięcia niedoścignionego dla innych sukcesu. Nie udało się również mnie zostać jego mistrzem, a jako nauczyciel informatyki straciłam z nim kontakt. W drugiej klasie Michał wybrał drogę samodzielnego uczenia się informatyki, przeszedł do klasy matematyczno-fizycznej, co uznałam za moją wielką porażkę. Uznał, że umie już z informatyki wystarczająco dużo i sam może doprowadzić do tego, by co najmniej powtórzyć sukces w drugiej klasie. Zajął się ekonomią i w kilka miesięcy pochłonął niezliczoną liczbę książek z tej dziedziny, co zaowocowało tytułem laureata Olimpiady Ekonomicznej. Jednak w Olimpiadzie Informatycznej nie udało mu się dotrzeć do finału. Była to dla niego wielka lekcja pokory – odwracał wzrok spotykając mnie na korytarzu szkoły. Mimo odrzucenia przez niego moich metod nauczania, postanowiłam nie poddawać się. Przy każdej okazji podkreślałam jego sukces z ekonomii i zaproponowałam opiekę nad jego uczestnictwem w wybranych zajęciach informatycznych na UMK. Z niedowierzaniem, że mogę chcieć dalej z nim współpracować, przyjął moją propozycję, zaczął spotykać się też ze swoimi rówieśnikami i ze mną podczas seminariów informatycznych, dzielił się z nami swoją wiedzą informatyczną. Uznawany przez pozostałych nauczycieli za osobę, z którą trudno rozmawiać, na informatyce stworzył się i stał się wspianym kolegą i uczniem. Jest teraz w trzeciej klasie i już przeszedł do finału OI. Wierzę, że zostanie jej laureatem.

Piotr jest uczniem naszej szkoły od gimnazjum. Już w pierwszej klasie gimnazjalnej zabłysnął znajomością języka programowania i szybkiego kodowania algorytmów. Niestety, pisane przez niego programy świadczyły o braku samorodnych zdolności do wymyślania algorytmów optymalnych. Był jednak bardzo ambitny i konsekwentny w rozwiązywaniu zadań z minionych olimpiad i znał wiele olimpijskich sztuczek algorytmicznych. Miałam wrażenie, że zna wszystkie rozwiązania zadań z nie-



bieskich książeczek. Jego pracowitość doprowadziła go do tytułu laureata zarówno OIG jak i OI, ale nigdy nie udało mu się uplasować na bezwzględnie najwyższych pozycjach. Sytuacji nie poprawiło również jego uczestnictwo w zajęciach uniwersyteckich dotyczących algorytmiki. W trzeciej klasie, gdy swą pracowitość skierował na zbyt wiele olimpiad, nie przeszedł w żadnej z nich do finału.

Uczeń sieciowiec

Marcin ma dużą łatwość w pisaniu programów na lekcji i seminarium. Myślę, że nie sprawiało by mu większej trudności rozwiązywanie zadań olimpijskich, jeśli tylko zechciałby poświęcić im więcej czasu. Marcin zachęcony przeze mnie tylko raz w pierwszej klasie wziął udział w OI i niestety nie przeszedł do drugiego etapu. Szybko wycofał się z tej drogi i nie był nigdy do niej przeze mnie zmuszany. Zauważalne było jego zainteresowanie sieciami komputerowymi. Dlatego zaproponowałam mu indywidualną ścieżkę w tym, zakresie. Już w pierwszej klasie liceum z powodzeniem ukończył dwa semestry Lokalnej Akademii Cisco. W drugiej klasie zaliczył ćwiczenia i zdał egzamin z przedmiotu Sieci komputerowe na drugim roku informatyki WMil UMK. Zajął pierwsze miejsce w Ogólnopolskim Konkursie InterSieć⁹ uzyskując tym samym indeks na studia informatyczne. Stworzył w klasie grupę uczniów o podobnych zainteresowaniach, która w jednym zespole z powodzeniem już po raz drugi startuje w Olimpiadzie Informatycznej DialNet Masters¹⁰.

Uczeń narzędziowiec

Sylwia była w trudnej sytuacji – jako dobry matematyk przyszła do klasy pierwszej liceum z niewielką znajomością języka programowania, w stosunku do pozostałych osób w klasie. Myślała, że w klasie informatycznej bardziej skupimy się na technologii informatycznej, bo takie wyobrażenie o informatyce miała z gimnazjum. Obnażyły to już pierwsze lekcje informatyki i widziałam, że czuje się z tym niekomfortowo. Postanowiłam jednak nie stracić jej dla informatyki. Szybko okazało się, że dziewczyna ma bardzo duże zdolności algorytmicznego myślenia. W tworzeniu schematów blokowych algorytmów była w klasie jedną z lepszych uczennic. Staralam się mocno podkreślać to, często eksponując jej rozwiązania na tablicy. To zmotywowało Sylwię do dodatkowej pracy i pisania programów. Poczula, że ma we mnie oparcie. Niespodziewanie dodatkowym jej atutem stało się uzyskanie certyfikatu Advanced ECDL¹¹ z edytora tekstów i arkusza kalkulacyjnego i tu również podkreślałam jej profesjonalizm. W drugim semestrze zaproponowałam jej uczestniczenie w zajęciach Matematyka dyskretna na WMil¹² UMK widząc w niej potencjał, który można zaangażować w tę tematykę. Jestem przekonana, że Sylwia będzie studiowała informatykę.

Uczeń konstruktor

Kornel był finalistą OIG. Obserwując go na lekcjach w pierwszej klasie liceum stwierdziłam, że mimo tego sukcesu nie ma gruntownej wiedzy algorytmicznej i nie radzi sobie z trudniejszymi problemami na lekcjach i seminarium. Przyczyną mógł być jego brak skupienia na konkretnym proble-

⁹ Strona Ogólnopolskiego Konkursu InterSieć jest dostępna pod adresem <http://oki.edu.pl/>.

¹⁰ Strona Olimpiady Informatycznej jest dostępna pod adresem <http://www.dialnetmasters.pl/>.

¹¹ Strona Europejskiego Certyfikatu Umiejętności Komputerowych jest dostępna pod adresem <http://www.ecdl.com.pl/>.

¹² Na mocy porozumienia między Uniwersytetem Mikołaja Kopernika i Gimnazjum i Liceum Akademickim, uczniowie po akceptacji Rady Pedagogicznej mogą uczestniczyć na pełnych prawach w wybranych zajęciach ze studentami każdego z kierunków uniwersytetu, m. in. w zajęciach Wydziału Matematyki i Informatyki UMK, <http://www.mat.uni.torun.pl>.

mie i ciągle dyskusje z kolegą na inne tematy. W pierwszej klasie przeszedł do drugiego etapu OI, ale zdobył niewiele punktów. Ponieważ Kornel od początku deklarował swoje zainteresowanie sieciami komputerowymi, skierowałam go na zajęcia z tego tematu na WMil UMK. Z czasem okazało się, że jego uczestnictwo w OI, to bardziej chęć zaspokojenia ambicji rodziców, związanych zawodowo z informatyką, a jego prawdziwe zainteresowania są związane z pracą nad sieciowymi aplikacjami. Zachęteni przeze mnie razem z kolegą przedstawili mi temat ich ciągłych dyskusji. Razem zaadaptowali jedną z gotowych aplikacji do obsługi radia przez Internet i sami dopisywali do niej nowe funkcjonalności, chcąc stworzyć radio uczniowskie. Aby umotywić go do dalszej pracy po porażce w OI postanowiłam wykorzystać temat radia i namówiłam obydwu uczniów do zespołowego przygotowania projektu na Olimpiadę Innowacyjności Technicznych¹³. Uczniowie ci pracują teraz nad połączeniem radia z telefonami komórkowymi, dzięki którym każdy uczeń na przerwie będzie mógł głosować na ulubioną piosenkę i w ten sposób mieć wpływ na repertuar muzyczny. Kornel zaczął jednocześnie regularnie pracować na lekcjach informatyki. Myślę, że w przyszłości ma duże szanse na sukces w OI.

Uczeń projektant stron WWW

Michał zawsze regularnie pracuje na lekcjach i seminariach informatycznych i nie ma większych kłopotów z podawaniem algorytmów rozwiązywania rozważanych podczas tych zajęć problemów. Cenne jest to, że lubi dyskutować i stara się przez to lepiej zrozumieć omawiane problemy. Niestety nigdy nie udało mu się odnieść sukcesu w żadnej z olimpiad, ani też zgłębić od strony programistycznej tematu sieci komputerowych. Sytuacja ta w drugiej klasie liceum spowodowała jego widoczne przygnębienie. Wśród kolegów, z których każdy znalazł swoje miejsce na sukcesy w informatyce, czuł się mało wartościowy i coraz częściej mówił o tym, że przestaje myśleć o wiązaniu swojej przyszłości z informatyką. Od początku zastanawiające było to, że prawie za każdym razem, gdy było to możliwe, rysował na tablicy wymyślone przez siebie grafiki. Okazało się, że ma duże zdolności plastyczne i wrażliwość na sztukę. Starałam się angażować go do projektowania elementów graficznych na komputerze i do tworzenia logo dla różnych przedsięwzięć w szkole. Jednak i w tym nie znalazł swojej pasji. Przy okazji lekcji formatowania dużego dokumentu zauważyłam, że w jego końcowym dokumencie jest zaskakująca harmonia w doborze czcionki, koloru, rozmieszczeniu tekstu i grafiki połączona z dobrym stylem tworzenia dokumentu od strony technicznej. Podobne odczucia towarzyszyły mi przy ocenianiu stworzonej przez niego strony WWW. Michał ma wyjątkowe wyczucie plastyczne, które sprawia, że tworzone przez niego dokumenty czyta się z przyjemnością, łatwo trafiają do użytkownika najważniejsze informacje ze strony internetowej. To cecha, którą ma rzadko który projektant stron WWW. Powiedziałam mu o tym oraz, że w przyszłości wróżę mu duże powodzenie w zakresie projektowania witryn Internetowych. Dziś jest już w klasie maturalnej przygotowuje się do matury z informatyki i ma zamiar studiować informatykę na politechnice. Twierdzi, że tamte moje słowa dodały mu skrzydeł informatycznych.

Podsumowanie

Skupianie się w klasach informatycznych jedynie na trenowaniu uczniów do olimpiady informatycznej może prowadzić do zgubienia wielu talentów i spowodować odwrócenie się ich od informatyki. Wielką sztuką dla nauczyciela jest nie odrzucanie uczniów, którzy wybierają drogę samodzielnego uczenia się nie uznając nauczyciela za mistrza. Często po uznaniu ich wyborów uczni-

¹³ Regulamin Olimpiady Innowacyjności Technicznych jest dostępny pod adresem http://www.lktir.com/mam-b02/index.php?option=com_content&task=view&id=29&Itemid=97.

wie ponownie do nas wracają. Zauważając zainteresowania uczniów zyskujemy ich zainteresowanie dla tematów realizowanych przez nas na lekcji.

Nawet w obrębie jednej profilowanej klasy można dostrzec uczniów lub całe ich grupy o różnych zainteresowaniach informatycznych. Każdy z nich potrzebuje akceptacji co do swych zainteresowań i wskazania przez nauczyciela możliwości ich rozwoju. Każdy z nich potrzebuje docenienia i pozytywnej motywacji. Często błędem jest uznawanie sukcesów uczniów tylko w Olimpiadzie Informatycznej, gdy tymczasem obszarów docenienia zainteresowań informatycznych uczniów jest znacznie więcej. Takie błędne podejście jest źródłem wielu rozczarowań uczniów, braku ich wiary we własne możliwości. Powoduje to utratę uczniów dla informatyki i rozpowszechnianie opinii, że jest bardzo trudnym przedmiotem.

Odpowiednie zdiagnozowanie wszystkich rodzajów zainteresowań i zaopiekowanie się tymi uczniami, którzy je mają, prowadzi do sukcesów uczniów w różnych konkursach i olimpiadach, rozwija ich i sprawia, że czują się zdolnymi do podjęcia coraz poważniejszych wyzwań, co nierzadko okryznaną drogą sprawia, że uczniowie zaczynają odnosić sukcesy w olimpiadach. Często sięgają potem po inne dziedziny i z powodzeniem startują w olimpiadach przedmiotowych z tych dziedzin. Muszą tylko znaleźć w nauczycielu mistrza, który pozwoli uwierzyć im w to, że są zdolni.

Literatura

- [1] Banachowski L., Diks K., Rytter W., *Algorytmy i struktury danych*, WNT, Warszawa 2003.
- [2] Broda P., Smolucha D., *Informatyka, Podręcznik cz. I., II, Liceum ogólnokształcące*, Operon, Gdynia 2005.
- [3] Cormen T. H., Leiserson C. E., Rivest R. L., *Wprowadzenie do algorytmów*, WNT, Warszawa 2000.
- [4] Gurbiel E., Hardt-Olejniczak G., Kołczyk E., Krupicka H., Sysło M. M., *Informatyka. Podręcznik Cz. 1. i 2. Kształcenie w zakresie rozszerzonym*, WSiP, Warszawa 2002.
- [5] Komitet Główny Olimpiady Informatycznej, *Olimpiada Informatyczna, niebieskie książeczki*, <http://www.oi.edu.pl>.
- [6] Lipski W., *Kombinatoryka dla programistów*, WNT, Warszawa 2004.
- [7] *Materiały elektroniczne do studiów Informatycznych na Uniwersytecie Warszawskim*, <http://wazniak.mimuw.edu.pl>.
- [8] Skiena S. S., Revilla M. A., *Wyzwania programistyczne*, WSiP, Warszawa 2004.
- [9] Stańczyk P., *Algorytmika praktyczna nie tylko dla mistrzów*, WN PWN, Warszawa 2009.
- [10] Sysło M. M., *Algorytmy*, WSiP, Warszawa 1997.
- [11] Sysło M. M., *Piramidy, szyszki i inne konstrukcje algorytmiczne*, WSiP, Warszawa 1998.



PRACA Z MŁODZIEŻĄ UZDOLNIONĄ INFORMATYCZNIE. DOŚWIADCZENIA RYSZARDA SZUBARTOWSKIEGO I STOWARZYSZENIA TALENT

Grzegorz Grochowski

Uniwersytet Kardynała Stefana Wyszyńskiego, Warszawa

grzegorz@grochowski.edu.pl

Innowacyjne formy i metody pracy z uzdolnioną młodzieżą (gimnazjum i liceum), realizowane przez Stowarzyszenie Talent, wyłoniły się jako efekt doświadczeń nauczyciela uczniów odnoszących sukcesy na olimpiadach i konkursach informatycznych w kraju i za granicą, mgr. inż. Ryszarda Szubartowskiego. Ryszard Szubartowski jest absolwentem Wydziału Cybernetyki Wojskowej Akademii Technicznej w Warszawie, posiada doświadczenie w pracy naukowo-dydaktycznej zdobyte w Akademii Marynarki Wojennej w Gdyni, a obecnie jest nauczycielem informatyki w szkołach: III Liceum Ogólnokształcące im. Marynarki Wojennej w Gdyni, Gimnazjum nr 24 w Gdyni, Gdańskie Liceum Autonomiczne, Gdańskie Autonomiczne Gimnazjum.

Bezprecedensowe sukcesy uczniów Ryszarda Szubartowskiego, odnoszone w międzynarodowych zawodach informatycznych, stanowią dla wielu młodych ludzi z całej Polski inspirację i wyzwanie dla poszukiwani dróg własnego rozwoju. W latach 1999-2008 Ryszard Szubartowski wprowadził łącznie 89 uczniów do etapu centralnego Ogólnopolskiej Olimpiady Informatycznej Licealistów, z czego 28 wróciło z tytułem laureata, zdobywając 7 złotych, 7 srebrnych i 14 brązowych medali. Ponadto, 17 uczniów Ryszarda Szubartowskiego reprezentowało Polskę w międzynarodowych olimpiadach informatycznych, zdobywając łącznie 34 medale, z czego aż 17 złotych, 8 srebrnych i 9 brązowych. Najbardziej utytułowanym jego uczniem jest Filip Wolski, obecnie student Uniwersytetu Warszawskiego, wielokrotny zwycięzca i medalista krajowych i międzynarodowych olimpiad informatycznych. Powszechnie znanym jego osiągnięciem jest zwycięstwo w Akademickich Mistrzostwach Świata w Programowaniu Zespołowym w Tokio (2007), a także na Międzynarodowej Olimpiadzie Informatycznej w Meksyku (2006). Niedawno Międzynarodowa Olimpiada Informatyczna (IOI) ogłosiła ranking 30 najlepszych uczestników tych prestiżowych zawodów i Filip Wolski uplasował się w nim <http://www.talent.edu.pl/images/stories/osiagi/4.jpg> na pierwszym miejscu. Uwzględniono przy tym, że każdy jego udział w Olimpiadzie zakończył się zdobyciem złotego medalu (2003, 2004, 2005 i 2006). Filip do tej pory jest najmłodszym w świecie złotym medalistą Międzynarodowej Olimpiady Informatycznej.¹

¹ Najważniejsze osiągnięcia **Filipa Wolskiego** (podawane w kolejności chronologicznej od końca) to: zwycięstwo w Akademickich Mistrzostwach Świata w Programowaniu Zespołowym w Tokio; jako pierwszy w historii Polak zwyciężył w XVIII Międzynarodowej Olimpiadzie Informatycznej w Meksyku; zwycięstwo w Drużynowych Mistrzostwach Środkowej Europy w Budapeszcie; zwycięstwo w XIII Olimpiadzie Informatycznej Krajów Środkowej Europy w Chorwacji; zwycięstwo w XI Bałtyckiej Olimpiadzie Informatycznej na Litwie; zwycięstwo w X Bałtyckiej Olimpiadzie Informatycznej na Łotwie; zwycięstwo w XIII Ogólnopolskiej Olimpiadzie Informatycznej; zwycięstwo w XII Ogólnopolskiej Olimpiadzie Informatycznej; najmłodszy na świecie złoty medalista w XV Międzynarodowej Olimpiadzie Informatycznej w USA, złoty medal na XVII Międzynarodowej Olimpiadzie Informatycznej w Polsce, złoty medal na XVI Międzynarodowej Olimpiadzie Informatycznej



Na uwagę zasługują również liczne osiągnięcia i sukcesy podopiecznego Ryszarda Szubartowskiego, ucznia III LO z Gdyni, Jakuba Pachockiego, który m.in. zdobył srebrny medal na XXI Międzynarodowej Olimpiadzie Informatycznej w 2009, złoty medal na XVI Olimpiadzie Informatycznej Krajów Europy Środkowej, zwyciężył podczas XV Bałtyckiej Olimpiady Informatycznej, XVI Olimpiady Informatycznej i XIV Bałtyckiej Olimpiady Informatycznej. W konkursie organizowanym przez Google Jakub zajął piąte miejsce na świecie!

Zaproponowane przez Ryszarda Szubartowskiego formy pracy z uczniami zdolnymi jest efektem zaangażowania i determinacji osoby, która przyciągnęła do współpracy innych. Dostrzegając unikatowe efekty wysiłków edukacyjnych², rodzice jego wychowanków i byli uczniowie założyli organizację pozarządową o nazwie Stowarzyszenie Talent, której zadaniem jest inicjowanie i realizowanie działań edukacyjnych i wychowawczych, rozwijających pasję informatyczną i matematyczną młodzieży gimnazjalnej i licealnej. Po powstaniu tego Stowarzyszenia, aktywność Ryszarda Szubartowskiego i jego współpracowników nabrała rozmachu. Obecnie w zajęciach organizowanych lub współorganizowanych przez Stowarzyszenie bierze udział rocznie około 300 zdolnych uczniów praktycznie z terenu całej Polski³. Ryszard Szubartowski podkreśla, że o skuteczności działań stanowią nie tyle zdolności pedagogiczne poszczególnych edukatorów, lecz przyjęte metody i realizowane procesy kształceniowe⁴, warto zatem przyrzeć się bliżej, na czym polega zaproponowana przez niego forma pracy z uczniami uzdolnionymi – poniżej poruszamy zagadnienia kluczowe dla jej zrozumienia.

1. Narodziny talentu

Jednym z założeń, tkwiących u podstaw metody realizowanej przez Stowarzyszenie Talent, jest przekonanie, że istotnym elementem strategii kształcenia dzieci utalentowanych – nie tylko w zakresie matematyki i informatyki – jest odkrywanie indywidualnych talentów i uświadamianie ich istnienia

w Grecji; złoty medal na XII Olimpiadzie Informatycznej Krajów Środkowej Europy na Węgrzech; złoty medal na XI Olimpiadzie Informatycznej Krajów Środkowej Europy w Polsce, złoty medal na X Olimpiadzie Informatycznej Krajów Środkowej Europy w Niemczech; złoty medal na XI Olimpiadzie Informatycznej, złoty medal na X Olimpiadzie Informatycznej; srebrny medal na IX Bałtyckiej Olimpiadzie Informatycznej w Estonii.

² Dla wykazania fenomenu skuteczności działań edukacyjnych można przytoczyć wyniki trzech kolejnych uczniów: **Marcin Michalski** – zwycięstwo w X Olimpiadzie Informatycznej; złoty medal na IX Bałtyckiej Olimpiadzie Informatycznej w Estonii; złoty medal na VIII Bałtyckiej Olimpiadzie Informatycznej na Litwie; srebrny medal na XV Międzynarodowej Olimpiadzie Informatycznej w USA; srebrny medal na X Olimpiadzie Informatycznej Krajów Środkowej Europy w Niemczech; srebrny medal na IX Olimpiadzie Informatycznej; brązowy medal na XIV Międzynarodowej Olimpiadzie Informatycznej w Korei Południowej; brązowy medal na XII Olimpiadzie Informatycznej Krajów Środkowej Europy na Słowacji; brązowy medal na VII Bałtyckiej Olimpiadzie Informatycznej w Polsce; brązowy medal na VIII Olimpiadzie Informatycznej. **Jakub Kallas** – złoty medal na XVIII Międzynarodowej Olimpiadzie Informatycznej w Meksyku; złoty medal na X Bałtyckiej Olimpiadzie Informatycznej na Łotwie; złoty medal na XIII Olimpiadzie Informatycznej; srebrny medal na XII Bałtyckiej Olimpiadzie Informatycznej w Finlandii; srebrny medal na XI Olimpiadzie Informatycznej. **Jakub Łacki** – złoty medal na XVII Międzynarodowej Olimpiadzie Informatycznej w Polsce; złoty medal na X Bałtyckiej Olimpiadzie Informatycznej na Łotwie; srebrny medal na XII Olimpiadzie Informatycznej; srebrny medal na XVI Międzynarodowej Olimpiadzie Informatycznej w Grecji; srebrny medal na XI Olimpiadzie Informatycznej Krajów Środkowej Europy w Polsce; srebrny medal na XI Olimpiadzie Informatycznej.

³ Według dokumentacji Stowarzyszenia, w roku szkolnym 2006/07 w obozach informatycznych uczestniczyli uczniowie z następujących miejscowości: Gdynia, Gdańsk, Sopot, Bydgoszcz, Bielsko-Biała, Katowice, Kraków, Warszawa, Wrocław, Gorzów Wlkp., Chorzów, Kielce, Swarzędz, Babice, Barwice, Białogard, Braniewo, Brzozowa, Bytów, Damnica, Elbląg, Gorlice, Gostyń, Jasło, Jastków, Jaworzno, Kamienna Góra, Kartuzy, Kolbudy, Konin, Kosakowo, Kostrzyn, Kramarzyny, Kwidzyn, Lębork, Luzino, Michałowice, Mrągowo, Nowe Ostrowy, Nowy Dwór, Olsztyn, Olsztynek, Ornetka, Pleszew, Puławy, Reda, Ruda Śl., Rumia, Sadlinki, Siechnice, Siemianice, Skarżysko Kam., Straszyn, Sułkowice, Szczecin, Tczew, Tomaszów Maz., Wejherowo, Wieluń, Zapałów, Zielona Góra, Zwolen, Żukowo.

⁴ Notatki z wywiadu z Ryszardem Szubartowskim



przede wszystkim rodzinie, następnie pedagogom i psychologom, a także szkole. Wydaje się oczywiste, że każda polska szkoła powinna mieć wypracowane reguły postępowania z uczniem zdolnym. Kształcenie uczniów utalentowanych wymaga bowiem użycia innowacyjnych metod nauczania, pozwalających w pełni rozbudzić i rozwinąć drzemiący w nich potencjał. Z doświadczeń Ryszarda Szubartowskiego i jego współpracowników wynika, że dzieci zdolne powinny być kształcone w sposób twórczy, który koncentruje się na już posiadanych możliwościach poznawczych i stymuluje do dalszej percepcji i weryfikacji rzeczywistości.

Według Karoliny Pałeczki, psychologa pracującego w Stowarzyszeniu Talent, każdy człowiek jest z natury twórczy, jednakże w toku rozwoju na swojej drodze napotyka wiele trudności i praktyk, które skutecznie hamują jego wrodzony potencjał i kreatywność. Tradycyjny system kształcenia (często wciąż dominujący w polskim szkolnictwie), bazujący na wiedzy encyklopedycznej, jest przykładem na to, że zła edukacja może skutecznie i bezpowrotnie zabić potencjał, tkwiący w młodym człowieku. W wyniku nawału informacji, jakie musi przetworzyć mózg ucznia, nie starcza miejsca na eksperymentowanie i spontaniczne tworzenie. Jednostki kształcone w ten sposób stają się typowymi odtwórcami. W podejściu realizowanym w ramach stowarzyszenia Talent dąży się natomiast do tego, aby uczniowie byli przeświadczeni, iż każdy może wymyślić coś nowego i wartościowego, że posiada wystarczający potencjał do tworzenia nowych koncepcji i rozwiązań. Ważne, aby jednostka posiadała oprócz wiedzy, umiejętności i narzędzia rozwiązywania problemów a także zaufanie i akceptację samego siebie.

2. Szlifowanie diamentów – formy pracy z uczniem zdolnym

Formy pracy, z których Stowarzyszenie Talent korzysta w ramach działań edukacyjnych, koncentrują się głównie na stymulacji takich cech osobowości, które sprzyjają twórcemu, samodzielnemu myśleniu i działaniu. Rozwija się więc **myślenie dywergencyjne**, podkreślające wielość poprawnych rozwiązań oraz dostrzegania odległych związków między faktami dzięki użyciu analogii. Równie przydatne jest **myślenie konwergencyjne**, które pozwala rozwijać dokładność, systematyczność i precyzję – cechy niezbędne dla przyszłych programistów. Poniżej przedstawiamy najważniejsze formy organizacyjne pracy z uczniem zdolnym, wypracowane przez lata doświadczeń Ryszarda Szubartowskiego i kierowanego przezeń Stowarzyszenia.

Grupy warsztatowe

Są to niejako podstawowe „komórki edukacyjne”, w których dokonuje się aktywne kształcenie młodzieży zdolnej. Zasadniczym celem grup warsztatowych jest stworzenie uczniom pola do bezpiecznego i swobodnego eksperymentowania. Chodzi o to, by sprowokować gimnazjalistów i licealistów do niezależności w działaniu, równocześnie oferując im pomoc i niezbędne wiadomości. Zajęcia są prowadzone przez nauczyciela i jego instruktorów (od dwóch do trzech, w zależności od wielkości grupy). Najbardziej korzystną sytuacją jest, gdy jeden instruktor zajmuje się małą grupą uczniów (od 4 do 6 osób). Zajęcia przebiegają zgodnie ze sprawdzonym postępowaniem. Najpierw nauczyciel wprowadza poszczególne zadania informatyczne, tłumaczy i wyjaśnia na czym polegają. Następnie część z nich przekazuje do samodzielnego rozwiązania przez uczniów. Gdy w toku rozwiązywania pojawiają się trudności lub wątpliwości, nauczyciel oraz instruktorzy w każdej chwili służą pomocą lub wskazówką. Tempo objaśniania jest ściśle dostosowane do możliwości ucznia. Bardzo ważną rzeczą jest stopniowanie trudności. Chodzi o zachowanie swego rodzaju ciągłości, dzięki której uczeń będzie czuł się coraz pewniej i bezpieczniej. Każde kolejne zajęcia będą go utwierdzały w przekonaniu, że umie i rozumie coraz więcej a jego samopoczucie i samoocena będzie wzrastać, co wydaje się niezbędne dla utrzymania silnej motywacji i podtrzymywania ciekawości poznawczej.



Warsztaty psychologiczne z twórczości i „gimnastyki mózgu” dla gimnazjalistów

W większości sytuacji uczeń zdolny samodzielnie zdobywa, a następnie przetwarza informacje. W związku z tym jest wskazane zapewnienie mu narzędzi usprawniających i optymalizujących myślenie. W tym celu są przeprowadzane spotkania w formie warsztatu (treningu), służące usprawnianiu myślenia. Uczeń zapoznaje się z różnymi strategiami, które pomagają lepiej zapamiętywać obszerny materiał, oraz z technikami notowania (np. tworzenie mapy myśli). Wykonuje ćwiczenia wpływające korzystnie na funkcjonowanie pamięci i koncentracji. Uczestnicy dowiadują się również, co robić, żeby czytać szybciej przy jednoczesnym rozumieniu tekstu. Drugi blok tematyczny koncentruje się na wzmacnianiu twórczości i kreatywności. Ćwiczenia mają na celu przede wszystkim pobudzić młodzież do częstszego korzystania z rozwiązań nieszablonowych i nowatorskich, wzmocnić motywację twórczą oraz koncentrację, uczyć rozpoznawania barier w twórczym myśleniu, wspierać niezależność i otwartość, wspomagać umiejętności rozwiązywania konfliktów i problemów z wykorzystaniem niestandardowych sposobów postępowania.

Gry matematyczno-logiczne

Jest to atrakcyjna, intelektualna rozrywka, w szczególności dla grupy wiekowej gimnazjalistów. Uczeń uczy się w formie zabawy korzystania z różnych technik kombinatorycznych. Ponadto uczestnicy zajęć muszą wykazać się takimi umiejętnościami, jak: strategiczne planowanie, taktyczne myślenie a także wyobraźnię przestrzenną, co korzystnie wpływa na ich rozwój zarówno intelektualny jak i społeczny.

Ligi zadaniowe

Ta forma nauczania, przeznaczona dla gimnazjalistów, opiera się głównie na tak zwanej regule powszechności. Uczniowie tworzą czteroosobowe drużyny. Każda drużyna ma do swojej dyspozycji dwa komputery. Najbardziej pożądanym składem takiego zespołu powinien wyglądać następująco: jeden uczeń z klasy trzeciej, który pełni rolę naturalnego lidera; dwóch uczniów z klasy drugiej na przeciętnym poziomie wiedzy z określonej dziedziny, którzy pełnią funkcję pomagających, oraz uczennica bądź uczeń z klasy pierwszej – jako reprezentant o najmniejszym doświadczeniu, będący w roli obserwatora. Taka konfiguracja grupy spełnia wiele ukrytych funkcji, które są kluczem do przyszłego sukcesu każdego z członków drużyny. Dzięki niej zawodnicy czerpią od siebie nawzajem korzyści zarówno praktyczne jak i społeczne i psychologiczne. Lider, posiadający największą wiedzę stanowi wzór i źródło wiedzy dla swoich współtowarzyszy, co niewątpliwie wzmacnia jego samoocenę i ustawia go w pozycji „eksperta”. Dzięki niemu zarówno pomagający jak i obserwator zdobywają wiedzę i cenne umiejętności, które w sposób świadomy i podświadomy przyswajają do swojego systemu poznawczego. Ważne i cenne z punktu widzenia mobilizacji jest to, że wszyscy uczestnicy otrzymują taką samą ilość punktów niezależnie od włożonego wkładu. Jest to czynnik, który działa na zawodników motywująco, zwłaszcza jeśli chodzi o obserwatora, którego wkład jest najmniejszy. Czuje się on zobowiązany wobec drużyny, dlatego w sposób naturalny zaczyna interesować się i zgłębiać daną dziedzinę. Dodatkową korzyścią dla wszystkich uczestników jest nabywanie umiejętności współpracy i komunikacji w grupie, rozdzielanie i wypełnianie poszczególnych poleceń, rozwiązywanie konfliktów i trudności.

Konkursy indywidualne dla gimnazjalistów

Głównym celem konkursów jest wypracowanie w uczniach takich cech, jak systematyczność i samodzielność myślenia. Uczeń musi sam na sam zmierzyć się z wyzwaniem w postaci zadań. Po fazie rozwiązywania następuje kolejny etap, to jest omówienie, podczas którego dochodzi do wymiany



myśli i burzy mózgów. Biorą w niej udział autorzy zadań i rozwiązujący je uczestnicy. Trzeba zaznaczyć, że na tym etapie uczniowie nie otrzymują gotowego, wzorcowego rozwiązania a jedynie wskazówki, które mogą ich skierować na właściwy tor myślenia. Jest to skuteczne podejście, gdyż skłania do drążenia problemu i dalszych poszukiwań. Prowokuje do stawiania własnych, spontanicznych hipotez i interpretacji. Ten rodzaj oddziaływania uczy sztuki argumentacji i pomaga przełamywać nieśmiałość. Ostatnim elementem konkursu indywidualnego jest swobodne eksperymentowanie uczestników. Zadania są umieszczane na otwartym portalu, gdzie można dalej pracować nad rozwiązaniem. Tutaj uczniowie mogą korzystać z wszelkich pomocy zarówno w Internecie jak i specjalistycznej literaturze. Szukanie prawidłowego rozwiązania może odbywać się w szkole, grupowo wraz z kolegami albo w domu.

Konkursy indywidualne dla licealistów

Konkursy przebiegają z podziałem na dwie grupy wiekowe. Pierwsza składa się z uczniów najstarszych, którzy pracują nad rozwiązaniami zadań samodzielnie przez okres od czterech do pięciu godzin. Działalność uczniów z grupy drugiej (młodszej) ogranicza się do identycznej aktywności (znalezienia prawidłowego rozwiązania otrzymanych zadań) z tą tylko różnicą, iż pracują o godzinę krócej. Następnie dochodzi do omówienia zadań na takich samych zasadach jakie obowiązują w grupie gimnazjalnej. Po zajęciach zadania są umieszczane na serwerze, gdzie można pracować, eksperymentować i konsultować się z kolegami.

Konkursy drużynowe dla licealistów

Drużyna składa się z trzech osób. Każdy zespół ma do dyspozycji jeden komputer. Konkurs opiera się na formule zawodów w programowaniu zespołowym dla studentów (ACM). Zasadniczym celem tego typu działań jest pogłębienie i usprawnienie nabytej w poprzednich latach wiedzy i umiejętności pracy w zespole, umiejętności rozwiązywania potencjalnych konfliktów a także zapobiegania im. Kolejnym pozytywnym efektem tej formy pracy jest możliwość zobrazowania uczniom zdolnym, jak wyglądają zawody dla studentów. Dzięki temu można mieć nadzieję, że w przyszłości łatwiej im będzie odnaleźć się w uczelnianym środowisku. Będzie to dla nich sytuacja znana i dzięki temu mniej stresująca czy pochłaniająca zasoby poznawcze, co w konsekwencji może mieć korzystny wpływ na przyszłe osiągnięcia.

Warsztaty psychologiczne z komunikacji i treningi relaksacyjne dla licealistów

Podjęmowane w ramach tych zajęć aktywności promują pracę w grupie, podkreślają zalety współpracy a także rozwijają umiejętność nawiązywania kontaktu, rozwiązywania konfliktów, wygłaszania przemówień, kształtują sprawności społeczne i komunikacyjne. Bardzo ważne jest tu uczestnictwo w dyskusjach, które uczą doboru trafnych argumentów oraz szacunku dla innych osób. Przeprowadzane są również treningi relaksacyjne (indywidualnie i grupowo), kształtujące umiejętności związane z redukowaniem negatywnych skutków stresu. W ramach tych zajęć realizowany jest także trening przy użyciu narzędzia jakim jest *biofeedback*, wzmacniający odporność na stres oraz usprawniający koncentrację. Jest to metoda poprawiająca funkcjonowanie mózgu.

Praca asystentów

Asystentami mogą zostać studenci oraz bardziej zaawansowani uczniowie. Do ich obowiązków należy głównie tworzenie zadań dla młodszych kolegów, co samo w sobie jest działalnością bardzo zaawansowaną, wymagającą myślenia abstrakcyjnego, uruchomienia wyobraźni, a także posiadania dużej wiedzy, którą można się swobodnie posługiwać. Kolejnym zadaniem asystentów jest prowa-



dzenie wykładów oraz warsztatów dla młodszych uczniów. W ten sposób asystenci uczą się jak skutecznie przekazywać wiadomości innym, szlifują własne umiejętności autoprezentacji, komunikacji oraz nieustannie doskonalą i pielęgnują własną wiedzę.

Obozy naukowe

Według Ryszarda Szubartowskiego praca z uczniem zdolnym wymaga **zmiany środowiska**. Zmiana ta powinna odbywać się co najmniej dwa razy w semestrze i mieć formę obozów naukowych, podczas których panują odmienne warunki niż w szkole. Uczeń zdolny potrzebuje różnorodności i stymulacji, a zmiana otoczenia przyspiesza jego fazy uczenia się. Dodatkowo obóz naukowy pełni bezcenną rolę integracyjną (uczeń poznaje osoby podobne do siebie o podobnych zainteresowaniach i możliwościach, również związane ze środowiskami akademickimi). W trakcie obozów ma miejsce wielowymiarowy rozwój ucznia zdolnego: naukowy, społeczny, emocjonalny, fizyczny, wzmacniane są również postawy prozdrowotne. Stowarzyszenie Talent organizuje dwa rodzaje obozów: sześciodniowe odbywające się w czasie roku szkolnego oraz dwunastodniowe organizowane w trakcie wakacji letnich. Podczas obozów są wykorzystywane następujące metody oraz formy pracy:

Formy pracy na obozie 6-dniowym gimnazjalistów: Odbywają się dwugodzinne zajęcia z informatyki. Podczas tych zajęć ma miejsce równoległe konkurs indywidualny i warsztatowy. Dzieci w zależności od stopnia zaawansowania zapisują się na jeden z nich. Inne formy aktywności to konkursy drużynowe, omawianie zadań, przeprowadzanie wykładów. Organizowane są również warsztaty psychologiczne, aktywność fizyczna lub spacerowanie po okolicy, gry matematyczno-logiczne, treningi relaksacyjne.

Formy pracy na obozie 12-dniowym gimnazjalistów: Zajęcia mają taką samą strukturę jak na obozie 6-dniowym z tą różnicą, że co dwa dni jeden dzień jest poświęcony rekreacji. Organizowane są wówczas wycieczki w góry, ogniska, wyjścia na basen lub do kina.

Formy pracy na obozie 6-dniowym licealistów: Zajęcia mają formułę konkursów indywidualnych, konkursu drużynowego, omawiania zadań, wykładów. Realizowane są również warsztaty psychologiczne z komunikacji i autoprezentacji, aktywność fizyczna, praca własna, konsultacje lub dodatkowe wykłady. Zasadą jest, że uczeń sam wybiera aktywność, która najbardziej odpowiada jego potrzebom.

Formy pracy na obozie 12-dniowym dla licealistów: Program zajęciowy jest jak opisano wyżej. Co trzeci dzień ma miejsce rekreacja i turystyka. Organizowane są również spotkania z interesującymi osobami oraz ogniska.

Powyższe formy organizacyjne i metody pracy stanowią wynik wieloletnich poszukiwań, niejednokrotnie podejmowania inicjatywy o charakterze eksperymentalnym i pionierskim. Nie są to jednak działania realizowane wyłącznie jako wynik intuicji pedagogicznej. Stanowią pozytywny przykład dialogu teorii i praktyki dydaktycznej oraz twórczej implementacji dobrych praktyk edukacyjnych.

3. Dialog teorii i praktyki w kształceniu uczniów zdolnych

Dla realizacji celów kształcenia informatycznego Stowarzyszenie Talent wykorzystuje zróżnicowane typy nauczania. Znajdziemy tu więc **nauczanie podające**, realizowane poprzez metody asymilacji wiedzy, takie jak: wykłady, objaśnianie reguł, algorytmów czy wzorów matematycznych. Typowe problemy z koncentracją czy dyscypliną, towarzyszące niejednokrotnie tego typu nauczaniu



[5, str. 210-213], są niwelowane poprzez fakt, że uczestnicy biorą udział w zajęciach dobrowolnie, wyposażeni w silną motywację opartą na rozbudzonej pasji poznawczej. Elementem wzmacniającym atrakcyjność nauczania podającego jest fakt, że wykłady czy objaśnienia są realizowane przez specjalistów, mających na swoim koncie kształcenie mistrzów i laureatów międzynarodowych olimpiad informatycznych.

Skutecznemu „szlifowaniu diamentów” służy także **podejście problemowe** [3, str. 109-116]. Towarzyszy ono rozwiązywaniu zadań informatycznych a następnie wspólnemu omawianiu ich wyników. W trakcie rozwiązywania zadań uczeń stykając się z problemami w nich zawartymi doświadcza ograniczenia i niewystarczalności własnej wiedzy. Powstałe odczucie braku skłania go do zadawania pytań nauczycielowi a także samodzielnych poszukiwań narzędzi teoretycznych umożliwiających rozwikłanie zagadnienia. Wyzwanie, jakim jest problem zawarty w zadaniu, mobilizuje uczestnika zajęć do przeglądu zdobytej uprzednio wiedzy teoretycznej jak również rozbudza myślenie twórcze. Skłania do stawiania autorskich hipotez oraz uaktywnia procedury ich weryfikacji. Jak już wcześniej stwierdzono, w celu uzyskania pożądanego efektu aktywizacyjnego często planuje się czynności dydaktyczne w ten sposób, aby uczniowie przystępowali do zadań zanim posiadą wiedzę teoretyczną umożliwiającą ich pełne rozwiązanie. Takie podejście powoduje, że uczestnicy dochodzą nieraz do rozumienia złożonych procesów niejako mimochodem, intuicyjnie, niejednokrotnie nie zdając sobie z tego sprawy. Jednakowoż warto powtórzyć, że ważne jest, by poziom trudności ćwiczeń skorelowany był z poziomem kompetencji intelektualnych uczniów, by ich zbyt nie przerastał. Aby tak się działo, uczestnicy obozów i zajęć pozalekcyjnych dzieleni są na grupy zarówno pod względem wiekowym jak i podług posiadanych zdolności. Nauczanie problemowe uwidacznia się również na etapie omawiania rozwiązań poszczególnych zadań, kiedy obok przedstawienia rozwiązania, mogącego stanowić wzorzec lub inspirację, zachęca się uczniów do zgłaszania własnych pomysłów i analizuje wspólnie możliwe warianty w podejściu do omawianych kwestii. Wtedy pojawiają się charakterystyczne dla nauczania problemowego metody pracy, jakimi są burza mózgów i dyskusja [5, str. 261-265].

Szczególnie ciekawą i skuteczną dydaktycznie formę stymulacji myślenia produktywnego, stanowi opracowywanie zadań. Jest to forma pracy samokształceniowej (walor edukacyjny) a zarazem działanie na rzecz dobra wspólnoty kształcenia (walor wychowawczy). Układanie zadań jest czynnością o wiele bardziej złożoną intelektualnie niż ich rozwiązywanie. Obok wiedzy merytorycznej wymaga ono w wysokim stopniu myślenia twórczego („odwracanie” problemu) a także wyobrażenia o możliwościach intelektualnych osób, które dane zadanie będą rozwiązywać. Z tych względów, tą formą pracy posługują się uczestnicy bardziej zaawansowani merytorycznie i formalnie. Układają oni zadania dla młodszych kolegów znajdujących się na wcześniejszych etapach „wtajemniczenia”. Tworzy się w ten sposób swoista „sztafeta transferu wiedzy”.

Pośród typów nauczania realizowanych w ramach Stowarzyszenia Talent jest także **nauczanie programowane** [4, str. 160-186]. Istotnym elementem procesu kształcenia, charakterystycznym dla nauczania programowanego, jest natychmiastowa i dająca się obiektywnie zmierzyć weryfikacja jego rezultatów. Dzięki odpowiednim programom sprawdzającym, uczestnicy zajęć mają możliwość oceny własnych wytworów zgodnie z przyjętymi kryteriami. Ocena ta jest tak dalece precyzyjna, że „cenzurki” wystawione przez komputer opisują efekt pracy w procentach. Tego typu podejście nakierowuje uwagę ucznia na obszary, w których jego działanie domaga się weryfikacji lub udoskonalenia. Ważną rolę odgrywa stosowanie jednakowych kryteriów oceny dla wszystkich uczestników działań. Umożliwia ono porównanie własnych umiejętności a także wzajemną inspirację osiągnięciami. Odpowiednie „rankingi” wyników są dostępne natychmiast w postaci elektronicznej lub drukowanej i w razie potrzeby przekazywane uczniom. Innym ważnym elementem nauczania programowanego jest możliwość indywidualizacji kształcenia poprzez dostosowywanie zadań do



poziomu intelektualnego uczniów a także możliwość wielokrotnego powracania do zagadnień, których uczestnicy zajęć nie byli w stanie wcześniej opanować.

Istotną cechą nauczania realizowanego w ramach omawianego systemu jest szacunek dla **dydaktycznej zasady związku teorii z praktyką** [1, str. 148-150]. Przyjęte jest tu jako „dogmat” założenie, że proces edukacyjny kończy się dopiero w momencie, gdy uczniowie są w stanie samodzielnie zaprezentować zdobytą wiedzę. Czynią to biorąc udział w olimpiadach i zawodach informatycznych, podczas których weryfikują swoje umiejętności, stosując poznane procedury w nowych kontekstach ćwiczebnych, w atmosferze autentycznego współzawodnictwa, przeżywając stres i emocje. Rozwiązywanie zadań staje się dla nich poważnym wyzwaniem o charakterze wytwórczym. Nie jest to „pisanie do szuflady”, lecz wytwarzanie konkretnego dzieła, które jest oceniane przez fachowców, często za pośrednictwem odpowiednich programów.

W działaniach Stowarzyszenia Talent dotyczących kształcenia istotną rolę odgrywają elementy **nauczania wielostronnego**. Jedną z podstaw sukcesu stanowi oddziaływanie na wyobraźnię oraz zakorzenienie w doświadczeniu. Poznawanie zarówno prostych jak i złożonych zagadnień informatycznych dokonuje się poprzez operacje na obiektach bliskich codziennemu życiu uczniów lub też znajdujących się w obrębie ich preferencji kulturowych (przedmioty codziennego użytku, relacje międzyludzkie, świat fantasy, gry liczbowe, polityka, komunikacja itp.). Przykładowo poszukując rozwiązań, służących automatyzacji czynności i procesów, uczniowie „usprawniają działanie parlamentu” (jaka koalicja ma szansę na przetrwanie, gdy liczba partii wynosi np. milion), „rozwiązują problemy zakochanych” (możliwość kombinacji „randek w ciemno”, dobieranych przez komputer), pomagają zwyciężać uczestnikom teleturniejów (którą „bramkę” z ukrytym prezentem bardziej opłaca się wybrać „A” czy „B”?) usprawniają działania policji (jak najtaniej i najbardziej optymalnie zablokować drogę groźnemu przestępcy?) itp. Autorom zadań przyświeca cel nie tylko urozmaicenia ich treści, ale również wskazania na związek algorytmiki z codziennym życiem, które składa się w sporej mierze z czynności, podlegających optymalizacji i automatyzacji.

4. Środowisko wspierające kształcenie zdolnych

Zarówno teoria jak i praktyka kształcenia uczniów zdolnych wskazuje na niedającą się przecenić rolę środowiska, które może zarówno hamować jak wspomagać edukację. Ryszard Szubartowski przyjmuje założenie, że dzieci utalentowane stanowią kapitał intelektualny Polski, w związku z tym należy uznać, iż ich obecność stanowi szansę dla społeczeństwa. Integracja otoczenia z uczniem zdolnym jest możliwa na zasadzie wzajemnej inspiracji i uczenia się funkcjonowania w jednej rzeczywistości. Należy przy tym pamiętać, że droga rozwoju ucznia zdolnego jest bardzo trudna. Uczeń utalentowany, ze względu na wysoki poziom kreatywności, tworzy własne pojęcia, operuje specyficznym językiem, co często spotyka się z niezrozumieniem ze strony otoczenia (odbierane jest jako fantasta, kłamca, czy wręcz zaburzony psychicznie). Prowadzi to nierzadko do izolacji. Należy zatem przyzwyczaić społeczeństwo do specyficznego funkcjonowania, odczuwania i doświadczania rzeczywistości przez ucznia zdolnego. Jeśli zaś idzie o samego ucznia, to w procesie jego integracji ze społecznością istotną rolę odgrywa psycholog, który powinien uczyć go takich umiejętności, jak kontrola stresu, umiejętność komunikacji i pracy w grupie.

Wśród instytucji wspierających rozwój dzieci zdolnych niewątpliwie centralną rolę odgrywa szkoła. Czy współczesna polska oświata dysponuje narzędziami prawnymi-organizacyjnymi aby diagnozować uczniowskie uzdolnienia i indywidualizować swoje podejście do uczniów zdolnych? Wydaje się, że najbardziej dostępną drogą rozwoju ucznia zdolnego w Polsce jest realizacja indywidualnego toku nauczania. Może ona odbywać się dwoma ścieżkami. Pierwsza z nich polega na tym, że uczeń realizuje indywidualny tok nauczania w swoim środowisku klasowym czy też w klasie integracyjnej.

Podejście to umożliwia pozostanie uczniowi w otoczeniu, w którym dobrze się czuje, a szkoła i młodzież uczy się rozumienia jego sposób myślenia i działania. Dla uczniów szczególnie uzdolnionych w określonym kierunku (np. informatycznym), pozostających w swoich klasach, powinien zostać przygotowany program dostosowujący do samodzielnej pracy z pozostałymi przedmiotami.

Drugą możliwością jest tworzenie na poziomie powiatów klas składających się z uczniów zdolnych, zaś na poziomie województwa – szkół. Każda uczelnia powinna umieć skorzystać z możliwości opieki nad uczniami zdolnymi, do czego może w znacznej mierze przyczynić się kontynuacja indywidualnego toku nauczania w całym procesie kształcenia – od szkoły podstawowej po naukę w uczelni wyższej. Dobrymi praktykami pod tym względem mogą poszczycić się już niektóre uczelnie, np. Uniwersytet Warszawski, Uniwersytet Wrocławski czy Uniwersytet Jagielloński. Wartościowym przykładem tworzenia „punktów centralnych”, w których rozwijają się talenty jest Gimnazjum i Liceum Akademickie w Toruniu. Ważnym elementem w procesie odkrywania i rozwoju zdolnej młodzieży jest właściwe rozpowszechnianie informacji o miejscach i placówkach rozwoju indywidualnych uzdolnień, umożliwiających uczniom i ich rodzicom podjęcie właściwych decyzji. Na etapie diagnozy talentów istotną rolę kontrolującą mogą odgrywać poradnie pedagogiczno-psychologiczne.

Aby system szkolnictwa skutecznie wspierał rozwój zdolnej młodzieży, potrzebna jest wykwalifikowana kadra pedagogiczna. Nauczyciele poświęcający czas na pracę z uczniami uzdolnionymi powinni wykazywać się wiedzą merytoryczną, dydaktyczną i umiejętnościami wychowawczymi służącymi kształtowaniu postaw etycznych. Według Ryszarda Szubartowskiego ich rola polega na „prowadzeniu ucznia mądrą drogą”. W celu dostarczenia takowej wiedzy i umiejętności, Stowarzyszenie Talent organizuje w trakcie wakacji obozy dla nauczycieli. Istotną rolę odgrywa również kształcenie studentów, przyszłych nauczycieli. Ci ostatni uczą się pracy z młodzieżą szkolną, pełniąc rolę asystentów. Pracę asystentów odbywają pod nadzorem i kontrolą nauczycieli. Asystenci przechodzą szkolenia w zakresie metod pracy z uczniem zdolnym, umiejętności komunikowania i referowania, redagowania tekstów, skutecznej prezentacji, planowania własnej przyszłości i ścieżki rozwoju. Kształcą się jednocześnie w kierunku pedagogicznym w celu uzyskania uprawnień opieki nad dziećmi i młodzieżą, np. jako wychowawcy kolonijni.

Mówiąc o znaczeniu środowiska w kształceniu uczniów zdolnych nie sposób pominąć faktu, iż spora liczba uzdolnionych uczniów pochodzi z małych i średnich miejscowości. Z doświadczenia i statystyk wynika, że uczniowie, którzy uzyskują duże sukcesy w Olimpiadzie Matematycznej i Informatycznej pochodzą właśnie z małych miasteczek czy wsi, natomiast uczą się w szkołach w dużych aglomeracjach, w konsekwencji ewidencjonowani są jako ich mieszkańcy. Uczniowie z małych miejscowości rokują szczególnie dużą nadzieję na dynamiczny rozwój talentu, gdyż charakteryzują się wysokim poziomem motywacji, determinacji i samodzielności w działaniu. Do tej pory jedyną metodą rozwijania ich potencjału było przesuwanie do wyższych klas, co często skutkuje zbyt wczesnym (i niejednokrotnie bolesnym) wkroczeniem w niezrozumiały emocjonalnie i społecznie świat starszych uczniów. W stworzeniu przyjaznego i skutecznego systemu wsparcia uczniów zdolnych z małych miejscowości kluczową rolę powinny odgrywać samorządy i administracja lokalna. Dlatego ważne jest, by na poziomie powiatu znajdowała się co najmniej jedna poradnia pedagogiczno-psychologiczna, specjalizująca się w diagnozowaniu zdolnych uczniów, stosująca przyjęte i wystandaryzowane narzędzia pomiaru. Jeśli jednak działania poradni mają charakter lokalny (tzn. są finansowane z budżetu miasta/gminy), dopuszcza się dowolność stosowanych metod zakładając, iż badania wykonują osoby posiadające uprawnienia i kompetencje. Uświadomienie rodzicom specyfiki potencjału (wyróżniających się predyspozycji) dziecka, odgrywa istotną rolę w procesie odkrywania talentu i stanowi klucz do sukcesów w całym procesie rozwoju.

Powyżej zostały przedstawione kluczowe zagadnienia umożliwiające poznanie i zrozumienie swoistego fenomenu, jakim jest działalność Ryszarda Szubartowskiego i Stowarzyszenia Talent. Jest



to wyrazista i wykazująca wyjątkową skuteczność forma aktywności pedagogicznej, która stawia pewnego rodzaju „wyzwanie” systemowi edukacji szkolnej w zakresie pracy z uczniami zdolnymi [2, str. 95-113]. W sposób istotny rozwija pasje poznawcze, podnosi wyniki nauczania a także ma ambicje kreowania typu osobowości odpowiedzialnej społecznie i działającej na rzecz dobra wspólnego. Być może część z doświadczeń Stowarzyszenia Talent można przeszczepić na grunt szkolnego systemu kształcenia z pożytkiem dla jego skuteczności.

Osoby, które korzystają z działań kształceniowych Stowarzyszenia Talent to przyszłe elity społeczeństwa i gospodarki opartej na wiedzy. O laureatów olimpiad informatycznych walczą wszak potężne i renomowane firmy⁵. Należy mieć nadzieję, że upowszechnienie doświadczeń nauczycieli, takich jak Ryszard Szubartowski, przyczyni się do rozwoju kapitału intelektualnego polskiego społeczeństwa.

Literatura

1. Bereźnicki F., *Dydaktyka ogólna w zarysie*, Miscellanea, Koszalin 1994.
2. Eby J. W. J., Smutny F., *Jak kształcić uzdolnienia dzieci i młodzieży*, WSiP, Warszawa 1998.
3. Kruszewski K. (red.), *Sztuka nauczania. Czynności nauczyciela*, WN PWN, Warszawa 1994.
4. Kupisiewicz Cz., *Podstawy dydaktyki ogólnej*, Polska Oficyna Wydawnicza BGW, Warszawa 1996.
5. Okoń W., *Wprowadzenie do dydaktyki ogólnej*, Wydawnictwo Akademickie Żak, Warszawa 1996.

⁵ Na przykład Filip Wolski jako jedyny w Europie student uzyskał stypendium firmy Rolex i odbywał praktyki w renomowanych firmach, takich jak Google.

CZEŚĆ III

PRZYKŁADOWE SCENARIUSZE ZAJĘĆ INFORMATYCZNYCH

Moduł: Algorytmika i programowanie

Moduł: Bazy danych

Moduł: Sieci komputerowe

Moduł: Multimedia



REKURENCJA – JAKIE TO PROSTE!

Maciej M. Sysło

Uniwersytet Wrocławski
Uniwersytet Mikołaja Kopernika w Toruniu
syslo@ii.uni.wroc.pl

1. Zagadnienia programowe

Rekurencja pojawia się w szkole na lekcjach matematyki i informatyki.

Matematyka, IV etap edukacyjny, zakres rozszerzony

W piątym dziale Ciągi, od ucznia oczekuje się, że: 1) wyznacza wyrazy ciągu [arytmetycznego lub geometrycznego – dopisek M. M. Sysło] określonego wzorem rekurencyjnym.

Komentarz. Powyższa umiejętność wymaga, by uczeń wiedział, co to jest wzór rekurencyjny – wyraz ciągu jest określony przez odwołanie do innych wyrazów tego ciągu; na ogół są to odwołania do wcześniejszych wyrazów, czasem tylko do jednego.

Informatyka, IV etap edukacyjny, zakres rozszerzony

Rekurencja pojawia się w trzech miejscach jako: metoda rozwiązywania problemów, realizacja algorytmów oraz ich implementacja w języku programowania.

5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń:
 - 9) stosuje rekurencję w prostych sytuacjach problemowych;
 - 11) opisuje podstawowe algorytmy i stosuje:
 - a. algorytmy na liczbach całkowitych, np.:
 - iteracyjna i rekurencyjna realizacja algorytmu Euklidesa,
 - iteracyjne i rekurencyjne obliczanie wartości liczb Fibonacciego,
 - f. algorytmy badające własności geometryczne, np.:
 - konstrukcje rekurencyjne: drzewo binarne, dywan Sierpińskiego, płatek Kocha;
23. stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe. Zagadnienia dodatkowe powinny być omawiane z uczniami z lepszym przygotowaniem matematycznym i informatycznym i wykazującymi większe zainteresowanie pogłębieniem znajomości metod rozwiązywania problemów z wykorzystaniem podejścia rekurencyjnego.



2. Temat zajęć

Rekurencja – jakie to proste!

Łagodne wprowadzenie myślenia rekurencyjnego, które:

- ułatwia rozwiązywanie problemów,
- skraca formułowanie rozwiązań (programów) dla komputerów,
- umożliwia lepsze wykorzystanie możliwości komputera.

Komentarz. Jest to dość trudny temat, zarówno dla uczniów, jak i dla nauczyciela, gdyż zrozumienie rekurencji, a zwłaszcza stosowanie rekurencji wymaga osiągnięcia wcześniej pewnego poziomu abstrakcyjnego myślenia, jak i doświadczenia w rozwiązywaniu problemów oraz w programowaniu. Proponowane ujęcie tego tematu w tym scenariuszu zajęć uwzględni wszystkie te aspekty i stara się łagodnie poprowadzić ucznia przez trudności, które mogą wystąpić.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- wykazać się zrozumieniem pojęcia rekurencji, objaśniając je na przykładzie opisu rozwiązania wybranego problemu, w którym wykorzystuje się rekurencję;
- zapisać w postaci rekurencyjnej typowe algorytmy, takie jak: obliczanie wartości funkcji silnia, generowanie kolejnych cyfr liczby przy ustalonej podstawie, algorytm Euklidesa, Wieże Hanoi, schemat Hornera, generowanie liczb Fibonacciego, porządkowanie metodą przez scalanie;
- zapisać w postaci rekurencyjnej wybrane algorytmy służące do generowania obiektów geometrycznych o rekurencyjnym charakterze (fraktale);
- zapisać algorytmy rekurencyjne w wybranym języku programowania;
- wyjaśnić, w jaki sposób procedura rekurencyjna jest wykonywana przez komputer;
- porównać dwie realizacje algorytmu wykonującego to samo zadanie, iteracyjną z rekurencyjną, pod względem: przejrzystości i elegancji zapisu oraz efektywności obliczeń.

Komentarz. Cele tych zajęć wykraczają poza podstawę programową.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów;
- znać podstawowe pojęcia z zakresu informatyki i algorytmiki, takie jak: porządkowanie, reprezentacja liczb przy różnych podstawach, iteracja, funkcja silnia, liczby Fibonacciego, pracochłonność (złożoność) algorytmu;
- umieć przedstawić podstawowe problemy i ich algorytmy, takie jak: wyznaczanie największego wspólnego dzielnika dwóch liczb i algorytm Euklidesa, obliczanie wartości wielomianu i schemat Hornera, Wieże Hanoi, sortowanie i wybrane algorytmy sortowania oraz podać ich iteracyjne realizacje;
- opisywać algorytmy w wybranej postaci: listy kroków, schematu blokowego, w pseudokodzie programowania;

- umieć programować w wybranym, algorytmicznym języku programowania (np. Pascal, C++), w szczególności zapisywać algorytmy zawierające iteracje.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które stanowią metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (algorytmu),
 - implementacja (realizacja) rozwiązania w postaci programu komputerowego,
 - testowanie i ewaluacja rozwiązania komputerowego,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- posłuszenie się przez nauczyciela i uczniów objaśnieniami i demonstracjami przykładowych algorytmów rekurencyjnych za pomocą schematów geometrycznych oraz w edukacyjnych programach komputerowych;
- przygotowanie przez uczniów schematów działania wybranych algorytmów, w tym w postaci rekurencyjnej;
- samodzielne sporządzenie przez uczniów rekurencyjnych opisów algorytmów;
- zapisanie algorytmów rekurencyjnych w postaci procedur i funkcji rekurencyjnych w wybranym języku programowania i uruchomienie odpowiednich programów na komputerze;
- samodzielne testowanie poprawności programów dla odpowiednio dobranych danych;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania algorytmu dla przedstawionego problemu;
- przygotowywanie opisów algorytmów może odbywać się indywidualnie lub w parach uczniów;
- programy dla komputerów uczniowie piszą samodzielnie;
- testowanie programów może odbywać się w grupach uczniów – uczniowie wspólnie poprawiają błędy w programach, dobierają dane i porównują wyniki działania swoich programów;
- końcowym efektem pracy nad danym problemem jest prezentacja opisu algorytmu, programu komputerowego oraz wyników działania programu.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z tablicy lub z tablicy z kartkami papieru, gdzie zapisują specyfikacje problemów, opisy algorytmów, schematy działania algorytmów, fragmenty programów. Przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami.



8. Środki dydaktyczne

Uczniowie (a także nauczyciel) wykorzystują w czasie tych zajęć:

- podręcznik, E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, *Informatyka, zakres rozszerzony*, WSiP, Warszawa 2004.
- publikacje uzupełniające, np.:
 - M. M. Sysło, *Algorytmy*, WSiP, Warszawa 1997; kolejne wydanie z zadaniami maturalnymi z algorytmiki.
 - M. M. Sysło, *Piramidy, szyszki i inne konstrukcje algorytmiczne*, WSiP, Warszawa 1998; w tej książce zawarto opisy wielu sytuacji praktycznych prowadzących do sytuacji problemowych, dla których rekurencyjne metody rozwiązywania są najbardziej naturalne; tę książkę można pobrać ze strony WSiP;
- materiały dotyczące algorytmiki, opracowane w projekcie Informatyka +;
- programy edukacyjne służące do demonstracji działania algorytmów rekurencyjnych – wiele takich programów jest dostępnych w sieci Internet;
- komputer i jego podstawowe oprogramowanie, w tym kompilatory algorytmicznych języków programowania.

9. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenie i omówienie rekurencji, mogą być rozłożone na kilka lekcji, niekoniecznie kolejnych. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel objaśnia rekurencję na przykładach czynności wziętych z życia (Lekcja 1);
- następnie uwaga zostaje skupiona na znanych uczniom problemach i ich rozwiązaniach iteracyjnych, które zostają przedstawione jako rozwiązania rekurencyjne (Lekcja 2);
- na kolejnym etapie uczniowie zajmują się problemami, które mają naturę rekurencyjną w tym sensie, że ich sformułowania i wynikające z nich algorytmy są rekurencyjne (Lekcja 3);
- kolejnymi etapami, które mogą następować w pewnych odstępach czasu, jest wprowadzanie rekurencyjnych rozwiązań dla problemów pojawiających się przy okazji omawiania różnych zagadnień;
- przy powyższym wprowadzaniu i rozszerzaniu zakresu tematu, wiele zagadnień jest omawianych w kontekście problemów i ich algorytmów, najbardziej do nich pasujących;
- proces rozwiązywania każdego z zagadnień powinien składać się z sześciu etapów, wyżej wymienionych (w Metodach pracy).

Szczegółowe materiały, dotyczące rekurencji, uzupełniające ten konspekt, znajdują się w wymienionym powyżej podręczniku i książkach.

Lekcja 1. Ilustracja rekurencji za pomocą przykładów wziętych z życia.

Czas: 30 min.

Ta część zajęć zapewne nie wypełni całej lekcji, ale może stanowić dobre, bardzo luźne wprowadzenie do tematu. Proponujemy przedstawienie w postaci „programów” w pseudojęzyku programowania dwóch czynności, których wykonywanie ma charakter ciągły do pewnego momentu, a więc

iteracyjny, i może być przedstawione w postaci procedur, które **odwołują się do siebie, a to właśnie jest rekurencją**.

Pierwszy przykład z życia – pochodzi od radzieckiego informatyka Andrieja Jerszowa – a dotyczy jedzenia kaszki z talerza. Czynność tę możemy opisać w następujący sposób:

```

procedure Jedz kaszkę;
  if talerz jest pusty then koniec jedzenia
  else begin
    weź łyżkę kaszki;
    Jedz kaszkę
  end

```

Drugi przykład z życia to taniec towarzyski do muzyki, który można zapisać podobnie:

```

procedure Tańcz;
  if nie gra muzyka then koniec tańczenia
  else begin
    zrób krok;
    Tańcz
  end

```

Objaśniając te dwa przykłady i dyskutując o nich z uczniami wskazujemy, po pierwsze – że program będący zapisem rekurencji ma postać procedury (lub funkcji), aby mógł odwoływać się do siebie, a po drugie – każda taka procedura zawiera tzw. **warunek zakończenia rekurencji** (zwany także warunkiem początkowym) oraz przynajmniej jedno miejsce, w którym procedura wywołuje samą siebie.

Można poprosić uczniów, by postarali się wskazać i zapisać jakąś inną czynność życiową w postaci procedury rekurencyjnej. Nie powinni mieć z tym specjalnego kłopotu.

Z tych przykładów powinna wypłynąć konkluzja, iż rekurencja jest zapisem czynności powtarzających, które kończą się po spełnieniu pewnego warunku, warunku zakończenia rekurencji, a dokładniej – warunku zakończenia wywołań rekurencyjnych.

Lekcja 2. Rekurencja jako inna realizacja iteracji. Czas: 60-90 min.

Celem tej lekcji jest zilustrowanie, w jaki sposób znane algorytmy, w których występuje iteracja, można zapisać z wykorzystaniem rekurencji.

Pierwszy przykład – funkcja silnia

Wybrany uczeń przypomina, jak jest definiowana funkcja silnia. Zapewne na tablicy pojawi się wzór:

$$n! = 1 * 2 * \dots * (n - 1) * n$$

Dodatkowo pytamy, jaką wartość mają 1! i 0! – w obu przypadkach przyjmujemy, że: 1! = 1 i 0! = 1. Zwracamy uwagę, że te wartości nie wynikają z powyższego wzoru, trzeba je określić dodatkowo.



Prosimy teraz uczniów o napisanie programu, który będzie służył do obliczania wartości funkcji silnia według powyższego wzoru, i o wykonanie obliczeń dla kilku wartości $n = 5, 10, 15, 25, 30$. Uczniowie powinni zauważyć, jak wartości tej funkcji bardzo szybko rosną. Może to spowodować uwzględnienie w programie odpowiednich typów danych dla przechowywania długich liczb naturalnych.

Staramy się teraz naprowadzić uczniów na rekurencyjną definicję funkcji silnia. Prosimy więc innego ucznia, by napisał, jaką ma postać wzór na $(n - 1)!$ – powinien podać:

$$(n - 1)! = 1 * 2 * \dots * (n - 2) * (n - 1).$$

Stąd już łatwo powinien zauważyć, że

$$n! = (n - 1)! * n$$

a zatem wartość $n!$ wyraża się przez wartość $(n - 1)!$. Aby lepiej zrozumieć ten wzór prosimy kolejnego ucznia o zapisanie na podstawie tego wzoru, jakie wartości funkcji silnia będą obliczane, dla na przykład $n = 6$. Uczeń powinien zacząć wypisywać wartości: 5!, 4!, 3!, 2!, 1!, 0!. Wypisywanie dalszych wartości nie ma już sensu, a więc na tym kończymy, przy czym za 0! przyjmujemy 1, jak ustaliliśmy powyżej. Na ogół ten ciąg wywołań kończy się na 1!. Dochodzimy w ten sposób do następującej, rekurencyjnej definicji funkcji silnia:

$$n! = \begin{cases} 1 & \text{dla } n = 1, \\ (n - 1)! * n & \text{dla } n > 1. \end{cases}$$

Wartość tej funkcji dla $n = 1$ nazywa się **warunkiem początkowym**, który, ze względu na swoją rolę w ciągu wywołań rekurencyjnych, nosi też nazwę **warunku zakończenia rekurencji**.

Prosimy teraz uczniów o napisanie programu, który będzie służył do obliczania wartości funkcji silnia według wzoru rekurencyjnego podanego powyżej. Zwracamy uwagę, że aby program mógł odwoływać się do siebie, musi mieć postać funkcji lub procedury rekurencyjnej. Ponownie, uczniowie wykonują obliczenia testowe ze swoimi programami i porównują wyniki z otrzymanymi za pomocą programów nie korzystających z rekurencji.

Na zakończenie tej części zajęć prosimy dwóch uczniów o zapisanie na tablicy obok siebie programów, będących realizacjami obu sposobów obliczania wartości funkcji silnia, bez użycia rekurencji i z użyciem rekurencji, oraz ich porównanie. Zwracamy uwagę na zwartość i przejrzystość zapisu rekurencyjnego.

Drugi przykład – schemat Hornera

Zajęcia można zacząć od komentarza-uzasadnienia, że wartości wszystkich funkcji, takich jak sin, cos, log są obliczane w komputerze w sposób przybliżony jako wartości pewnych wielomianów lub

ilorazów wielomianów, bo dzięki temu do ich obliczania mogą być użyte tylko podstawowe działania arytmetyczne: dodawanie/odejmowanie, mnożenie i dzielenie, które „potrafi” wykonywać komputer.

Przyjmujemy dalej, że uczniowie wcześniej poznali schemat Hornera, służący do obliczania wartości wielomianu, a celem tych zajęć jest zapisanie tego algorytmu w postaci rekurencyjnej.

Wielomian stopnia n ma postać:

$$w_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$$

i po wielokrotnym grupowaniu wyrazów i wylączeniu x , otrzymujemy następującą postać:

$$\begin{aligned} w_n(x) &= a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = (a_0x^{n-1} + a_1x^{n-2} + \dots + a_{n-1})x + a_n \\ &= ((a_0x^{n-2} + a_1x^{n-3} + \dots + a_{n-2})x + a_{n-1})x + a_n \\ &= (\dots((a_0x + a_1)x + a_2)x + \dots + a_{n-2})x + a_{n-1})x + a_n \end{aligned}$$

z której wynika schemat Hornera dla obliczania wartości wielomiany $w_n(x)$ w punkcie z , czyli $w_n(z)$:

$$\begin{aligned} y &:= a_0 \\ y &:= yz + a_i \quad \text{dla } i = 1, 2, \dots, n. \end{aligned}$$

Wynika stąd, że aby obliczyć wartość wielomianu stopnia n wystarczy wykonać n mnożeń i n dodawań.

Tyle przypomnienia. Przystępujemy do wprowadzenia rekurencji w schemacie Hornera. Zwracamy uwagę uczniów na pierwszy krok w wyprowadzeniu schematu:

$$w_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = (a_0x^{n-1} + a_1x^{n-2} + \dots + a_{n-1})x + a_n$$

i pytamy, jak inaczej można zapisać ten wielomian korzystając z ostatniej postaci. Kierujemy w tym celu uwagę uczniów na wyrażenie w nawiasie – czym ono jest? Przypomina ten sam wielomian, ale o stopniu o 1 mniejszym. A zatem możemy zapisać:

$$w_n(x) = w_{n-1}(x)x + a_n$$

gdzie $w_{n-1}(x) = a_0x^{n-1} + a_1x^{n-2} + \dots + a_{n-1}$. A zatem, wartość wielomianu stopnia n może być obliczona, gdy tylko znamy wartość odpowiedniego wielomianu stopnia $n - 1$ dla tego samego argumentu. Zachodzi to dla każdego n , spełniającego nierówność $n \geq 1$. Dla $n = 1$ mamy $w_1(x) = w_0(x)x + a_1$, gdzie $w_0(x) = a_0$, czyli poprawnie. Powyższy wzór nie może być jednak użyty dla $n = 0$, czyli do znalezienia wartości wielomianu stopnia 0, gdyż po prawej stronie otrzymujemy wielomian stopnia -1 , a takich wielomianów nie ma. Musimy więc przyjąć w tym przypadku, że $w_0(x) = a_0$. Z tej dyskusji wynika, że wielomian stopnia n , dla $n \geq 0$, można zapisać następująco:

$$w_n(x) = \begin{cases} a_0 & n = 0, \\ w_{n-1}(x)x + a_n & n \geq 1. \end{cases}$$

Otrzymaliśmy **zależność rekurencyjną**, w której wielkość definiowana (czyli wielomian stopnia n , dla $n \geq 1$) jest równa wyrażeniu, w którym występuje ta sama wielkość, ale dla parametru mniejszego (wielomian stopnia $n - 1$), dla tego samego argumentu. Dla $n = 0$ wielkość definiowana jest określona wprost – jest to tzw. **warunek początkowy**, czyli **warunek zakończenia rekurencji**, dzięki któremu każdy ciąg odwołań rekurencyjnych jest skończony.

Teraz zadaniem uczniów jest zaprogramowanie rekurencyjnej wersji schematu Hornera i porównanie jej z postacią iteracyjną na przykładowych danych.

Dla pogłębienia rozumienia rekurencji, warto zilustrować, w jaki sposób przebiega wykonywanie algorytmu rekurencyjnego przez komputer. Odpowiednia ilustracja znajduje się w podręczniku i w książce *Algorytmy*. W obliczeniach na podstawie zależności rekurencyjnej można wyróżnić dwa etapy:

1. Ciąg wywołań rekurencyjnych, kończący się skorzystaniem z warunku zakończenia rekurencji.
2. Powrót z kolejnych wywołań, stanowiący właściwe obliczenia.

Uczniowie powinni zauważyć, że ten drugi etap w przypadku schematu Hornera jest realizacją iteracyjnego algorytmu Hornera. Potwierdza to naszą zapowiedź, że rekurencja jest innym sposobem wykonania iteracji i do tego bardziej „rozzutnym”, gdyż dodatkowo jest wykonywany pierwszy etap, który nie ma swojego odpowiednika w algorytmie iteracyjnym.

Dalsze lekcje w ramach zajęć poświęconych rekurencji omawiamy tutaj skrótowo – szczegółowe konspekty powinny być przedmiotem innych dokumentów. Przy poszczególnych tematach zwracamy głównie uwagę na te elementy, które są charakterystyczne dla danego tematu i nie pojawiają się przy innych tematach. Szczegółowe rozważania można znaleźć w podanych wyżej podręcznikach i opracowaniach książkowych.

Lekcja 3. Problemy o rekurencyjnej naturze. Czas: 90 min.

Liczby Fibonacciego

Jest to temat dość naturalny ze względu na stojącą za nim historią i od opisu tej sytuacji problemowej należy rozpocząć zajęcia, chociaż wydaje się, że bardziej naturalną sytuacją jest na przykład chaotyczne chodzenie po schodach (również prowadzące do liczb Fibonacciego), opisane w podręczniku.

Na podstawie opisu sytuacji problemowej uczniowie powinni poradzić sobie z zapisaniem zależności, jakie spełniają liczby Fibonacciego, iteracyjnej i rekurencyjnej, oraz ze sporządzeniem odpowiednich programów służących do obliczania wartości liczb Fibonacciego według tych zależności.

Polecamy również ćwiczenia polegające na znalezieniu liczb Fibonacciego w różnych obiektach przyrody: szyszkach, słonecznikach, pędach dębu itp. Pewną trudnością (dla nauczyciela) w tych ćwiczeniach jest brak wyjaśnienia, dlaczego te liczby pojawiają się w oglądanych obiektach (patrz książka *Piramidy, szyszki, ...* a uczniów można poprosić, by znaleźli w Internecie stronę poświęconą występowaniu liczb Fibonacciego, nie tylko w przyrodzie).

Algorytm Euklidesa

Jest to najstarszy znany algorytm (podany 300 lat p.n.e). Służy on do znajdowania największego wspólnego dzielnika (w skrócie NWD) dwóch liczb. Postać rekurencyjna tego algorytmu jest chyba najbardziej naturalnym sposobem jego przedstawienia. Warto uzasadnić tę postać. Można to zrobić korzystając z oczywistej równości. Jeśli n dzielimy w sposób całkowity przez m (zakładamy, że $n > m$), to otrzymujemy całkowity iloraz q oraz resztę r , która jest mniejsza od dzielnej, czyli od m . A zatem zachodzi równość:

$$n = qm + r, \quad \text{gdzie } 0 \leq r < m.$$

Z tej równości wynika, że jeśli jakaś liczba dzieli n i m , to dzieli również m i r . Także na odwrót, liczba, która dzieli m i r dzieli także n i m . Stąd otrzymujemy zależność rekurencyjną:

$$\text{NWD}(n, m) = \text{NWD}(m, r).$$

Zatem, NWD dzielnej i dzielnika jest równy NWD dzielnika i reszty. Tę zależność można kontynuować, aż dojdziemy do reszty r równej 0, a wtedy $\text{NWD}(m, 0) = m$, bo 0 jest podzielne przez każdą liczbę.

Zanim uczniowie przestąpią do programowania algorytmu Euklidesa warto, by posłużyli się tym algorytmem w obliczeniach dla wybranych liczb n i m . W szczególności polecamy wykonanie algorytmu Euklidesa dla dwóch sąsiednich liczb Fibonacciego.

Wieże Hanoi

To kolejny problem z pogranicza matematyki i rozrywki. Dla zaznajomienia się z tą zagadką, polecamy uczniom demonstrację w sieci: <http://wipos.p.lodz.pl/zylla/games/hanoi5p.html>. Po jakimś czasie zabawy prosimy o sformułowanie strategii – naprowadzamy uzmysławiając, że w każdym momencie najmniejszy krążek jest na górze i można go przenieść na każdy inny palik, dostępny jest również inny krążek, ale można go przenieść tylko na jeden palik, bo nie można położyć go na najmniejszym krążku. Stąd pojawia się strategia iteracyjna. Strategia rekurencyjna jest pewnym ułatwieniem w opisie, chociaż jej wykonanie stwarza kłopoty. Strategię rekurencyjną polecamy uczniom zaprogramować. Program powinien drukować kolejno wykonywane przeniesienia pojedynczych krążków. Wypisane przez program ruchy uczniowie powinni wykonać w programie demonstracyjnym.

Ciekawym zadaniem jest obliczenie, ile w tej grze jest wykonywanych pojedynczych przeniesień krążków. Można posłużyć się odpowiednią zależnością, wynikającą z algorytmu rekurencyjnego i rozwiązać ją metodą przez wstawianie. Szczegóły można znaleźć w książce *Algorytmy*.

Lekcja 4. Iteracja a rekurencja – porównanie efektywności rozwiązań.

Czas: 30 min.

Przy okazji omawiania schematu Hornera, przyjrzenie się, jak jest wykonywana jego rekurencyjna wersja, prowadzi do wniosku, że drugi etap rekurencyjnego algorytmu jest algorytm iteracyjny Hornera. Można więc sądzić, że rekurencja jest zbędna w tym przypadku. Do pewnego stopnia tak – często płacimy większą cenę za wykonanie rekurencyjnego algorytmu w porównaniu z algorytmem iteracyjnym.

Dobitniej ilustruje to rekurencyjna realizacja wzoru na liczby Fibonacciego. By się o tym przekonać, prosimy uczniów o naszkicowanie drzewa wywołań rekurencyjnej funkcji obliczającej wartości liczb Fibonacciego, np. dla $n = 5, 6$. Powinni zauważyć, że te same wartości są wielokrotnie obliczane niezależnie.

Lekcja 5. Dalsze Problemy i naturalne pojawianie się rekurencji w ich rozwiązaniach. Czas: 120 min.

Generowanie kolejnych cyfr liczby

Jest to typowe zadanie informatyczne (programistyczne): dla danej liczby naturalnej n i dla podstawy systemu liczenia p ($2 \leq p \leq 10$), należy napisać program, który będzie drukował kolejne, od najbardziej znaczącej, cyfry liczby n w systemie o podstawie p . Dzięki rekurencji odpowiednia procedura ma bardzo prostą postać – podajemy ją tutaj:

```
procedure KolejnaCyfra (n, p: integer);
  if n < p then write(n)
  else begin
    KolejnaCyfra(n div p);
    write (m mod p)
  end
```



Ciekawym pytaniem do uczniów jest: jak zmodyfikować tę procedurę, by drukowane były cyfry od najmniej znaczącej – odpowiedź: zamienić kolejność dwóch ostatnich instrukcji.

Sortowanie przez scalanie

Rekurencja jest stosowana w wielu algorytmach sortujących ciąg danych liczb. Klasycznym przykładem jest **sortowanie przez scalanie**. Sama procedura jest dość prosta. Zanim jednak uczniowie sięgą do programowania, proponujemy, by wykonali tę procedurę na danym ciągu liczb (ilość liczb nie powinna być potęgą liczby 2) i przedstawili, w jakiej kolejności są wykonywane kroki podziału ciągu na dwie prawie równe części oraz scalania podciągów uporządkowanych. Odpowiednia procedura jest również przykładem, jak prosty może być rekurencyjny zapis algorytmu.

Dziel i zwyciężaj

Sortowanie przez scalanie może posłużyć do wyjaśnić idei metody **dziel i zwyciężaj** – na każdym kroku algorytmu ciąg jest dzielony na dwie części, każda z nich jest osobno porządkowana a tak uporządkowane podciągi są później scalane. Ta ilustracja powinna posłużyć również do zobrazowania, że do wykonania obliczeń na podciągach danych jest stosowany ... ten sam algorytm, dlatego więc możemy zastosować rekurencję w komputerowym opisie algorytmu.

Algorytmy geometryczne – fraktale

Fraktale są często określane jako obiekty, w których każda część wygląda podobnie jak całość. A zatem, te obiekty mają budowę rekurencyjną. Zadania związane z programowaniem i rysowaniem fraktali są bardzo ciekawe dla uczniów, gdyż efektem otrzymanych programów są obiekty geometryczne. Bardzo naturalnie programuje się rysowanie fraktali w języku Logo. Wiele demonstracji powstawania fraktali można znaleźć w sieci – prosimy uczniów, by odnaleźli odpowiednie programy, które służą do demonstracji powstawania fraktali.



LOGARYTM W INFORMATYCE – JEGO WYSTĘPOWANIE I ZNACZENIE

Maciej M. Sysło

Uniwersytet Wrocławski
Uniwersytet Mikołaja Kopernika w Toruniu

syslo@ii.uni.wroc.pl

1. Zagadnienia programowe

Zgodnie z podstawą programową, logarytm i funkcja logarytmiczna pojawiają się w szkole na lekcjach matematyki w szkole ponadgimnazjalnej.

Matematyka, IV etap edukacyjny

ZAKRES PODSTAWOWY	ZAKRES ROZSZERZONY
1. Liczby rzeczywiste	
6) Wykorzystuje definicję logarytmu i stosuje w obliczeniach wzory na logarytm iloczynu, logarytm ilorazu i logarytm potęgi o wykładniku naturalnym.	wymagania określone dla zakresu podstawowego, a ponadto: 2) wykorzystuje definicję logarytmu i stosuje w obliczeniach wzory na logarytm iloczynu, logarytm ilorazu, logarytm potęgi oraz wzór na zamianę podstawy logarytmu.
4. Funkcje	
14) szkicuje wykresy funkcji wykładniczych dla różnych podstaw; 15) posługuje się funkcjami wykładniczymi do opisu zjawisk fizycznych, chemicznych, a także w zagadnieniach osadzonych w kontekście praktycznym.	wymagania określone dla zakresu podstawowego, a ponadto: 2) szkicuje wykresy funkcji logarytmicznych dla różnych podstaw; 3) posługuje się funkcjami logarytmicznymi do opisu zjawisk fizycznych, chemicznych, a także w zagadnieniach osadzonych w kontekście praktycznym;

Jak widać w tabeli, uczeń powinien umieć posługiwać się funkcjami logarytmicznymi w wybranych zastosowaniach i sytuacjach praktycznych. Informatyka nie jest niestety wymieniona wśród możliwych zastosowań, chociaż, jak ilustrujemy, logarytm pojawia się w zagadnieniach informatycznych w sposób bardzo naturalny. Niniejsza propozycja zajęć ma uzupełnić ten brak.

Informatyka, IV etap edukacyjny, zakres rozszerzony

Ani logarytm, ani funkcja logarytmiczna nie występują jawnie w zapisach podstawy programowej dla przedmiotu informatyka. Uwzględniając jednak, że uczniowie powinni znać te pojęcia z lekcji matematyki, na zajęciach z informatyki powinny pojawić się w tych zagadnieniach, w których mogą poszerzyć zrozumienie przedmiotu. Poniżej wymieniamy te zagadnienia z podstawy programowej, których dotyczą nasze propozycje wykorzystania logarytmu i funkcji logarytmicznych.

5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń:
 - 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;
 - 9) stosuje rekurencję w prostych sytuacjach problemowych;
 - 11) opisuje podstawowe algorytmy i stosuje:
 - a. algorytmy na liczbach całkowitych, np.:
 - reprezentacja liczb w dowolnym systemie pozycyjnym, w tym, w dwójkowym i szesnastkowym,
 - iteracyjna i rekurencyjna realizacja algorytmu Euklidesa,
 - b. algorytmy wyszukiwania i porządkowania (sortowania), np.:
 - algorytmy sortowania ciągu liczb: ... przez wstawianie binarne, przez scalanie,
 - c. algorytmy numeryczne, np.:
 - obliczanie wartości wielomianu za pomocą schematu Hornera,
 - zastosowania schematu Hornera: reprezentacja liczb w różnych systemach liczbowych, szybkie podnoszenie do potęgi,
 - e. algorytmy kompresji i szyfrowania, np.:
 - szyfr z kluczem jawnym (RSA),
 - 18) oblicza liczbę operacji wykonywanych przez algorytm;

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe. Zagadnienia dodatkowe powinny być omawiane z uczniami z lepszym przygotowaniem matematycznym i informatycznym i wykazującymi większe zainteresowania pogłębieniem znajomości metod rozwiązywania problemów z ich własności, w szczególności złożoności obliczeniowej.

2. Temat zajęć

Logarytm w informatyce – jego występowanie i znaczenie

Zajęcia te mają na celu zwrócenie uwagi uczniów na złożoność (pracochłonność) algorytmów, czyli liczbę wykonywanych działań, a także wielkość zajmowanej przez dane pamięci. Przedstawione zagadnienia ilustrują m.in., że szybkość działania efektywnych algorytmów komputerowych jest związana z występowaniem funkcji logarytmicznej we wzorach na liczbę wykonywanych przez algorytm operacji.

Dodatkowym celem tych zajęć jest praktyczne wykorzystanie logarytmów w zagadnieniach algorytmicznych¹. Faktycznie, to wykorzystanie logarytmów może stanowić inny sposób ich wprowadzenia, bardziej intuicyjny, podparty praktycznym wykorzystaniem.

¹ Ciekawe spostrzeżenie – słowo algorytm jest anagramem słowa logarytm. Podobnie jest w języku angielskim. Te obiekty pojawiły się jednak niezależnie od siebie, chociaż logarytm został wprowadzony, by ułatwić wykonywanie rachunków (algorytmów) na dużych liczbach.

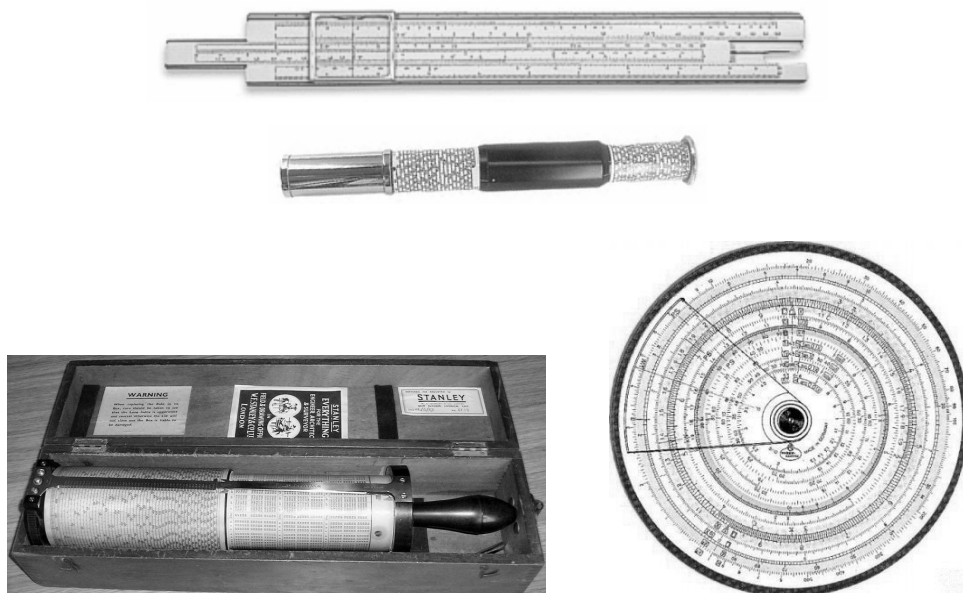
Przedstawione własności logarytmów i sytuacje algorytmiczne, w których się pojawiają, powinny wzbogacić nieraz suche wprowadzenie i stosowanie logarytmów na lekcjach matematyki.

Komentarz, uwagi historyczne

Funkcje nie są obiektami matematycznymi lubianymi przez uczniów zwłaszcza, gdy są wprowadzane w sposób formalny jako przekształcenia (odwzorowania). Wśród nich, „najgorszą sławą” cieszy się logarytm i funkcja logarymiczna, bo nie dość, że jest to funkcja, to jeszcze jest to funkcja odwrotna. Odwołanie się na informatyce do funkcji logarymicznej, wprowadzonej na matematyce „na sucho”, na ogół nie poprawia tej sytuacji.

W przeszłości, uzasadnieniem dla posługiwania się logarytmem, były jego własności, które legły u podstaw jego wprowadzenia do matematyki, a dokładniej – do obliczeń. Logarytm ułatwia wykonywanie złożonych obliczeń, na przykład dzięki zastąpieniu działań multiplikatywnych, takich jak mnożenie i dzielenie, przez dodawanie i odejmowanie. Nie tak dawno jeszcze, w szkołach posługiwano się tablicami logarymicznymi, a w uczelniach i w pracy przyszli i zawodowi inżynierowie korzystali z suwaków logarymicznych.

Za odkrywcę logarytmu uznaje się matematyka szkockiego Johna Napiera (1550-1617), a suwak logarymiczny wynalazł William Oughtred (1575-1660) w 1632 roku. Na Rysunku 1 są pokazane różne typy suwaków: prosty, cylindryczny (model Otis King), na walcu (model Fullera) i okrągły. Różnią się one długością skali, od czego zależy dokładność obliczeń – prosty ma skalę o długości 30 cm, cylindryczny – 1,5 metra, a na walcu – ponad 12 metrów.



Rysunek 1. Suwaki logarymiczne: prosty, cylindryczny, na walcu i okrągły

Rok 1972 to początek agonii suwaków – zaczęły je wypierać stworzone z pomocą suwaków kalkulatory elektroniczne. Ponad 40 milionów wcześniej wyprodukowanych suwaków stało się nagle bezu-

żytecznych i obecnie stanowią głównie eksponaty kolekcjonerskie, jak te na ilustracjach (należą one do kolekcji autora).

Dzisiaj jednak nie można wyobrazić sobie zajmowania się informatyką, nawet na najniższym poziomie w szkole, bez przynajmniej „otarcia” się o logarytmy, pośredni kontakt.

Przedstawione propozycje tematów zajęć i ich realizacji służyć mogą wyjaśnieniu, co wspólnego mają ze sobą rozważania następujących pytań i odpowiedzi na nie:

- ile należy przejrzeć kartek w słowniku, aby znaleźć poszukiwane hasło?
- ile miejsca w komputerze, a dokładniej – bitów, zajmuje w komputerze liczba naturalna?
- jak szybko można wykonywać potęgowanie dla dużych potęg?
- Ile trwa obliczanie największego wspólnego dzielnika dwóch liczb za pomocą algorytmu Euklidesa?
- a ogólniej – ile kroków wykonuje algorytm typu dzieli i zwyciężaj, zastosowany do danych o n elementach?

Wspólną cechą odpowiedzi na te pytania jest to, że nie można ich udzielić, by nie dotknąć logarytmu.

Wydaje się jednak, że właśnie praktyczne wykorzystanie funkcji logarytmicznej w zajęciach informatycznych może stanowić motywację dla zrozumienia, co to jest logarytm i funkcja logarytmiczna, i jakie może mieć praktyczne zastosowanie w obliczeniach komputerowych.



3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- wykazać się pogłębionym zrozumieniem pojęć logarytm i funkcja logarytmiczna;
- określić długość reprezentacji liczb w systemie pozycyjnym, w szczególności w systemie binarnym;
- oszacować liczbę operacji w algorytmach typu dzieli i zwyciężaj;
- uzasadnić złożoność algorytmu Euklidesa;
- wykonywać szybkie potęgowanie;
- wskazać efektywne metody porządkowania liczb;
- a przede wszystkim wykształcić intuicję, która powinna podpowiadać mu, że jeśli algorytm polega na sukcesywnym podziale zbioru danych na przynajmniej dwie prawie równe części, to liczba powtórzeń (iteracji) w takim algorytmie jest w przybliżeniu równa logarytmowi przy podstawie 2 (lub większej) z liczby danych.

4. Przygotowanie uczniów

Ucniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów;
- znać podstawowe pojęcia z zakresu informatyki i algorytmiki, takie jak: porządkowanie, reprezentacja liczb przy różnych podstawach, iteracja, pracochłonność (złożoność) algorytmu;
- umieć przedstawić podstawowe problemy i ich algorytmy, takie jak: zamiana reprezentacji liczb przy równych podstawach, wyznaczanie największego wspólnego dzielnika dwóch liczb i algorytm Euklidesa, obliczanie wartości wielomianu i schemat Hornera, sortowanie i wybrane algorytmy sortowania;

- opisywać algorytmy w wybranej postaci: listy kroków, schematu blokowego, w pseudojęzyku programowania;
- umieć programować w wybranym, algorytmicznym języku programowania (np. Pascal, C++);
- z zakresu matematyki – znać pojęcie logarytmu i funkcji logarytmicznej.

Ten ostatni wymóg może nie być spełniony. Jednym z celów tej propozycji zajęć jest przybliżenie uczniom pojęcia logarytm, mogą więc poznać to pojęcie od strony praktycznej, a później dopiero sprecyzować je na gruncie matematyki.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, jeśli na zajęciach pojawia się problem, który ma być rozwiązany z pomocą komputera, to proces jego rozwiązywania powinien się składać z następujących sześciu etapów:
 - opis, dyskusja i zrozumienie sytuacji problemowej;
 - podanie specyfikacji problemów do rozwiązania;
 - zaprojektowanie rozwiązania (algorytmu);
 - implementacja (realizacja) rozwiązania w postaci programu komputerowego;
 - testowanie i ewaluacja rozwiązania komputerowego;
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania.
- posłużenie się przez nauczyciela i uczniów objaśnieniami, demonstracjami i przykładami obliczeń i algorytmów, które służą wprowadzeniu i objaśnieniu pojęć;
- samodzielne sporządzenie przez uczniów opisów algorytmów;
- zapisanie algorytmów w wybranym języku programowania i uruchomienie odpowiednich programów na komputerze;
- samodzielna analiza pracochłonności wybranych algorytmów;
- prezentacja otrzymanych rezultatów i rozwiązań problemów.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania odpowiedzi na stawiane pytania lub algorytmicznego rozwiązania rozważanego problemu;
- przygotowywanie opisów algorytmów może odbywać się indywidualnie lub w parach uczniów;
- programy dla komputerów uczniowie piszą samodzielnie;
- końcowym efektem pracy nad danym pytaniem lub problemem jest prezentacja opisu rozwiązania i uzasadnienie jego poprawności oraz wyznaczenie pracochłonności.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z tablicy lub z tablicy z kartkami papieru, gdzie zapisują pytania lub specyfikacje problemów, proponowane odpowiedzi lub opisy algorytmów, schematy działania algorytmów, fragmenty programów, obliczenia pracochłonności algorytmów.



8. Środki dydaktyczne

Uczniowie (a także nauczyciel) wykorzystują w czasie tych zajęć:

- podręcznik E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, *Informatyka, zakres rozszerzony*, WSiP, Warszawa 2004.
- publikacje uzupełniające, np.:
 - M. M. Sysło, *Algorytmy*, WSiP, Warszawa 1997; kolejne wydanie z zadaniami maturalnymi z algorytmiki.
 - M. M. Sysło, *Piramidy, szyszki i inne konstrukcje algorytmiczne*, WSiP, Warszawa 1998; w tej książce zawarto opisy wielu sytuacji praktycznych prowadzących do sytuacji problemowych, dla których rekurencyjne metody rozwiązywania są najbardziej naturalne; tę książkę można pobrać ze strony WSiP;
- materiały dotyczące algorytmiki, opracowane w projekcie Informatyka +;
- programy edukacyjne służące do demonstracji działania algorytmów – wiele takich programów jest dostępnych w sieci Internet;
- komputer i jego podstawowe oprogramowanie, w tym kompilatory algorytmicznych języków programowania.

9. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest przybliżenie z punktu widzenia informatyki pojęcia logarytm i funkcja logarytmiczna, mogą być rozłożone na kilka lekcji, niekoniecznie kolejnych. Najlepszym wyborem jest uzupełnienie przedstawionymi rozważaniami zajęć, które są poświęcone opisanym tutaj zagadnieniom. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, z zastrzeżeniem podanym na końcu punktu 4, dobrze jest przypomnieć pojęcie logarytmu i jego użycie w obliczeniach; warto również uzmysłowić uczniom, jak wolno rośnie funkcja logarytm (Lekcja 1);
- pojęcie logarytmu przy podstawie 2 można wprowadzić nieformalnie posługując się algorytmem poszukiwania przez połowienie (Lekcja 2);
- kolejnym zagadnieniem jest określenie, ile miejsca (bitów) w komputerze zajmuje liczba naturalna – te rozważania można łatwo rozszerzyć na system pozycyjny przy innej niż 2 podstawie (Lekcja 3);
- jednym z ważniejszych miejsc, w których pojawia się logarytm jest określenie, jak szybko można podnosić do potęgi przy dużych wykładnikach (Lekcja 4);
- bardzo ciekawe dla uczniów może być bliższe przyjrzenie się, jak działa algorytm Euklidesa, i wyciągnięcie konkluzji co do jego związków z logarytmem (Lekcja 5);
- na koniec proponujemy krótkie spojrzenie na inne algorytmy typu dziel i zwyciężaj i określenie ich pracochłonności (Lekcja 6).

Szczegółowe materiały dotyczące analizy pracochłonności algorytmów, uzupełniające ten materiał, znajdują się w wymienionym powyżej podręczniku i książkach.

Lekcja 1. Co to jest logarytm. Czas: 30 min.

Nie musi to być osobna lekcja. Chcemy jedynie przypomnieć co to jest logarytm i funkcje logarytmiczne tym uczniom, którzy poznali te pojęcia na lekcjach matematyki. Można w tym celu poprosić nauczyciela matematyki, by przedstawił, jak i w jakim zakresie wprowadził te pojęcia, by rzeczywiście było to przypomnieniem.

Ważne jest, by uczeń rozumiał, że log jest oznaczeniem operacji (funkcji) odwrotnej do potęgowania. Dobrze jest zacząć od oczywistych przykładów, np.:

$$2^4 = 16, \text{ więc } \log_2 16 = 4,$$

$$2^{10} = 1024, \text{ więc } \log_2 1024 = 10,$$

$$\text{podobnie } 3^3 = 81, \text{ więc } \log_3 81 = 3.$$

Praktycznie, drugie równości w tych wierszach można traktować, jako „wzięcie logarytmu” z obu stron tych pierwszych równości, przy czym podstawa logarytmu powinna być równa liczbie potęgowanej. Korzysta się przy tym z własności logarytmów:

$$\log_p n^b = b \log_p n \quad \text{i} \quad \log_p p = 1.$$

Te przykłady można podsumować stwierdzeniem, że:

$$\text{wartość } \log_p n \text{ jest równa takiej liczbie } x, \text{ że } p^x = n.$$

O znaczeniu i „potędze” logarytmów i funkcji logarytmicznej w informatyce, a ogólniej – w obliczeniach decyduje szybkość wzrostu jej wartości, **nieporównywalnie mała** względem szybkości wzrostu jej argumentu, co ilustrujemy w Tabeli 1.

Tabela 1. Wartości funkcji logarytm dla przykładowych argumentów

n	$\log_2 n$
128	7
1 024	10
65 536	16
1 048 576	20
10^{10}	ok. 34
10^{20}	ok. 67
10^{30}	ok. 100
10^{50}	ok. 167
10^{100}	ok. 333

A zatem, dla liczb, które mają ok. sto cyfr, wartość logarytmu wynosi **tylko** ok. 333.

Można poprosić uczniów o sporządzenie podobnej tabeli. Mogą posłużyć się kalkulatorem w systemie Windows. Ponieważ jednak nie zawiera on funkcji logarytm przy podstawie 2, to ćwiczenie może być okazją do przypomnienia sobie (lub wprowadzenia) wzoru na zamianę podstawy logarytmu.

Dalsze tematy zajęć służą głównie pogłębieniu rozumienia pojęcia logarytm na przykładach praktycznych sytuacji algorytmicznych.

Lekcja 2. Jak szybko można znaleźć hasło w papierowej encyklopedii.

Czas: 30 min.

Nadal wydawane są na papierze słowniki, encyklopedie, książki telefoniczne i wiele innych książek, w których informacje są uporządkowane, na ogół alfabetycznie. Można rozpocząć realizację tego tematu od dyskusji z uczniami o ich doświadczeniach w korzystaniu z takich książek – czy mają

w domu, czy zaglądną do nich, a jeśli tak – to jak często, w jaki sposób znajdują poszukiwane hasła, jaką widzą przyszłość przed takimi książkami.

Można również zadać uczniom prowokujące pytanie: jeśli masz przed sobą komputer, a za sobą na półce papierową encyklopedię, to w jakich przypadkach korzystasz z encyklopedii papierowej, a w jakich z tej w komputerze lub na płycie? Od swojego Syna usłyszałem, że do papierowej encyklopedii lub słownika sięga, gdy chce coś szybko znaleźć, a do komputerowej – gdy chce ... skopiować!

Dalsza dyskusja powinna dotyczyć strategii poszukiwań w encyklopedii, w słowniku lub w książce telefonicznej. Załóżmy, że jest to książka telefoniczna prywatnych posiadaczy telefonów, zawierająca 1200 stron.

Strategia poszukiwania numeru telefonu osoby o danym nazwisku zależy, od jakiej litery zaczyna się nazwisko. Jeśli jest to początkowa litera alfabetu, to sięgamy do kartek bliżej początku książki, a jeśli środkowa – to blisko środka. Aby nie uzależniać się od danych, proponujemy uczniom, by podali metodę, która nie zależy od miejsca w książce nazwiska osoby, której numer telefonu chcemy znaleźć.

Aby naprowadzić uczniów na oczekiwane przez nas rozwiązanie, zamiast poszukiwania informacji w zbiorach uporządkowanych można posłużyć się grą w zgadywanie liczby²: jedna osoba ukrywa liczbę w zadanym przedziale, a druga ma ją odgadnąć. Na podany przez osobę odgadującą wybór liczby, osoba, która ukryła liczbę, może odpowiadać tylko Tak, Za mała, Za duża.

Wydaje się, że we wszystkich przypadkach, strategia podziału zbioru na dwie połowy (lub prawie równe części) jest najlepszą strategią. Czyli na każdym kroku sprawdzamy w książce telefonicznej, jakie nazwisko jest na stronie leżącej w połowie pozostałej do przeszukania części książki i albo znajdujemy poszukiwane nazwisko na odkrytej stronie, albo pozostawiamy tylko tę część (połowę), w której może znajdować się poszukiwane nazwisko. W najgorszym razie musimy tak długo prowadzić poszukiwania odpowiedniej strony, aż pozostanie tylko jedna strona, i albo jest na niej poszukiwane nazwisko, albo nie ma. Jest to tak zwane **poszukiwanie przez połowienie**, zwane również **przeszukiwaniem binarnym**.

Teraz zadajemy uczniom pytanie, ile należało przejrzeć stron książki w najgorszym przypadku, tzn. gdy na końcu zostaje nam jedna strona. Pomocą w odpowiedzi może być prześledzenie ciągu liczby stron pozostałych do przeszukania w kolejnych krokach poszukiwania, na przykład dla 1200 stron. Jest to ciąg (wybieramy „większą połowę”, jeśli liczba pozostałych stron jest nieparzysta):

$$1200, 600, 300, 150, 75, 38, 19, 10, 5, 3, 2, 1.$$

A więc należało zajrzeć na 11 stron. Jaki jest związek tych dwóch liczb – 1200 i 11?

Proponujemy najpierw prostsze ćwiczenie. Przypuśćmy, że książka ma $n = 32$ strony. Wtedy jedna strona pozostaje po pięciu podziałach kolejnych liczb poczynając od liczby 32. a jeśli mamy książkę o 16 stronach, to po czterech podziałach. Stąd można wywnioskować, że jeśli mamy książkę o liczbie stron między 16 a 32, to wykonamy nie więcej niż pięć porównań. Jeśli $n = 1200$, to liczba porównań stron w poszukiwaniu przez połowienie wyniesie 11, gdyż:

$$2^{10} < 1200 \leq 2^{11},$$

co już zaobserwowaliśmy.

² Ta gra jest dość szczegółowo opisana w książce *Piramidy, szyszki i inne konstrukcje algorytmiczne*.

Jeśli uczniowie znają już funkcję logarytm, to można przeprowadzić nieco inne rozumowanie. Najpierw powinni zauważyć, że dzielenie kolejno przez 2 jest w rzeczywistości dzieleniem liczby n przez potęgę liczby 2. W przypadku, gdy $n = 1200$, pytanie o liczbę kroków w poszukiwaniu przez połowie- nie jest więc pytaniem o wartość liczby k takiej, że:

$$1200/2^k = 1.$$

Mnożymy obie strony przez 2^k i otrzymujemy:

$$1200 = 2^k.$$

Stąd na podstawie definicji logarytmu (patrz lekcja 1) mamy:

$$k = \log_2 1200.$$

Ponieważ k ma być liczbą całkowitą, więc przyjmujemy za prawą stronę najmniejszą potęgę liczby 2, która jest większa lub równa 1200. Wiemy, że $2^{10} = 1024$ i $2^{11} = 2048$, a zatem $k = 11$ i tyle kroków wykonałoby w naszym przykładowym przeszukiwaniu binarnym. Pojawia się więc tutaj w naturalny sposób funkcja odwrotna do potęgowania – **logarytm**. Można nawet przyjąć „**informatyczną**” definicję tej funkcji:

$\log_2 n$ jest równy liczbie kroków prowadzących od n do 1, w których bieżąca liczba jest zastępowana przez zaokrąglenie w górę jej połowy

lub

logarytm z liczby n przy podstawie 2 wynosi tyle, ile razy trzeba podzielić n , aby otrzymać 1.

Jednym z celów tych zajęć jest wykształcenie intuicji, która powinna podpowiadać, że jeśli pewien proces polega na sukcesywnym podziale zbioru danych na przynajmniej dwie prawie równe części, to liczba powtórzeń takiego procesu jest równa logarytmowi przy podstawie 2 z liczby danych.

Innym celem tej lekcji jest przekonanie uczniów, że:

integralną częścią informacji jest jej uporządkowanie,

gdyż w przeciwnym razie ... nie jest to informacja. To stwierdzenie nie jest naukowym określeniem informacji³, ale odnosi się do informacji w potocznym znaczeniu. Aby to zilustrować dobitniej, prosimy uczniów, by porównali:

- w alfabetycznym spisie telefonów na 1200 stronach wystarczy przejrzeć co najwyżej 11 stron, by znaleźć numer telefonu danej osoby;
- a jeśli trzeba znaleźć osobę, która ma telefon o numerze 1234567, to w najgorszym przypadku musimy przejrzeć wszystkie 1200 stron.

Czy to porównanie nie świadczy o potędze uporządkowania i o sile algorytmu zastosowanego do uporządkowanego wykazu?

Przedstawiony sposób poszukiwania przez połowie- nie w zbiorze uporządkowanym jest przykładem zastosowania **zasady dziel i zwyciężaj**, o której będzie jeszcze dalej mowa.

³ W teorii informacji, informacja jest definiowana jako „miara niepewności zajścia pewnego zdarzenia spośród skończonego zbioru zdarzeń możliwych” – na podstawie *Nowej encyklopedii powszechnej PWN*.

Poszukiwanie interpolacyjne, czyli poszukiwania w słownikach

Czy rzeczywiście przeszukiwanie binarne jest najszybszą metodą znajdowania elementu w ciągu uporządkowanym? Wyobraźmy sobie, że mamy znaleźć w książce telefonicznej numer telefonu Pana Bogusza Alfreda. Wtedy zapewne skorzystamy z tego, że litera B jest blisko początku alfabetu i, owszem, zastosujemy metodę podziału, ale w pierwszej próbie nie będziemy jednak dzielić książki na dwie połowy, ale raczej spróbujemy trafić blisko tych stron, na których znajdują się nazwiska zaczynające się na literę B. W dalszych krokach będziemy postępować podobnie. Tę obserwację można wykorzystać w algorytmie poszukiwania. Zauważmy najpierw, że w algorytmach binarnych jest sprawdzana jedynie relacja, **czy** dana liczba jest większa (lub mniejsza lub równa) od wybranej z ciągu, natomiast nie sprawdzamy i nie wykorzystujemy tego, **jak bardzo** jest większa. Podczas odnajdywania wyrazów w encyklopediach korzystamy natomiast z informacji, w jakim miejscu alfabetu znajduje się litera, którą rozpoczyna się poszukiwany wyraz, i w zależności od tego wybieramy odpowiednią porcję kartek. Strategia ta nazywa się **interpolacyjnym poszukiwaniem**, gdyż uwzględnia nie tylko położeniu szukanej liczby względem środka ciągu, ale uwzględnia jej wartość względem rozpiętości krańcowych wartości w ciągu. Szczegółowe informacje na temat interpolacyjnego poszukiwania można znaleźć w książce *Algorytmy*.

Lekcja 3. Ile miejsca zajmują liczby w komputerze. Czas: 60 min.

Uczniowie zapewne znają sposób otrzymywania binarnej reprezentacji liczby. Na wszelki wypadek nie zaszkodzi, jeśli jeden z nich przypomni wszystkim, jak jest generowany ciąg bitów w binarnym rozwinięciu liczby. Tradycyjnie, algorytm generowania kolejnych bitów liczby przedstawia się w postaci słupka, jak ilustrujemy to w Tabeli 2.

Tabela 2. Przykład wyznaczania reprezentacji binarnej liczby

Dana liczba i kolejne ilorazy	Podstawa systemu dzielnik	Reszta
57	2	1
28	2	0
14	2	0
7	2	1
3	2	1
1	2	1
0		

Otrzymaliśmy więc: $57 = (111001)_2$. Ważne, by uczniowie zauważyli, że rozwinięcie binarne liczby otrzymane tą metodą jest generowane od najmniej znaczącego bitu.

Sposób generowania binarnej reprezentacji liczby przypomina algorytm z poprzedniej lekcji – dzielimy daną liczbę przez 2 tak długo, aż otrzymamy iloraz 0. A zatem długość binarnej reprezentacji liczby n wynosi około $\log_2 n$. Można to policzyć dokładniej w trochę inny sposób. Zapytajmy uczniów, jaka jest największa liczba, jaką można zapisać na k bitach? Ta liczba ma oczywiście k bitów, każdy równy 1, a zatem taka liczba ma wartość:

$$(111\dots1)_2 = 2^{k-1} + 2^{k-2} + \dots + 2^2 + 2^1 + 2^0,$$

gdyż kolejne bity odpowiadają kolejnej potęgze liczby 2 i tych potęg jest k , od 0 do $k - 1$. Aby obliczyć wartość po prawej stronie wystarczy zauważyć, że ta liczba jest o 1 mniejsza od następnej potęgi liczby 2, a zatem otrzymujemy, że największa liczba zapisana na k bitach ma wartość:

$$(111\dots1)_2 = 2^k - 1.$$

A zatem inne liczby n zapisane na k bitach spełniają nierówności:

$$2^{k-1} - 1 < n \leq 2^k - 1,$$

gdzie ta nierówność z lewej strony wynika z faktu, że liczba ma zajmować k bitów a nie mniej. Stąd otrzymujemy, że liczba bitów potrzebnych do zapisania liczby n , jest najmniejszą liczbą k spełniającą nierówność:

$$\log_2(n + 1) \leq k.$$

Jako ćwiczenie można zapytać uczniów, ile pozycji będzie miała liczba dziesiętna n zapisana w systemie pozycyjnym o podstawie p , dla p spełniającego nierówności $2 \leq p \leq 10$.

Innym ćwiczeniem może być napisanie programu, który generuje kolejne cyfry rozwinięcia danej liczby n w systemie pozycyjnym o podstawie p , $2 \leq p \leq 10$. Nieco większym wyzwaniem jest napisanie takiego programu, który generuje kolejne cyfry, ale od najbardziej znaczącej. Piszemy o tym w innym scenariuszu zajęć, poświęconym rekurencji.

Lekcja 4. Jak szybko można podnosić do potęgi. Czas: 60 min.

Podnoszenie do potęgi jest bardzo prostym, szkolnym zadaniem. Na przykład, aby obliczyć 3^4 , wykonujemy trzy mnożenia $3 \cdot 3 \cdot 3$. A zatem w ogólności, aby obliczyć wartość potęgi x^n tym sposobem należy wykonać $n - 1$ mnożeń, o jedno mniej niż wynosi wykładnik potęgi – ten algorytm będziemy nazywali **algorytmem szkolnym**. Czy ten algorytm jest na tyle szybki, by obliczyć na przykład wartość następującej potęgi:

$$x^{12345678912345678912345678912345678912345}$$

która może pojawić się przy szyfrowaniu metodą RSA informacji przesyłanych w Internecie?

Spróbujmy obliczyć, ile czasu będzie trwało obliczanie powyższej potęgi stosując szkolny algorytm, czyli ile czasu zabierze wykonanie $12345678912345678912345678912344$ mnożeń. Przypuśćmy, że dysponujemy superkomputerem, czyli obecnie najszybszym komputerem. Taki komputer w 2010 roku miał szybkość ok. 1 PFlops, czyli wykonywał 10^{15} operacji na sekundę. A zatem, obliczenie powyższej potęgi będzie trwało (poniższe obliczenia powinni wykonać uczniowie, posługując się np. kalkulatorem w systemie Windows):

$$12345678912345678,912345678912344 \text{ sekund}$$

$$12345678912345678,912345678912344/60 = 205761315205761,31520576131520567 \text{ minut}$$

$$205761315205761,31520576131520567/60 = 3429355253429,3552534293552534278 \text{ godzin}$$

$$3429355253429,3552534293552534278/24 = 142889802226,22313555955646889282 \text{ dób}$$

$$142889802226,22313555955646889282/365 = 391478910,20883050838234649011733 \text{ lat,}$$

czyli około $4 \cdot 10^8$ lat. Jeśli taki algorytm byłby stosowany do szyfrowania naszej poczty w Internecie, to nigdy nie otrzymalibyśmy żadnego listu. Tutaj trzeba dodać, że w praktycznych sytuacjach muszą być obliczane potęgi o wykładnikach, które mają kilkaset cyfr.

Ta lekcja jest poświęcona algorytmom, które służą do szybkiego obliczania wartości potęg.



Rozkład wykładnika na sumę potęg liczby 2

Przypuśćmy, że chcemy obliczyć wartość potęgi x^{22} . Proste wymnożenie podstawy przez siebie $x^{22} = x \cdot x \cdot \dots \cdot x$ to 21 mnożeń. Uczniowie powinni zauważyć, że gdyby wykładnik był potęgą liczby 2, na przykład x^{16} , to wartość tej potęgi można obliczyć przez kolejne podnoszenie do potęgi jako $x^{16} = ((x^2)^2)^2$, a zatem zamiast 15 mnożeń szkolną metodą są wykonywane tylko 4 mnożenia – podnoszenie do kwadratu to jedno mnożenie. Jeśli wykładnik nie jest potęgą liczby 2, to możemy go rozłożyć na sumę potęg liczby 2, czyli sprowadzić do postaci binarnej. Dla wykładnika potęgi 22 mamy – obliczenia, które następują uczniowie powinni wykonać samodzielnie:

$$22 = 2 + 4 + 16 = 2^1 + 2^2 + 2^4,$$

wtedy nasza potęga przyjmuje postać $x^{22} = x^{2+4+16} = x^2 \cdot x^4 \cdot x^{16}$ i możemy ją obliczyć wielokrotnie podnosząc do kwadratu podstawę x i mnożąc przez siebie odpowiednie czynniki. A zatem w tym przypadku obliczanie potęgi przebiega następująco:

$x^2, x^4 = (x^2)^2, x^8 = (x^4)^2, x^{16} = (x^8)^2$ i mnożymy przez siebie $x^2 \cdot x^4 \cdot x^{16}$, co jest równe x^{22} .

Wykonaliśmy 6 mnożeń.

Wykorzystanie postaci binarnej wykładnika i schematu Hornera

Inny sposób potęgowania wykorzystuje przedstawienie wykładnika w postaci binarnej $22 = (101110)_2$ i posłużenie się schematem Hornera do zapisania wykładnika. W naszym przypadku mamy:

$$22 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (((2 + 0)2 + 1)2 + 1)2 + 0$$

i po wstawieniu tego wykładnika do potęgi i skorzystaniu z własności potęgowania otrzymujemy kolejno – te obliczenia uczniowie powinni również wykonać samodzielnie:

$$\begin{aligned} x^{(((2+0)2+1)2+1)2+0} &= (x^{((2+0)2+1)2+1})^2 = (x^{(2+0)2+1})^2 = (x^{2+0+1})^2 x^2 \\ &= (x^{(2+0)2} x^2)^2 = (x^{(2+0)2} x^2)^2 = (((x^2)^2 x)^2)^2. \end{aligned}$$

Korzystając z tego zapisu, wartość potęgi x^{22} obliczamy wykonując kolejno następujące mnożenia:

$$x^2, x^4 = (x^2)^2, x^5 = x^4 \cdot x, x^{10} = (x^5)^2, x^{11} = x^{10} \cdot x, x^{22} = (x^{11})^2.$$

Wykonujemy również 6 mnożeń, ale obliczane są inne potęgi niż w poprzednim algorytmie.

Jako ćwiczenie, prosimy uczniów o podanie kolejności wykonywania mnożeń podczas obliczania wartości potęgi x^{313} oraz obliczenie, ile mnożeń jest wykonywanych w tym przypadku.

Posłużenie się rekurencją

Ta część zajęć może być przeprowadzona, gdy uczniowie znają już rekurencję. Może również posłużyć do wprowadzenia rekurencji.

Zauważmy, że jeśli n jest liczbą parzystą, to zamiast obliczać wartość potęgi x^n , wystarczy obliczyć $y = x^{n/2}$ a następnie ponieść y do kwadratu. Jeśli n jest liczbą nieparzystą, to $n - 1$ jest liczbą parzystą. A zatem mamy następującą zależność:

$$x^n = \begin{cases} 1 & \text{jeśli } n = 0, \\ (x^{n/2})^2 & \text{jeśli } n \text{ jest liczbą parzystą,} \\ (x^{n-1})x & \text{jeśli } n \text{ jest liczbą nieparzystą.} \end{cases}$$

która ma charakter **rekurencyjny** – po prawej stronie równości są odwołania do potęgowania, czyli do tej samej operacji, której wartości liczymy, ale dla mniejszych wykładników. Pierwszy wiersz

w powyższej równości to tzw. **warunek początkowy** – służy do zakończenia odwołań rekurencyjnych dla coraz mniejszych wykładników, aby cały proces obliczeń zakończył się.

Prosimy uczniów, by przedstawili obliczenia potęgi x^{22} tą metodą. Powinni otrzymać następujący ciąg przekształceń:

$$x^{22} = (x^{11})^2 = ((x^5)^2 x)^2 = (((x^2)^2 x)^2 x)^2$$

a zatem z tego wzoru wynika, że wartość potęgi x^{22} jest obliczana w następujący sposób:

$$x^2, x^4 = (x^2)^2, x^5 = x^4 \cdot x, x^{10} = (x^5)^2, x^{11} = x^{10} \cdot x, x^{22} = (x^{11})^2,$$

czyli tak samo, jak w metodzie, polegającej na rozkładzie wykładnika 22 na postać binarną.

Można poprosić uczniów, by zaprogramowali przedstawione algorytmy – szczegółowe opisy tych algorytmów w postaci listy kroków znajdują się w książce *Algorytmy*. Koniecznie powinna pojawić się również komputerowa realizacja algorytmu rekurencyjnego, gdyż jest ilustracją, jak zwięzyły może być zapis algorytmu rekurencyjnego. Realizacja algorytmu rekurencyjnego może mieć postać:

```
function potega (n:integer):real;
begin
  if n = 0 then potega:=1
  else if n mod 2 = 0 then potega:=potega(n div 2)^2
  else potega:=potega (n-1) *x
```

Wracamy teraz do obliczania wartości potęgi x^n dla dużych wartości wykładnika n . Z przedstawionych przykładów wynika, że proponowane algorytm wykonują znacznie mniej mnożeń, niż szkolny algorytm. Czy można określić dokładniej, ile? To nie jest specjalnie trudne. Przeprowadzimy odpowiednie rozumowanie dla algorytmu rekurencyjnego, będzie więc ono poprawne również dla algorytmu, który korzysta z rozkładu wykładnika na postać binarną.

Najpierw należy zwrócić uwagę uczniów, że w dwóch ostatnich algorytmach liczba mnożeń zależy od liczby bitów w przedstawieniu wykładnika n w postaci binarnej. Z poprzedniej lekcji wiadomo, że liczba bitów w binarnej reprezentacji liczby n wynosi około $\log_2 n$. W przedstawionych algorytmach, podnosimy do kwadratu tyle razy, ile jest pozycji w reprezentacji binarnej i dodatkowo mnożymy przez x , gdy w rozwinięciu pojawia się bit 1. Bit równy 1 może pojawić się na każdej pozycji, a zatem w sumie jest wykonywanych nie więcej niż $2 \log_2 n$ mnożeń. Dla naszej potęgi mamy więc:

$$2 \log_2 n = 2 \log_2 12345678912345678912345678912345 = 2 \cdot 103,28 = 206,56,$$

a zatem obliczenie przykładowej wartości potęgi, której wykładnik ma 32 cyfry wymaga wykonania nie więcej niż 206 mnożeń, co nawet na zwykłym komputerze osobistym potrwa niezauważalny ułamek sekundy.

Z Tabeli 1 wynika, że obliczenie wartości potęgi dla wykładnika o stu cyfrach wymaga wykonania nie więcej niż 670 mnożeń, a dla wykładnika o 500 cyfrach – nie więcej niż 3000 mnożeń, co także potrwa na komputerze osobistym ułamek sekundy.

Problem i algorytmy potęgowania są najlepszą ilustracją słów Ralfa Gomory'ego, naukowego szefa firmy IBM:

Najlepszym sposobem przyspieszania pracy komputerów
jest obarczanie ich mniejszą liczbą działań.

czyli prawdziwe przyspieszanie obliczeń osiągamy dzięki efektywnym algorytmom, a nie szybszym komputerom.



Lekcja 5. Algorytm Euklidesa, a metoda połowienia. Czas: 60 min.

Algorytm Euklidesa jest najstarszym znanym algorytmem, został podany ok. 300 lat p.n.e. Służy on do znajdowania największego wspólnego dzielnika (w skrócie NWD) dwóch liczb. Ten algorytm można uzasadnić w następujący sposób. Przypuśćmy, że chcemy obliczyć, ile wynosi NWD(n, m) dla dwóch liczb naturalnych n i m , gdzie możemy przyjąć, że $n > m$. Skorzystajmy z oczywistej równości. Jeśli n dzielimy w sposób całkowity przez m , to otrzymujemy całkowity **iloraz** q oraz **resztę** r , która jest mniejsza od dzielnej, czyli od m . A zatem zachodzi równość:

$$n = qm + r, \quad \text{gdzie } 0 \leq r < m.$$

Z tej równości wynika, że jeśli jakaś liczba dzieli n i m , to dzieli również m i r . Także na odwrót, liczba, która dzieli m i r dzieli także n i m . Stąd otrzymujemy zależność rekurencyjną:

$$\text{NWD}(n, m) = \text{NWD}(m, r).$$

Zatem, NWD dzielnej i dzielnika jest równy NWD dzielnika i reszty. Tę zależność można kontynuować, aż dojdziemy do reszty r równej 0, a wtedy $\text{NWD}(m, 0) = m$, bo 0 jest podzielne przez każdą liczbę.

Prosimy uczniów o wykonanie obliczeń dla np. $\text{NWD}(70, 25)$. Powinni otrzymać:

$$\text{NWD}(70, 25) = \text{NWD}(25, 20) = \text{NWD}(20, 5) = \text{NWD}(5, 0) = 5,$$

gdyż obliczenia mają postać:

$$\begin{array}{l} 70 = 2 \cdot 25 + 20 \\ 25 = 1 \cdot 20 + 5 \\ 20 = 4 \cdot 5 \end{array}$$

W tym przykładzie pojawia się ciąg liczb: $n = 70, m = 25, r = 20, 5$. Prosimy uczniów o wykonanie obliczeń np. dla liczb: 34 i 21. Wtedy powinien pojawić się ciąg liczb: 34, 21, 13, 8, 5, 3, 2, 1, czyli $\text{NWD}(34, 21) = 1$, a więc te liczby są względnie pierwsze. Na tych ciągach trudno jest zauważyć pewną regularność, wyjaśnimy ją więc pomagając sobie rysunkami. Chcemy pokazać, że

reszta r z dzielenia n przez m nie jest większa od połowy liczby n .

Rozważmy dwa przypadki, gdy m jest mniejsze od połowy n i większe od połowy n .

Przypadek I. $m \leq n/2$.

n : _____

m : _____

Jeśli teraz n dzielimy przez m , to może pozostać reszta r , ale nie jest ona większa niż m , a ponieważ założyliśmy, że $m \leq n/2$, więc mamy również $r \leq n/2$.

Przypadek II. $m > n/2$.

n : _____

m : _____

Jeśli teraz n dzielimy przez m , to w tym przypadku resztą r jest równa $n - m$, a ponieważ $m > n/2$, więc pozostanie mniej niż $n/2$, czyli mamy również $r \leq n/2$.



Z tego rozumowania wynika, że każda liczba w ciągu generowanym w algorytmie Euklidesa jest przynajmniej dwa razy mniejsza, niż liczba w tym ciągu stojąca o dwie pozycje wcześniej. Przypomina to naszą intuicyjną definicję logarytmu z lekcji 2 z tą różnicą, że tam każda liczba była połową poprzedniej, a tutaj jest nie większa niż połowa drugiej poprzedniej. To jednak ma tylko taki wpływ, że musimy wykonać co najwyżej dwa razy więcej kroków, by osiągnąć koniec ciągu. Stąd mamy wniosek:

W algorytmie Euklidesa dla obliczenia $NWD(n, m)$, gdzie $n > m$, jest wykonywanych nie więcej niż $2 \cdot \log_2 n$ iteracji (czyli dzielen).

Nasuwa się tutaj dygresja, że Euklides był bardzo blisko wynalezienia logarytmu, co zrobiono dopiero prawie dwa tysiąclecia po nim!

Ciekawym wyzwaniem dla uczniów może być znalezienie liczb, dla których algorytm Euklidesa wykonuje możliwie najwięcej operacji (iteracji) i sprawdzenie, jak ta liczba iteracji ma się do jej teoretycznego oszacowania.

Lekcja 6. Rzut oka na inne algorytmy. Czas: 120 min.

Przy każdym innym algorytmie, w którym jest stosowana metoda **dziel i zwyciężaj**, można ilustrować wspólną cechę tych metod polegającą na tym, że na każdym kroku algorytmu ciąg danych jest dzielony na dwie, prawie równe części. To z kolei prowadzi do wniosku, że takie algorytmy wykonują liczbę operacji określaną za pomocą logarytmu przy podstawie 2.

Przykładami takich algorytmów są:

1. przeszukiwanie ciągu uporządkowanego przez połowienie (patrz lekcja 2);
2. sortowanie przez binarne umieszczanie;
3. sortowanie przez scalanie.

Te zagadnienia są szczegółowo omówione w książce *Algorytmy*.



GDZIE MOŻNA SPOTKAĆ DRZEWA ROZPINAJĄCE

Błażej Osiński

Uniwersytet Warszawski

blazej.osinski@gmail.com

1. Zagadnienia programowe

Ten scenariusz zajęć jest poświęcony algorytmom grafowym, które służą do znajdowania drzewa rozpinającego w grafie z obciążonymi krawędziami. Te zagadnienia wykraczają poza podstawę programową informatyki, również w zakresie rozszerzonym.

2. Temat zajęć:

Gdzie można spotkać drzewa rozpinające

Zajęcia mają na celu zaprezentowanie uczniom kilku ciekawych problemów algorytmicznych, których wspólnym mianownikiem jest problem znajdowania minimalnego drzewa rozpinającego w grafie.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- zaimplementować algorytmy Kruskala i Prima;
- zastosować standardowe algorytmy wyznaczania drzew rozpinających w sytuacjach nietypowych;
- rozróżnić rozwiązanie aproksymacyjne od heurystycznego.

Nie mniej ważnym celem zajęć jest uświadomienie uczniom, że do rozwiązywania złożonych problemów nie wystarczy znajomość sposobu działania standardowych algorytmów, ale niezbędne jest dogłębne zrozumienie ich istoty.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- posiadać podstawową wiedzę na temat grafów, znać takie pojęcia, jak: graf nieskierowany, spójność grafów, drzewo, przeszukiwanie grafów, most, cykl, cykl Eulera;
- umieć rozwiązywać podstawowe problemy grafowe, wraz z implementacją w wybranym języku programowania;
- umieć konstruować algorytmy zachłanne, a także uzasadniać ich poprawność;
- znać podstawowe struktury danych, w szczególności kopce i strukturę zbiorów rozłącznych.



5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (algorytmu),
 - implementacja (realizacja) rozwiązania w postaci programu komputerowego,
 - testowanie i ewaluacja rozwiązania komputerowego,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- grupowa praca, burza mózgów sterowana podpowiedziami nauczyciela, w celu znalezienia rozwiązania problemów;
- implementacja omawianych rozwiązań w wybranym języku programowania;
- zastosowanie poznanych metod do rzeczywistych danych i porównanie wyników z realną sytuacją.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania algorytmu dla przedstawionego problemu;
- implementacja w wybranym języku programowania odbywa się samodzielnie, lub w parach (dobrych tak, by uczniowie byli na podobnym poziomie swoich informatycznych umiejętności);
- testowanie programów może odbywać się w grupach uczniów, jednak na końcu programy będą poddane automatycznej ewaluacji.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z tablicy lub z tablicy z kartkami papieru, gdzie zapisują specyfikacje problemów, opisy algorytmów, schematy działania algorytmów, fragmenty programów. Przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami. Do prezentacji wyników pracy (na przykład lekcji 2.) przyda się rzutnik multimedialny..

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- materiały dotyczące algorytmiki, opracowane w projekcie Informatyka +;
- dane, a także narzędzia do wizualizacji dostępne w sieci Internet;
- komputer i jego podstawowe oprogramowanie, w tym kompilatory algorytmicznych języków programowania.



9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest omówienie ciekawych problemów związanych z wyszukiwaniem drzew rozpinających, mogą być rozłożone na kilka lekcji, niekoniecznie kolejnych. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku prezentowane, lub przypomniane są algorytmy Kruskala i Prima (Lekcja 1);
- następnie uczniowie zastosują poznane algorytmy do rzeczywistych danych – porównają planowaną w Polsce sieć autostrad z minimalnym drzewem rozpinającym (Lekcja 2);
- w dalszej części uczniowie są konfrontowani z problemami o rosnącym stopniu trudności, których rozwiązanie wymaga zmodyfikowania poznanych algorytmów (Lekcja 3);
- na koniec uczniowie zajmują się problemem aproksymacji minimalnego drzewa łączącego podzbiór wierzchołków. Szczególnie ważne jest uświadomienie różnicy między rozwiązaniem aproksymacyjnym a heurystycznym (Lekcja 4).

Lekcja 1. Porównanie algorytmów znajdowania minimalnego drzewa rozpinającego. Czas: 60 min.

Lekcja rozpoczyna się od postawienia przez nauczyciela problemu znalezienia minimalnego drzewa rozpinającego, zilustrowanego na przykładzie budowy najtańszej sieci autostrad (do tego przykładu wracamy na drugiej lekcji). Można też wspomnieć, że pierwsze poprawne rozwiązanie tego problemu zostało opublikowane w 1926 przez Czecha Otakara Borůvka, który starał się znaleźć optymalną sieć elektryczną dla swojego kraju. Po wprowadzaniu bardziej formalnego opisu problemu pracując z grupą należy opisać poprawne rozwiązania: algorytmy Kruskala i Prima. Dokładny ich opis można znaleźć m.in. w książce *Wprowadzanie do algorytmów*.

Z dużym prawdopodobieństwem część uczniów zna rozwiązanie tego problemu i chętnie będą oni o nim rozmawiać. Na początku zajmiemy się tym algorytmem, który pierwszy pojawi się w pracy z uczniami. Należy omówić go dokładnie, w szczególności zwrócić uwagę na kwestię pełnego uzasadnienia poprawności zastosowanej metody zachłannej (nie jest to zadanie trudne, a jednak często uczniowie mają z nim problemy).

Przy omawianiu algorytmu Prima należy porównać go z algorytmem Dijkstry: na przykład niech uczniowie skonstruują graf, dla którego te algorytmy wybierają **różne** krawędzie. Warto też omówić sposoby prostej implementacji kopca w języku C++ z wykorzystaniem biblioteki STL (za pomocą kontenerów *set* lub *priority_queue*).

Przy algorytmie Kruskala należy natomiast poświęcić trochę czasu strukturze zbiorów rozłącznych: uczniowie powinni uzasadnić, że zamortyzowany koszt pojedynczej operacji wynosi $O(\log n)$. Jako ciekawostkę można dodać, że koszt ten to tylko $O(\log^* n)$ i podać przykłady, jak wolno rośnie funkcja \log^* .

Po omówieniu pierwszego z poprawnych algorytmów uczniowie przystępują do jego implementacji (pojedynczo lub w parach), a wynikowy program poddają automatycznej ocenie na zestawie testów przygotowanym przez nauczyciela. Jeżeli uczestnicy będą się męczyli z tym bardzo długo, należy dokończenie implementacji pozostawić jako pracę domową, by został czas na drugą część zajęć.

Teraz należy zaprezentować uczniom drugi z poprawnych algorytmów, który nie pojawił się w dyskusji z uczniami, a następnie porównać go z poprzednim. Zestawienie można przeprowadzać

w postaci pytań, lub krótkich zadań dla uczestników. Przykładowe problemy, na które uczniowie powinni udzielić odpowiedzi, to:

- Dokładne obliczenie złożoności czasowej w zależności od liczb n i m , liczby wierzchołków i liczby krawędzi w grafie (oznaczenia n i m stosujemy również w dalszej części).
- Który algorytm będzie prostszy w implementacji? Tu nie ma jednoznacznej odpowiedzi, ale zastanowienie się nad tym pytaniem przyczyni się do lepszego zrozumienia obu algorytmów.
- Jak zmodyfikować oba algorytmy, by znaleźć minimalny las rozpinający w grafie niespójnym?
- Czy oba algorytmy można zmodyfikować tak, by znalazły maksymalne drzewo rozpinające?

Te pytania nawiązują do jednego z celów lekcji, którym jest przekonanie uczniów, by nie patrzyli na algorytmy jak na „czarne skrzynki”, ale raczej jak na metody, które należy rozumieć i dostosowywać w zależności od potrzeb.

Jako zadanie domowe należy zadać implementację drugiego algorytmu znajdowania minimalnego drzewa rozpinającego. Ważne jest, by uczniowie samodzielnie zakodowali oba algorytmy, gdyż będą z nich korzystać podczas kolejnych lekcji. W szczególności będą modyfikować i dostosowywać swoje programy, dlatego bardzo istotne jest, by były to programy przez nich napisane i które bardzo dobrze znają i rozumieją.

Lekcja 2. Sieć autostrad jako minimalne drzewo rozpinające. Czas: 45 min.

Celem tej lekcji jest przekonanie uczniów, że algorytmy są często sposobami rozwiązania rzeczywistych problemów, a nie jedynie teoretyczną rozrywką. Jako przykładem posłużymy się wspomnianym już problemem budowy sieci autostrad. Jest to na tyle ważny i obecny w mediach temat, że powinien zainteresować wielu uczniów.

Państwo planując budowę dróg staje przed pewnym dylematem. Z jednej strony chciałoby, żeby sieć autostrad umożliwiała przedostanie się między dowolnymi dwoma ważnymi punktami (np. dużymi miastami, przejściami granicznymi). Z drugiej zaś strony – budowanie dróg jest drogie, więc nie można sobie pozwolić na marnowanie pieniędzy na zbędne połączenia. Wydaje się, że rozsądnym rozwiązaniem może być takie zaplanowanie autostrad, by stanowiły drzewo rozpinające grafu, którego wierzchołkami będą ważne punkty w państwie.

Celem uczniów jest sprawdzenie tej hipotezy na przykładzie sieci autostrad w Polsce. Pracować będą w parach lub w małych grupach. Kolejne zadania do wykonania przez uczestników to:

1. Odnalezienie odległości (dla uproszczenia w linii prostej) między punktami połączonymi przez aktualny system autostrad (między dużymi miastami i przejściami granicznymi): Gdańsk, Łódź, Warszawa, Kraków, Wrocław, Świecko, Olszyna, Jędrzychowice, Gorzyczki, Korczowa, Kukuryki.
Potrzebne informacje można odnaleźć na stronie: <http://www.odleglosci.pl/odleglosci-kody.php> Nauczyciel może też dla uproszczenia przygotować dla uczniów tabelkę z tymi odległościami.
2. Przekształcenie danych do formatu, którego używa program z poprzednich zajęć (lub dostosowanie programu do formatu danych).



3. Obliczenie minimalnego drzewa rozpinającego (nie wystarczy jego wielkość, potrzebne będą też użyte krawędzie).
4. Wizualne przedstawienie otrzymanego rozwiązania. Uczniowie mogą nanieść wierzchołki i krawędzie grafu na mapę użytkownika (moje mapy) na stronie internetowej: <http://maps.google.com/>.
5. Porównanie zarówno kształtu, jak i wielkości swojego rozwiązania z rzeczywistym systemem autostrad w Polsce. Informacje na temat istniejących i planowanych dróg są dostępne pod adresem: http://pl.wikipedia.org/wiki/Autostrady_w_Polsce.

Należy zwrócić uwagę na to, że sieć planowanych autostrad jest rzeczywiście drzewem, nie ma cykli. Oznacza to, że założenie o braku „nadmiarowych” połączeń nie jest w przypadku Polski bezpodstawne.

Lekcja 3. Występowanie krawędzi w drzewach rozpinających.

Czas: 90-120 min.

Na tych zajęciach zajmiemy się kilkoma problemami, których wspólnym mianownikiem jest badanie występowania wybranych krawędzi w drzewie rozpinającym. Rozwiązania dwóch pierwszych z nich są na tyle proste, że uczniowie powinni je wymyślić w czasie wspólnej pracy. Wszystkie rozwiązania kładą nacisk na zrozumienie istoty działania poznanych algorytmów zachłannych, a nie tylko umiejętność ich zakodowania.

Problem na rozgrzewkę

Wybermy w grafie jedną ustaloną krawędź. Czy należy ona do jakiegoś minimalnego drzewa rozpinającego?

Rozwiązanie jest bardzo proste – na początku liczymy wielkość minimalnego drzewa rozpinającego, a następnie sprawdzamy, czy jakiekolwiek drzewo rozpinające zawierające ustaloną krawędź ma tę samą wielkość co minimalne (robimy to za pomocą algorytmu Kruskala, w którym jako pierwszą przetwarzamy ustaloną krawędź, a następnie wszystkie pozostałe w kolejności niemalejących wag). Złożoność rozwiązania jest taka, jak dwukrotnego użycia algorytmu Kruskala, czyli $O(m \log m + m \log^* n)$. Po wspólnym wymyśleniu rozwiązania uczestnicy powinni zmodyfikować napisany wcześniej algorytm Kruskala, by rozwiązywał ten problem.

Bajtocja, I OIG

Drugim problemem jest udzielenie odpowiedzi na powyższe pytanie dla **wszystkich** krawędzi w grafie. Zadanie to pojawiło się na I Olimpiadzie Informatycznej Gimnazjalistów. Treść zadania Bajtocja znajduje się pod adresem: <http://main.edu.pl/user.phtml?op=showtask&task=baj&con=OIG1>. Dostępna jest też tam możliwość automatycznej weryfikacji rozwiązania tego zadania na zestawie testów użytych podczas zawodów.

Oczywistym rozwiązaniem, które będzie pewnie pierwszą propozycją uczniów, jest m -krotne zastosowanie poprzedniego algorytmu. Niestety złożoność takiego rozwiązania to $O(m \log m + m^2 \log^* n)$. Należy zachęcić uczniów do znalezienia szybszego rozwiązania. Jest nim inna modyfikacja algorytmu Kruskala. Zauważmy, że jedyną możliwością, by dostać różne drzewa rozpinające w wyniku działania algorytmu Kruskala, jest istnienie kilku krawędzi o tej samej wadze, z których tylko część zostanie wybrana do drzewa. Jest to jedyna możliwość gdyż, jeżeli pominiemy jakąś krawędź,

a weźmiemy inną o większej wadze, to nie dostaniemy minimalnego drzewa, zgodnie z dowodem poprawności algorytmu Kruskala. Aby zatem stwierdzić, czy dana krawędź **może** pojawić się w jakimś minimalnym drzewie rozpinającym wystarczy sprawdzić, czy byłaby wybrana przez algorytm Kruskala, jeżeli byłaby przetwarzana jako pierwsza spośród krawędzi o tej samej wadze. Zauważmy też, że niezależnie od tego jaki podzbiór krawędzi o tej samej wadze weźmiemy do drzewa rozpinającego, otrzymamy ten sam podział na zbiory rozłączne. Jest tak ponieważ wszystkie krawędzie o wadze nie większej niż ustalona wartość w wyznaczają podział na zbiory rozłączne, taki sam jak minimalny las rozpinający z nich wybrany.

Złożoność tego rozwiązania jest asymptotycznie taka sama jak algorytmu Kruskala. Po wspólnej analizie uczniowie powinni przystąpić do implementacji. Ważne jest, by każdy z programów przeszedł przez zestaw testów z Olimpiady (jeżeli nie uda się tego osiągnąć na zajęciach, to pozostawiamy jako pracę domową).

Wszystkie drzewa rozpinające

Omawiany problem można jeszcze trochę utrudnić: dla każdej krawędzi stwierdź, czy występuje we **wszystkich** minimalnych drzewach rozpinających. Ten problem warto omawiać tylko pod warunkiem, że uczniowie mają większą wiedzę na temat grafów, w szczególności znają pojęcie **mostu** i potrafią zaimplementować algorytm służący do znajdowania mostów w grafie.

Rozwiązanie jest podobne do poprzedniego. Należy rozważać kolejne grupy krawędzi o tych samych wagach. Budujemy z nich graf, którego wierzchołkami są otrzymane do tej pory zbiory rozłączne. Nie trudno zauważyć, że jeżeli dana krawędź ma być obecna w dowolnym minimalnym drzewie rozpinającym, to nie może być zastąpiona (w sensie łączenia wierzchołków) przez inny podzbiór krawędzi o tej samej wadze. Oznacza to, że będzie ona mostem w zbudowanym przed chwilą grafie. Mosty możemy oczywiście wyznaczyć za pomocą standardowego algorytmu.

Na chwilę uwagi zasługuje pytanie o czas działania otrzymanego algorytmu. Na pierwszy rzut oka wydawać by się mogło, że trzeba pesymistycznie m razy wykonywać algorytm wyszukiwania mostów o złożoności $O(n + m)$, czyli sumaryczna złożoność wyniosłaby $O(mn + m^2)$. Zauważmy jednak, że kolejne wyszukiwania mostów odbywają się w grafach indukowanych przez podzbiory krawędzi, które to podzbiory stanowią podział wszystkich krawędzi grafu. Oznacza to, że w rzeczywistości sumaryczny czas wyszukania wszystkich mostów wyniesie tyle, co suma wielkości tych indukowanych grafów, czyli $O(m)$. Ponieważ musimy jeszcze posortować krawędzie otrzymamy całkowitą złożoność $O(m \log m)$.

Przejmowanie linii lotniczych

Kolejny ciekawy problem wymagający modyfikacji algorytmu Kruskala nazywa się *Aviamachinations* i można go znaleźć na stronie: <http://acm.sgu.ru/problem.php?contest=0&problem=323> (tam też można automatycznie przetestować rozwiązanie).

Mamy daną sieć złożoną z k połączeń lotniczych podzieloną między m linii lotniczych. Chcemy, by jedna linia lotnicza przejęła od pozostałych połączenia tak, by mogła transportować podróżnych między dowolnymi lotniskami (być może z przesiadkami). Przejęcie każdego połączenia kosztuje pewną kwotę. Którą linię lotniczą należy wybrać, by koszty przejścia były jak najmniejsze?

Pierwszym rozwiązaniem, które przychodzi do głowy, jest wywołanie algorytmu Kruskala osobno dla każdej linii lotniczej, na początku wrzucając za darmo połączenia obsługiwane przez daną linię. Jest to oczywiście rozwiązanie poprawne, jednak zbyt wolne. Istnieje jednak algorytm szybszy: najpierw znajdujemy minimalne drzewo rozpinające T dla całej sieci, nie biorąc pod uwagę podziału na



linie lotnicze. Następnie dla każdej linii lotniczej najpierw wstawiamy za darmo własne połączenia, a następnie dobieramy pozostałe za pomocą algorytmu Kruskala, ale tylko z puli krawędzi wykorzystanych w T . Suma wag tak wybranych krawędzi stanowi najmniejszy sumaryczny koszt przejścia dla danej linii lotniczej.

Dowód tego faktu nie jest zupełnie prosty, ale opiera się na standardowej metodzie dowodzenia dla algorytmów zachłanych. Dla linii lotniczej p weźmy nasze rozwiązanie (zbiór krawędzi) Z oraz rozwiązanie optymalne O . Załóżmy nie wprost, że wielkość (suma wag krawędzi) O jest mniejsza niż W . W obu zbiorach znajdują się (a jeżeli nie, to je dodamy bez kosztu) darmowe krawędzie, czyli połączenia należące do linii p . Część wspólną zbiorów Z i O oznaczmy przez W (w szczególności w zbiorze W są zawarte wszystkie darmowe krawędzie), natomiast $Z - W$ przez Z_0 i $O - W$ przez O_0 . Z naszego założenia wynika, że sumaryczna waga krawędzi w O_0 jest mniejsza niż w Z_0 . Wówczas jednak waga zbioru $T - Z_0 + O_0$ byłaby mniejsza niż waga minimalnego drzewa rozpinającego. Uzyskana sprzeczność dowodzi poprawności naszego algorytmu.

Powyższy dowód nie jest chyba zbyt skomplikowany, zaciemnia go jedynie trochę duża liczba zbiorów, którymi operujemy.

Lekcja 4. Problemy aproksymacyjne. Czas: 90 min.

W ramach tej lekcji uczniowie prawdopodobnie po raz pierwszy spotkają się z algorytmami aproksymacyjnymi, czyli rozwiązującymi problemy w sposób przybliżony. Choć ten temat nie pojawia się zwykle na zawodach informatycznych, to warto skrótkowo przedstawić go uczniom, ze względu na jego ważne miejsce w aktualnym rozwoju algorytmiki.

Pokrycie wierzchołkowe

Jako wprowadzenie do tematu rozważmy znany problem pokrycia wierzchołkowego: mając dany graf, wybrać taki zbiór wierzchołków, by każda krawędź była incydentna z co najmniej jednym wybranym wierzchołkiem. Należy poinformować uczniów, że jest to problem NP-zupełny, czyli nie jest znany dla niego algorytm rozwiązania, które działałoby wydajnie dla dużych danych.

To, co możemy próbować robić, to znaleźć rozwiązanie przybliżone. Odruchem jest często wskazanie jakiegoś algorytmu heurystycznego¹ (w skrócie **heurystyki**). Plusem heurystyki jest to, że często działa dość dobrze, ale nie jesteśmy w stanie stwierdzić, jak dobrze. Aby uświadomić uczniom ten problem rozważmy następującą heurystykę rozwiązującą problem pokrycia wierzchołkowego grafu G :

```
na początku pokrycie jest zbiorem pustym;
while  $G$  nie jest pusty do
     $w$ :=dowolny wierzchołek  $G$ ;
    zwiększ pokrycie o  $w$ ;
    usuń z  $G$  wierzchołek  $w$  oraz wszystkie krawędzie
    incydentne z nim
```

Weźmy graf G będący n -wierzchołkową gwiazdką (jeden wierzchołek w jest połączony ze wszystkimi pozostałymi wierzchołkami i brak jest krawędzi między pozostałymi wierzchołkami). Gdy w pierw-

¹ Na ogół przyjmuje się, że algorytm aproksymacyjny tym różni się od algorytmu heurystycznego, że w przypadku tego pierwszego można określić, jak bardzo generowane rozwiązania są odległe od optymalnego, zaś w przypadku algorytmu heurystycznego na ogół nie można podać takiego oszacowania generowanych rozwiązań.

szej iteracji pętli zostanie wybrany wierzchołek w , to otrzymamy rozwiązanie optymalne. Jeżeli jednak będziemy mieli pecha wybierając wierzchołki leżące na końcach ramion gwiazdy, to możemy skonstruować pokrycie bardzo dalekie od optymalnego, nawet o wielkości $n - 1$.

Algorytmy aproksymacyjne mają natomiast tę zaletę, że jesteśmy w stanie wskazać konkretne ograniczenie, jakie będzie spełniało wygenerowane przez nie rozwiązanie. Zwykle żądamy, by rozwiązanie było co najwyżej k razy gorsze od optymalnego, gdzie k jest określoną stałą, lub funkcją od wielkości danych wejściowych. Okazuje się, że dla problemu pokrycia wierzchołkowego istnieje bardzo prosty algorytm aproksymacyjny, który jest 2-aproksymacją, czyli zwraca pokrycie co najwyżej dwa razy większe od optymalnego.

Oto skrótowy opis algorytmu:

```
na początku pokrycie jest zbiorem pustym;
while G nie jest pusty do
    ( $w_1, w_2$ ):=dowolna krawędź w G;
    zwiększ pokrycie o  $w_1$  i  $w_2$ ;
    usuń z G wierzchołki  $w_1$  i  $w_2$  i wszystkie krawędzie
    incydentne z nimi
```

Z pozoru, ten algorytm jest bardzo podobny do poprzedniej heurystyki. Niemniej teraz jesteśmy już w stanie pokazać, że zwracane pokrycie będzie co najwyżej dwa razy większe od optymalnego. Jest to bardzo proste – dla każdej krawędzi wybieranej przez algorytm co najmniej jeden jej wierzchołek musi należeć do pokrycia (wynika to wprost z definicji pokrycia wierzchołkowego). My natomiast zamiast jednego bierzemy oba, zatem za każdym razem bierzemy co najwyżej dwa razy za dużo.

Aby uczniowie lepiej zrozumieli działanie tego algorytmu, powinni wykonać go na kilku małych grafach, a także ręcznie znaleźć wynik optymalny i porównać go z przybliżonym. W szczególności, warto sprawdzić, jak działa nasz algorytm dla uprzednio rozważanej „gwiazdy”.

Koleje, XIV OI

Uprzednie rozważania stanowiły jedynie wprowadzenie do tematyki algorytmów aproksymacyjnych. Teraz zajmiemy się trochę trudniejszym problemem, który pojawił się w finale XIV Olimpiady Informatycznej, jako zadanie próbne. Ponownie, treść zadania oraz możliwość automatycznej oceny rozwiązania są dostępne na stronie: <http://main.edu.pl/user.phtml?op=showtask&task=kol&con=OI14>

W zadaniu dany jest spójny graf G wśród którego wierzchołków wyróżniono zbiór V' mocy p . Zadanie polega na wybraniu takiego podzbioru krawędzi G o minimalnej łącznej sumie wag, aby istniała ścieżka złożona wyłącznie z wybranych krawędzi łącząca dowolne dwa wierzchołki z V' .

Po poinformowaniu uczniów, że jest to także problem NP-zupełny, można zachęcić ich do prób wspólnego rozwiązania tego problemu. Na pewno pojawi się wiele różnych propozycji heurystyk, trzeba wtedy stanowczo zażądać dowodu 2-aproksymacji. Po pewnym czasie należy zaprezentować poprawne rozwiązanie, lub umiejętnie doprowadzić do niego podpowiedziami.

Najpierw znajdujemy minimalne odległości między wierzchołkami z V' (ze względu na dodatkowe ograniczenia w treści zadania, najszybszym rozwiązaniem jest p krotne razy użycie algorytmu Dijkstry). Następnie w otrzymanym grafie pełnym G' , w którym krawędź (a, b) ma wagę równą

najkrótszej ścieżce między a i b w G , znajdujemy minimalne drzewo rozpinające. Aby uzyskać wynik wystarczy z powrotem przekształcić krawędzie tego drzewa rozpinającego w ścieżki złożone z oryginalnych krawędzi grafu G (i usunąć duplikaty).

Algorytm jest bardzo prosty, trochę bardziej złożony jest dowód, dlaczego rzeczywiście jest to algorytm 2-aproksymacyjny. Aby to uzasadnić posłużymy się następującą konstrukcją: weźmy wynik optymalny dla tego problemu. Jest to z pewnością drzewo w wyjściowym grafie G . Teraz zamieńmy je na cykl poprzez podwojenie każdej krawędzi i przejście standardowym algorytmem znajdującym cykl Eulera. Uzyskany cykl ma oczywiście wielkość dwa razy większą od rozwiązania optymalnego. Pokażemy teraz, że jest on większy od wyniku naszego algorytmu.

Gdy za pomocą krawędzi z G' połączymy w listę pierwsze wystąpienia kolejnych wierzchołków z V' na cyklu uzyskamy jakieś drzewo rozpinające grafu G' (lista jest w szczególności drzewem). Zauważmy, że wielkość tego drzewa jest z pewnością mniejsza niż długość cyklu: po pierwsze nie dodajemy krawędzi od ostatniego wierzchołka, po drugie – krawędzie w G' mają długość najkrótszych ścieżek w G , więc w szczególności nie dłuższych od ścieżek idących wzdłuż jakiegoś ustalonego drzewa w G .

Znaleźliśmy zatem jakieś drzewo rozpinające w grafie G' . Wielkość tego drzewa jest nie większa niż wielkość dwukrotnego rozwiązania optymalnego. Tymczasem rozwiązanie zwracane przez nasz algorytm jest najmniejszym drzewem rozpinającym w G' , zatem w szczególności mniejszym niż to przed chwilą znalezione. Kończy to dowód faktu, że nasz algorytm jest 2-aproksymacją postawionego problemu.

Bardziej szczegółowy dowód poprawności, lepsze rozwiązania, a także dyskusja nad metodą implementacji rozwiązań tego zadania znajdują się w książce *XIV Olimpiada Informatyczna 2006/2007*, Warszawa 2007. Z pewnością warto się zapoznać z tym materiałem przed przeprowadzeniem zajęć – dla ułatwienia zastosowałem tutaj takie same oznaczenia, jak w oryginalnym opracowaniu.



JAK WYGRYWAĆ W GRACH

Bolesław Kulbabiński

Uniwersytet Warszawski

bkulbabinski@gmail.com

1. Zagadnienia programowe

Te zajęcia są wprowadzeniem do teorii gier, która nie występuje w podstawie programowej. Rozważania są jednak na elementarnym poziomie matematycznym i informatycznym i ze względu na swój zakres mogą zainteresować uczniów, których pociąga logiczne i algorytmiczne myślenie. Podczas zajęć uczniowie posługują się komputerem jako „partnerem” do gry.

2. Temat zajęć

Jak wygrywać w grach

Celem tych zajęć jest:

- przedstawienie ścisłego, matematycznego podejścia do prostych, dwuosobowych gier kombinatorycznych;
- uświadomienie uczniom, że pewna klasa gier ma zawsze z góry ustalonego zwycięzcę i przegranego;
- przedstawienie metod znajdowania optymalnych ruchów w grach.

3. Cele lekcji (zajęć)

W wyniku tych zajęć uczeń powinien umieć:

- zdefiniować pozycje przegrywające i wygrywające oraz podać warunki na ich istnienie;
- rozwiązać grę NIM, to znaczy stwierdzić, czy pozycja jest przegrywająca czy wygrywająca i podać optymalny ruch;
- znać definicję funkcji Sprague-Grundy'ego i potrafić wyliczyć jej wartości dla gry będącej sumą gier.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- znać i rozumieć zapis binarny liczb całkowitych;
- znać i rozumieć funkcję XOR oraz jej własności, takie jak: łączność, przemienność oraz możliwość skracania;
- znać technikę programowania dynamicznego;
- mieć podstawową wiedzę z teorii grafów, w szczególności wiedzieć, co to jest acykliczny graf skierowany (DAG);



5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- rozgrywanie kilku partii analizowanej gry w parach – ma to doprowadzić do pełnego zrozumienia zasad gry i próby opracowania własnej strategii;
- posłużenie się przez nauczyciela i uczniów objaśnieniami i demonstracjami przykładowych strategii w rozważanych grach za pomocą schematów;
- przygotowanie przez uczniów schematów działania wybranych algorytmów;
- zapisanie algorytmów w wybranym języku programowania i uruchomienie odpowiednich programów na komputerze;
- samodzielne testowanie poprawności programów dla odpowiednio dobranych danych;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania algorytmu dla przedstawionego problemu;
- przygotowywanie opisów algorytmów może odbywać się indywidualnie lub w parach uczniów;
- programy dla komputerów uczniowie piszą samodzielnie;
- testowanie programów może odbywać się w grupach uczniów – uczniowie wspólnie poprawiają błędy w programach, dobierają dane i porównują wyniki działania swoich programów;
- końcowym efektem pracy nad danym problemem jest prezentacja opisu algorytmu, programu komputerowego oraz wyników działania programu.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z tablicy lub z tablicy z kartkami papieru, gdzie zapisują specyfikacje problemów, opisy algorytmów, schematy działania algorytmów, fragmenty programów. Przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami.

Uczniowie nieraz są proszeni o rozgrywanie między sobą partii gier, polegających głównie na układaniu żetonów, przemieszczaniu pionków lub rysowaniu. Użyteczne będą zatem kartki papieru, przybory do pisania oraz zbiór jednakowych żetonów, klocków czy też monet (na przykład jednogroszowych).

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- materiały dotyczące algorytmiki, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, w tym kompilatory algorytmicznych języków programowania.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest omówienie podstaw teorii gier, mogą być rozłożone na kilka lekcji, niekoniecznie kolejnych. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel wyjaśnia uczniom, jakie dokładnie gry będą analizowali, oraz przedstawia kilka prostych gier, prosząc uczniów o znalezienie jak najlepszych strategii (Lekcja 1);
- następnie są definiowane pozycje wygrywające i przegrywające oraz zostaje omówiona klasyczna gra NIM (Lekcja 2);
- jako kolejny etap formalizacji zostaje wprowadzona funkcja Sprague-Grundy'ego oraz jej podstawowa własność umożliwiające sumowanie gier (Lekcja 3);
- na koniec, nauczyciel przedstawia uczniom kilka gier, które pozornie są zupełnie inne od wcześniej omawianych, jednakże można sprowadzić je do znanych schematów (Lekcja 4).

Lekcja 1. Przedstawienie kilku prostych gier. Czas: 45 minut.

Zanim zajmiemy się ścisłą analizą gier, musimy zdefiniować jakimi dokładnie grami będziemy się zajmować. Chcemy, aby analizowane gry spełniały pięć poniższych warunków.

- Po pierwsze, skupimy się tylko na grach dwuosobowych, w których gracze wykonują ruchy na przemian.
- Nie będziemy rozważać gier nieskończonych. Oznacza to, że w analizowanych przez nas grach zawsze będzie można wyłonić zwycięzcę po pewnej, skończonej liczbie ruchów, niezależnie od sposobu gry obu graczy.
- Trzecią, ważną własnością jest tzw. pełna informacja, co oznacza, że obydwaj gracze zawsze widzą cały stan gry (planszy, pionków, kart itp.). Przykładowo, nie rozważamy gier karcianych, takich jak brydż, gdzie gracze mają w rękach własne karty, których inni gracze nie znają.
- Kolejną zasadą niech będzie to, że zbiór dopuszczalnych ruchów nie zależy od gracza, który aktualnie wykonuje ruch. Przykładowo, nie będziemy rozważać gry w szachy, gdzie jeden gracz może ruszać się tylko białymi pionkami a drugi tylko czarnymi. W naszych grach dopuszczalne ruchy nie zależą od tego, który gracz je wykonuje.
- Wreszcie ostatnie założenie – nie dopuszczamy remisów, tzn. każda gra zawsze kończy się wygraną jednego z graczy.

Warto zapisać w widocznym miejscu te pięć zasad (gry **dwuosobowe, skończone, z pełną informacją, wspólnym zbiorem ruchów i bez remisów**), aby w przyszłości można było skontrolować, że rozważane gry rzeczywiście są tego typu.

Zacznijmy od prezentacji prostej gry w zabieranie. Na stole leży 13 monet. Gracze na przemian zabierają monety ze stołu, przy czym w jednym ruchu można zabrać 1, 2 lub 3 monety (nie można nie wziąć żadnej). Gracz, który zabierze ostatnią monetę wygrywa. Pozwólmy uczniom pograć między sobą w tę prostą grę, jednocześnie prosząc, aby spróbowali opracować zwycięską strategię. Który z graczy ma w tej grze przewagę – rozpoczynający czy może ten drugi? Następnie, niech uczniowie spróbują zagrać w tę samą grę, jednak zaczynając z 12 monetami. Jak zmieni się sytuacja obu graczy?

Prawdopodobnie niektórzy uczniowie odgadną prawidłową strategię w powyższej grze, którą jest tzw. „uzupełnianie do czterech”. **Pozycją** w tej grze nazywać będziemy liczbę monet aktualnie znaj-



dujących się na stole. Gracz, który wykonuje swój ruch z pozycji będącej liczbą podzielną przez 4, zawsze przejdzie do pozycji niepodzielnej przez 4. Z kolei gracz, który zostanie pozycję niepodzielnej przez 4, będzie w stanie odpowiednim ruchem dojść do pozycji podzielnej przez 4. Pozostaje jeszcze zauważyć, że gracz przegrywa jeśli musi wykonać ruch, gdy na stole znajduje się zero monet, a zero jest liczbą podzielną przez 4. A zatem gracz, który zaczyna z pozycji niepodzielnej przez 4, zawsze jest w stanie wygrać! Wystarczy, że przejdzie do pozycji podzielnej przez 4, czym zmusi swojego przeciwnika do przejścia do pozycji niepodzielnej przez 4, z której to będzie mógł kontynuować swoją strategię. W końcu gracz pierwszy doprowadzi do pozycji z liczbą monet równą zero i tym samym wygra grę.

Jeśli więc rozpoczynamy grę z 13 monetami, pierwszy gracz powinien zacząć od zabrania jednej monety ze stołu a następnie uzupełniać ruchy przeciwnika do pełnych czwórek. Jeśli przeciwnik zabierze trzy monety, my zabieramy jedną; jeśli dwie – my też dwie; jeśli natomiast zabierze jedną, my zabieramy trzy. W każdej z tych sytuacji doprowadzamy do pozycji podzielnej przez 4, więc uda nam się w końcu doprowadzić do zera. W przypadku rozpoczęcia gry od 12 monet gracz pierwszy jest niestety skazany na niepowodzenie. Każdy nasz ruch doprowadzi do pozycji niepodzielnej przez 4, z której to nasz przeciwnik może wystartować ze znaną już nam zwycięską strategią.

Dobrym ćwiczeniem programistycznym jest napisanie programu grającego w grę w zabieranie z użytkownikiem. Program powinien być interaktywny oraz wypisywać czytelne informacje i polecenia, na przykład „Podaj liczbę monet:”, „Zabrałem 3 monety. Zostało 8 monet. Podaj swój ruch:”. Oczywiście najważniejsze jest, aby program grał optymalnie, stosując odkrytą przez nas strategię. Uczniowie mogą testować nawzajem swoje programy próbując wygrać, gdy zaczynają z potencjalnie przegranej pozycji.

Rozważmy jeszcze jedną prostą grę. Tym razem na stole znajdują się dwa stosiki monet (nie koniecznie tej samej wysokości). Gracz wykonujący ruch wybiera **jeden** ze stosików, po czym zdejmuje z niego dowolną liczbę monet. Może zabrać nawet wszystkie, ale musi wziąć przynajmniej jedną. Wygrywa gracz, który zabierze ze stołu ostatnią monetę. Ponownie pozwólmy uczniom rozegrać między sobą kilka partii tej gry. Pozycją w tej grze nazywać będziemy aktualne wysokości stosów, czyli **parę liczb** całkowitych nieujemnych. Tym razem dobrą strategią na wygraną będzie „kopiowanie ruchów przeciwnika”. Załóżmy, że gra zaczyna się od dwóch **równych** stosów i gracz nr 1 wykonał pewien ruch. Gracz nr 2 może w tym momencie wykonać taki sam ruch na drugim stosie, doprowadzając z powrotem do układu, w którym stosy są równej wielkości. Kontynuując tę strategię, gracz drugi zapewni sobie wygraną! W pewnym momencie gracz pierwszy będzie zmuszony opróżnić jeden ze stosów, po czym gracz drugi szybko opróżni drugi stos i wygra grę. Widzimy, że pozycja składająca się z dwóch równych stosików stawia na straconej pozycji gracza rozpoczynającego. Jak jest w przypadku stosików różnej wysokości? Gracz rozpoczynający może zdjąć odpowiednią liczbę monet z wyższego stosu tak, aby stosiki się wyrównały. Tym samym zmusi on drugiego gracza do rozpoczęcia z „pechowej” pozycji i w konsekwencji do przegranej. Zwróćmy jeszcze uwagę na podobieństwo do gry w zabieranie. Tam strategią na wygraną było nieustanne doprowadzanie przeciwnika do pozycji podzielnej przez 4. Natomiast w grze ze stosami, klucz do sukcesu leży w doprowadzaniu do układu, w którym oba stosy są równe.

Na koniec możemy zastanowić się nad nieco mniej formalną grą, będącą raczej zagadką logiczną. Zasady gry są następujące: w grze uczestniczy dwóch graczy, którzy na prostokątnym stole układają na przemian jednakowe monety (np. jednogroszowe). Warunek jest jeden – monety nie mogą na siebie zachodzić. Przegrywa gracz, dla którego zabraknie miejsca na stole na położenie monety. Czy



któryś z graczy może łatwo zapewnić sobie wygraną? Pozwólmy uczniom zastanowić się chwilę nad tym problemem. Z pewnością uważni uczniowie dostrzegą tutaj możliwość zastosowania strategii kopiowania ruchów przeciwnika. Wystarczy bowiem, że pierwszy gracz położy pierwszą monetę dokładnie na środku stołu, a następnie będzie kopiował ruchy przeciwnika symetrycznie względem środka stołu. Jeśli przeciwnik znajdzie miejsce na położenie monety, to gracz pierwszy także, tym samym zapewniając sobie zwycięstwo.

Lekcja 2. Pozycje wygrywające i przegrywające na przykładzie gry NIM. Czas: 45 minut.

Drugą lekcję rozpoczniemy od podania ścisłych definicji pozycji przegrywających i wygrywających, którymi będziemy się później wielokrotnie posługiwać.

Pozycją przegrywającą (pozycją P) nazywamy pozycję w grze, z której **każdy** ruch prowadzi do pewnej pozycji wygrywającej **lub** z pozycji tej nie można wykonać żadnego ruchu (jest ona pozycją końcową). **Pozycją wygrywającą** (pozycją W) nazywamy pozycję w grze, z której **istnieje** ruch do pewnej pozycji przegrywającej.

Warto zwrócić uwagę uczniów na zwroty „istnieje ruch” i „każdy ruch”. Definicje te są oczywiście zgodne z intuicją. Jeśli mianowicie z pewnej pozycji istnieje ruch prowadzący do pozycji przegrywającej, to możemy go wykonać aby postawić w niej przeciwnika, co sprawia, że nasza pozycja jest wygrywająca. Analogicznie, jeśli z pewnej pozycji każdy ruch prowadzi do pozycji wygrywającej, to niezależnie od naszego posunięcia postawimy przeciwnika w pozycji wygrywającej, więc sami przegramy.

Spróbujmy oznaczyć pozycje w znanej nam grze w zabieranie. Drobną subtelnością może być fakt, że pozycja 0 (brak monet na stole) jest pozycją przegrywającą. Znalezienie się w takiej pozycji oznacza bowiem, że przeciwnik przed chwilą zabrał ostatnią monetę, czyli wygrał. Tabela pozycji wygląda następująco:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
P	W	W	W	P	W	W	W	P	W	W	W	P	W

Przykładowo, pozycja 2 jest wygrywająca ponieważ **istnieje** ruch prowadzący do pozycji przegrywającej (pozycji 0), natomiast pozycja 8 jest przegrywająca ponieważ **każdy** ruch prowadzi do pozycji wygrywającej (5,6 lub 7).

Przejdźmy teraz do opisu klasycznej gry NIM oraz nieco zaskakującego jej rozwiązania. Na poprzedniej lekcji omawialiśmy grę z dwoma słupkami monet. Był to szczególny przypadek gry NIM. W ogólności słupków może być dowolnie wiele (niech będzie ich k). A zatem pozycja w grze NIM to **ciąg k liczb całkowitych nieujemnych**. Gracz wykonujący ruch wybiera najpierw jeden z k słupków, który nie jest pusty, a następnie zdejmuje z niego dowolną liczbę monet (co najmniej jedną). Wygrywa gracz, który zabierze ostatnią monetę.

Przedstawimy teraz rozwiązanie gry NIM oraz uzasadnimy jego poprawność. Oznaczmy wysokości słupków przez a_1, a_2, \dots, a_k . Twierdzimy, że pozycja jest przegrywająca wtedy i tylko wtedy, gdy

$$a_1 \oplus a_2 \oplus \dots \oplus a_k = 0 \text{ (XOR wartości } a_1, a_2, \dots, a_k \text{ jest równy } 0).$$

Na pierwszy rzut oka może to wydawać się zupełnie nieintuicyjne. Argumentacja będzie jednak podobna do poprzednich strategii – pokażemy, że gdy wartość XOR wysokości słupków jest różna od zera, to istnieje ruch prowadzący do pozycji o wartości XOR równej 0, a także, że z pozycji o wartości XOR równej zero, każdy ruch zmienia tę wartość na niezerową.

Przeanalizujmy najpierw pozycję (a_1, a_2, \dots, a_k) taką, że $a_1 \oplus a_2 \oplus \dots \oplus a_k = 0$. Jeśli wszystkie wysokości słupków są zerami, to jest to oczywiście pozycja przegrywająca. W przeciwnym wypadku rozważmy dowolny ruch polegający na zabraniu pewnej liczby monet z słupka o numerze i . Oznaczmy liczbę monet, jaka została na tym słupku przez b_i . Oczywiście $b_i < a_i$. Załóżmy, że otrzymana pozycja jest również pozycją o wartości XOR równej 0. Zachodzi więc

$$a_1 \oplus \dots \oplus a_i \oplus \dots \oplus a_k = a_1 \oplus \dots \oplus b_i \oplus \dots \oplus a_k.$$

Jak wiemy operacja XOR ma własność skracania, dzięki której możemy pominąć te same wyrazy po obu stronach równania otrzymując $a_i = b_i$, co daje sprzeczność. A zatem rozważany ruch musiał prowadzić do pozycji o wartości XOR różnej od zera. Był on wybrany dowolnie, dzięki czemu możemy śmiało stwierdzić, że każdy ruch prowadzi do takiej pozycji.

Pozostaje rozważyć pozycję (a_1, a_2, \dots, a_k) taką, że $a_1 \oplus a_2 \oplus \dots \oplus a_k > 0$. Oznaczmy tę wartość XOR przez m . Rozważmy teraz zapis binarny liczby m . Oznaczmy przez p pozycję najbardziej znaczącej jedyki w zapisie binarnym liczby m . Z definicji funkcji XOR wynika, że skoro liczba m ma na pozycji p jedynekę, to nieparzyście wiele spośród liczb a_1, a_2, \dots, a_k ma jedynekę na tej właśnie pozycji p . Oznacza to, że istnieje co najmniej jedna taka liczba a_i . Niech $b_i := a_i \oplus m$. Twierdzimy, że szukany przez nas ruchem jest zredukowanie liczby monet na i -tym stosiku z a_i do b_i . Istotnie, korzystając z prawa skracania oraz przemienności otrzymujemy kolejno równości:

$$\begin{aligned} a_1 \oplus \dots \oplus a_{i-1} \oplus a_i \oplus a_{i+1} \oplus \dots \oplus a_k &= m, \\ a_1 \oplus \dots \oplus a_{i-1} \oplus (a_i \oplus m) \oplus a_{i+1} \oplus \dots \oplus a_k &= m \oplus m, \\ a_1 \oplus \dots \oplus a_{i-1} \oplus b_i \oplus a_{i+1} \oplus \dots \oplus a_k &= 0. \end{aligned}$$

Oznacza to, że pozycja, do której przejdziemy jest przegrywająca, przez co nasza jest wygrywająca. Pominęliśmy jednak w dotychczasowym rozumowaniu pewien szczegół. Musimy jeszcze wykazać, że liczba b_i jest mniejsza niż a_i (nie możemy bowiem dokładać monet do stosików!). Wynika to łatwo z tego, że liczba $a_i \oplus m$ ma w zapisie binarnym zero na p -tej pozycji, natomiast cyfry na pozycjach większych niż p ma takie same jak liczba a_i .

Udało nam się zatem podzielić zbiór pozycji w grze NIM na dwa zbiory, których elementy spełniają definicję pozycji wygrywających i przegrywających. Wzorem poprzednich ćwiczeń, uczniowie mogą spróbować napisać program grający optymalnie w grę NIM. Znalezienie optymalnego ruchu nie jest już takie trywialne jak poprzednio i może wymagać trochę sprawności programistycznej (np. znajomości operacji bitowych).

Lekcja 3. Funkcja Sprague-Grundy'ego na przykładzie gry grafowej. Czas: 45 minut.

Przy rozważaniu bardziej złożonych gier pomocne może być twierdzenie Sprague-Grundy'ego, które wprowadzimy na tej lekcji. Zaczniemy jednak od dwóch definicji.

Przez

$$\text{mex}\{a_1, a_2, \dots, a_n\}$$

oznaczać będziemy **najmniejszą liczbę całkowitą dodatnią**, która **nie występuje** w zbiorze $\{a_1, a_2, \dots, a_n\}$. Na przykład $\text{mex}\{3,0,1,5\} = 2$, natomiast $\text{mex}\{1,2,3,5\} = 0$ (nazwa mex pochodzi od angielskiego *minimal-excludant*).

Możemy teraz zdefiniować **funkcję Sprague-Grundy'ego** (w skrócie funkcję Grundy'ego) $SG(p)$, określoną dla każdej pozycji p w grze w następujący sposób:

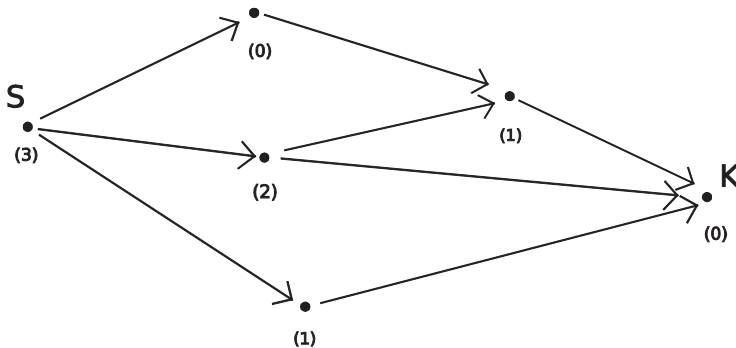
$$SG(p) = 0, \quad \text{gdy } p \text{ jest pozycją końcową (przegrywającą),}$$

$$SG(p) = \text{mex}\{SG(q) : \text{z pozycji } p \text{ można przejść do pozycji } q\}.$$

Innymi słowy, wartość SG dla pozycji p jest najmniejszą liczbą całkowitą dodatnią, która nie jest wartością SG żadnej pozycji osiągalnej bezpośrednio z p .

Łatwo zauważyć, że pozycje przegrywające będą zawsze miały wartość SG równą zero, natomiast dla pozycji wygrywających te wartości będą dodatnie. Pytanie do uczniów – jak znaleźć optymalny ruch z pozycji wygrywającej? (Odpowiedź: trzeba przejść do pozycji, która ma wartość SG równą zero).

Rozważmy teraz prosty przykład gry grafowej. Niech dany będzie graf acykliczny, w którym jeden wierzchołek oznaczono jako startowy. Wyobraźmy sobie, że w wierzchołku startowym znajduje się pionek, który gracz na zmianę przesuwają wzdłuż krawędzi grafu. Przegrywa gracz, który nie może wykonać ruchu, to znaczy dojdzie do wierzchołka, z którego nie wychodzi żadna krawędź. Taki wierzchołek zawsze istnieje, ponieważ graf jest acykliczny, a zatem gra jest skończona. Spróbujmy wyznaczyć wartości funkcji Grundy'ego dla gry na prostym grafie, na przykład takim, jak na poniższym rysunku. S oznacza pozycję startową pionka, K oznacza (jedyną) pozycję końcową. W nawiasach podano wartości funkcji SG.



Zdefiniujemy teraz ważne pojęcie sumy gier.

Sumą gier G_1, \dots, G_n nazywamy układ (G_1, \dots, G_n) , traktowany jako jedna gra, w której ruchy wykonujemy w następujący sposób. Gracz najpierw wybiera grę G_i będącą jedną z gier G_1, \dots, G_n , w której nie została jeszcze osiągnięta pozycja końcowa. Następnie wykonuje ruch w grze G_i zgodnie z jej zasadami. Przegrywa gracz, który nie może wykonać dozwolonego ruchu w żadnej z gier G_1, \dots, G_n . Fizycznie, sumę gier możemy wyobrazić sobie jako ustawienie obok siebie plansz do kilku gier, a następnie wykonywanie na przemian ruchów w dowolnej z nich.

Do analizowania pozycji w grze będącej sumą gier przydatne bywa poniższe twierdzenie.

Twierdzenie Sprague-Grundy'ego

Niech G będzie sumą gier G_1, \dots, G_n . Niech p będzie pozycją w grze G taką, że $p = (p_1, \dots, p_n)$ – p jest złożona z pozycji p_1 w grze G_1, p_2 w grze G_2 itd.). Wtedy wartość funkcji Grundy'ego dla pozycji p dana jest wzorem

$$SG(p) = SG(p_1) \oplus SG(p_2) \oplus \dots \oplus SG(p_n),$$

czyli jest wartością XOR dla wartości SG pozycji w grach składowych.

Twierdzenie to umożliwia łatwe obliczanie wartości funkcji Grundy'ego dla pozornie złożonych gier, będących sumami innych, prostszych gier, pod warunkiem, że umiemy obliczyć SG dla gier składowych. Twierdzenie Sprague-Grundy'ego jest w pewnym sensie uogólnieniem wcześniej udowodnionego twierdzenia o obliczaniu wartości XOR dla wysokości słupków w grze NIM. Jego dowód przebiega analogicznie – pozostawiamy go ambitniejszym uczniom do samodzielnego opracowania.

Jako ilustrację twierdzenia przeanalizujemy wspólnie z uczniami grę będącą sumą dwóch gier grafowych, czyli dla dwóch różnych grafów.

Warto też przedstawić uczniom inne spojrzenie na wcześniej analizowaną grę NIM. Można bowiem traktować ją jako sumę gier złożonych z pojedynczych słupków. Gra NIM z jednym słupkiem jest trywialna, aczkolwiek wyliczenie dla niej wartości funkcji SG pokazuje, że twierdzenie Sprague-Grundy'ego pokrywa się z wcześniej zaprezentowanym rozwiązaniem gry NIM. Okazuje się bowiem, że wartości funkcji SG w tym przypadku to po prostu wysokości słupków.

Lekcja 4. Rozwiązywanie klasycznych gier. Czas: 60-90 minut.

Ostatnia lekcja poświęcona jest rozwiązywaniu zadań polegających na znajdowaniu pozycji wygrywających i przegrywających w różnych grach. Czas jej trwania jest uzależniony głównie od tempa pracy uczniów nad opracowywaniem rozwiązań, jak również nad implementacją algorytmów. Poniższe gry podane są w kolejności od najłatwiejszej do najtrudniejszej.

Uogólniona gra w zabieranie

W omawianej przez nas wersji gry w zabieranie dopuszczaliśmy zabranie 1, 2 lub 3 monet. Nic nie stoi jednak na przeszkodzie, aby zbiór dopuszczalnych ruchów $\{1, 2, 3\}$ zastąpić dowolnym innym, na przykład $\{2, 3, 7, 8\}$. W takiej grze gracze mogą w pojedynczym ruchu zabrać 2, 3, 7 lub 8 monet. Przegrywa ten, kto nie może wykonać legalnego ruchu. Pozycje wygrywające i przegrywające oraz odpowiednie ruchy wyznaczamy standardowo korzystając z ich definicji. Zadaniem dla uczniów może być na przykład zmodyfikowanie programu z lekcji nr 1 tak, aby przed rozpoczęciem gry pytał użytkownika o zbiór dopuszczalnych ruchów.

Gra Wythoffa

Gra Wythoffa jest rozgrywana na zwykłej szachownicy. Na pewnym polu ustawiony jest hetman. Gracze na przemian przesuwiają hetmana, przy czym dozwolone są tylko ruchy w lewo, w dół lub ukośnie w lewy dół (o dowolną liczbę pól). Wygrywa gracz któremu uda się ustawić hetmana w lewym dolnym rogu szachownicy.

Aby rozwiązać to zadanie wystarczy potraktować pola szachownicy i dozwolone ruchy jako graf i wyznaczyć pozycje wygrywające i przegrywające wprost z definicji dla każdego pola.

Gra Kayles

Gra Kayles rozpoczyna się od pozycji, w której n butelek jest ustawionych w rzędzie tak, że odległość pomiędzy sąsiednimi butelkami wynosi dokładnie 10 cm. Gracze na przemian rzucają kulami do kręgli w stronę butelek. W jednym ruchu gracz może rozbić jedną dowolną butelkę, lub dwie butelki, ale tylko pod warunkiem, że stoją one obok siebie (odległość między nimi wynosi 10 cm). Zakładamy, że gracze zawsze trafiają, przez co w każdym ruchu co najmniej jedna butelka zostaje rozbita. Wygrywa gracz, który rozbija ostatnią butelkę. Który z graczy ma strategię wygrywającą i jakie powinien wykonywać ruchy?

Rozwiązanie gry Kayles nie jest trudne. Wystarczy zauważyć, że gdy rozbijemy butelki stojące z brzegu to otrzymamy tę samą grę Kayles z mniejszą liczbą butelek. Natomiast, gdy rozbijemy butelkę (lub butelki) ze środka rzędu to podzielimy naszą planszę na dwa rzędy, które możemy rozpatrywać niezależnie (przez to, że rzędy odseparowane są przerwą większą niż 10 cm, nie da się w jednym ruchu zbić butelek z różnych rzędów). Skoro gry te są niezależne, to możemy potraktować je jako jedną grę będącą sumą dwóch innych i zastosować twierdzenie Sprague-Grundy'ego. W rozwiązaniu wartości funkcji S możemy wyliczać stosując technikę programowania dynamicznego, poczynając od najmniejszych n .

Stworzenie własnej gry

Dodatkowo, ciekawym ćwiczeniem może być zaproponowanie uczniom, aby opracowali własne gry i przedstawili ich zasady. Następnie można wspólnie zastanowić się nad opracowaniem strategii wygrywających do gier, które wzbudzą największe zainteresowanie w grupie. Nauczyciel powinien przede wszystkim zwracać uwagę na to, czy proponowane przez uczniów gry spełniają warunki przedstawione na pierwszej lekcji, to znaczy, czy są dwuosobowe, skończone, bez remisów, z pełną informacją i wspólnym zbiorem ruchów.



INTEGRALNOŚĆ DANYCH – REGUŁY I OGRANICZENIA

Andrzej Ptasznik

Warszawska Wyższa Szkoła Informatyki

aptaszni@wwsi.edu.pl

1. Zagadnienia programowe

Integralność danych pojawia się w kontekście tworzenia relacyjnej bazy danych, tylko na rozszerzonym poziomie zajęć. Bazy danych, od których na ogół wymagamy integralności danych, występują również na poziomie podstawowym.

Informatyka, IV etap edukacyjny, zakres podstawowy

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) znajduje dokumenty i informacje w udostępnianych w Internecie bazach danych (np. bibliotecznych, statystycznych, w sklepach internetowych), ocenia ich przydatność i wiarygodność i gromadzi je na potrzeby realizowanych projektów z różnych dziedzin;
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 6) tworzy bazę danych, posługuje się formularzami, porządkuje dane, wyszukuje informacje stosując filtrowanie;
 - 7) wykonuje podstawowe operacje modyfikowania, i wyszukiwania informacji na relacyjnej bazie danych;

Informatyka, IV etap edukacyjny, zakres rozszerzony

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
 - 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
 - 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;
 - 5) opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora ogniowa.

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe i powinien być adresowany do uczniów z lepszym przygotowaniem informatycznym i wykazujących większe zainteresowania pogłębieniem wiedzy z zakresu baz danych.

2. Temat zajęć

Integralność danych – reguły i ograniczenia

Zajęcia są poświęcone omówieniu problemów związanych z zapewnieniem integralności danych z wykorzystaniem Systemu Zarządzania Bazami Danych MS SQL Server 2008 R2 Express Edition.

Komentarz. Omówienie istoty stosowania reguł i ograniczeń na etapie implementacji bazy danych zrealizowane powinno być na prostym przykładzie, zawierającym odniesienia do różnych problemów związanych z zapewnieniem integralności danych.

3. Cele zajęć

W wyniku realizacji tych zajęć uczeń powinien umieć:

- identyfikować problemy związane z integralnością danych dla konkretnego przykładu;
- definiować w języku SQL odpowiednie postaci ograniczeń;
- sprawdzać działanie zdefiniowanych reguł;
- rozumieć i umieć wyjaśnić działanie reguł i ograniczeń.

Komentarz. Cele tych zajęć wykraczają poza podstawę programową.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni znać:

- podstawowe pojęcia z zakresu baz danych, takie jak: klucz podstawowy, klucz obcy, klucz potencjalny;
- podstawy tworzenia relacyjnych baz danych;
- podstawy języka SQL.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (algorytmu),
 - implementacja (realizacja) rozwiązania w postaci programu komputerowego,
 - testowanie i ewaluacja rozwiązania komputerowego,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- przygotowanie przez uczniów listy problemów związanych z integralnością danych, które mogą wystąpić w omawianym przykładzie bazy danych;
- samodzielne sporządzenie przez uczniów opisów sposobów rozwiązania poszczególnych problemów;
- zapisanie odpowiednich poleceń w języku SQL i ich testowanie;



- samodzielne testowanie poprawności zdefiniowanych reguł dla odpowiednio dobranych danych;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania problemów związanych z integralnością danych dla określonego przykładu bazy danych;
- każdy uczeń tworzy swoją bazę danych a w niej tabele, dla których będą definiowane ograniczenia;
- polecenia definiujące konkretne ograniczenia uczniowie piszą samodzielnie;
- testowanie działania zdefiniowanych reguł i ograniczeń jest realizowane indywidualnie przez poszczególnych uczniów;
- końcowym efektem pracy nad danym problemem jest prezentacja działania zdefiniowanych reguł.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z tablicy lub z tablicy z kartkami papieru, gdzie zapisują specyfikacje problemów, opisy sposobów rozwiązania, postaci poleceń w języku SQL.

Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- podręcznik, np.: E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, *Informatyka, zakres rozszerzony*, WSiP, Warszawa 2004.
- materiały dotyczące baz danych, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, w tym oprogramowanie MS SQL2008 R2 Express Edition oraz MS SQL Server Management Studio.

Komentarz. Oprogramowanie MS SQL Server 2008 R2 Express Edition jest darmowe; można je pobrać ze strony <http://www.microsoft.com/express/Database>.

8. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenia reguł i ograniczeń, zapewniających integralność danych, mogą być rozłożone na trzy kolejne lekcje. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel objaśnia istotę problemów związanych z zapewnieniem integralności danych oraz wyjaśnia rolę reguł i ograniczeń definiowanych w bazach danych (Lekcja 1);
- następnie uwaga zostaje skupiona na omówienie przykładowego fragmentu bazy danych i identyfikacji problemów, które powinny zostać rozwiązane (Lekcja 1);
- uczniowie przystępują do tworzenia bazy danych i przykładowych tabel (Lekcja 1);
- następnie uczniowie zajmują się kolejnymi problemami, dla których należy zdefiniować odpowiednie ograniczenia (Lekcja 2);

- kolejnym etapem jest zapoznanie się ze składnią poleceń języka SQL, które będą wykonywane przy definiowaniu reguł i ograniczeń (Lekcja 2);
- w ostatnim etapie uczniowie powinni wykonać odpowiednie polecenia, przygotowane na etapach wcześniejszych, i testować ich działanie (Lekcja 3);
- rozwiązywanie każdego z zagadnień składa się z sześciu etapów, wyżej wymienionych (w Metodach pracy).

Lekcja 1. Tworzenie przykładowej bazy danych i definiowanie problemów związanych z integralnością danych. Czas: 30 min.

W tej części lekcji należy przedstawić przykładowe tabele, które uczniowie utworzą w swojej bazie danych oraz omówić, jakie problemy związane z integralnością danych mogą wystąpić.

Osoby			
	Column Name	Data Type	Allow Nulls
	idosoby	int	<input type="checkbox"/>
	Nazwisko	varchar(64)	<input type="checkbox"/>
	Imie	varchar(32)	<input type="checkbox"/>
	Pesel	char(11)	<input type="checkbox"/>
	DataUrodzenia	date	<input type="checkbox"/>
	CzyKobieta	bit	<input type="checkbox"/>
	idmiasta	int	<input type="checkbox"/>
			<input type="checkbox"/>

Rysunek 1. Tabela Osoby

Miasta			
	Column Name	Data Type	Allow Nulls
	idmiasta	int	<input type="checkbox"/>
	Nazwa	varchar(64)	<input type="checkbox"/>
			<input type="checkbox"/>

Rysunek 2. Tabela Miasta

Na rysunkach 1 i 2 przedstawiono przykładowe tabele, w których dla poszczególnych kolumn określono typ danych, zdefiniowano klucze podstawowe (kolumna idosoby w tabeli Osoby i kolumna idmiasta w tabeli Miasta). Po omówieniu przeznaczenia przykładowych tabel uczniowie powinni przystąpić do tworzenia własnej bazy danych i w niej tabel, pokazanych na rysunkach 1 i 2.

Do realizacji zadania uczniowie wykorzystują SQL Server Management Studio. Każdy uczeń wykonuje polecenie:

CREATE DATABASE *Nazwiskolmie*

Tworzenie tabel można realizować poprzez napisanie polecenia w języku SQL. Dla tabeli Osoby polecenie to miałoby następującą postać;

```

CREATE TABLE Osoby (
    [idosoby] [int] IDENTITY(1,1) NOT NULL,
    [Nazwisko] [varchar](64) NOT NULL,
    [Imie] [varchar](32) NOT NULL,
    [Pesel] [char](11) NOT NULL,
    [DataUrodzenia] [date] NOT NULL,
    [CzyKobieta] [bit] NOT NULL,
    [idmiasta] [int] NOT NULL,
CONSTRAINT [PK_Osoby] PRIMARY KEY CLUSTERED
(
    [idosoby] ASC
),

```

lub z wykorzystaniem okienkowego projektanta tabel, uruchamianego z poziomu SQL Server Management Studio.

Można poprosić uczniów, by postarali się wskazać, jakie problemy związane z integralnością danych, mogą wystąpić w omawianym przykładzie. W dalszej części lekcji będziemy rozwiązywać następujące problemy:

- numer Pesel powinien być unikatowy w tabeli, chociaż nie jest kluczem podstawowym (klucz potencjalny);
- nazwa miasta w tabeli Miasta również powinna przyjmować wartości unikatowe;
- zadeklarowany typ char(11) dla numeru Pesel dopuszcza jako poprawne wszystkie ciągi znakowe o długości nie większej niż 11 znaków a powinno to być dokładnie 11 cyfr;
- kolumna idmiasta w tabeli Osoby może przyjmować tylko te wartości, które występują w tabeli miasta;
- sześć pierwszych cyfr numeru Pesel musi być zgodne z data urodzenia;
- przedostatnia cyfra numeru Pesel musi być zgodna z płcią danej osoby (kobiety – cyfra parzysta, mężczyźni – cyfra nieparzysta);
- nie można usuwać wierszy z tabeli Miasta, jeżeli istnieje choć jedna osoba powiązana z tym miastem (wartość kolumny id miasta).

Z tych przykładów powinna wypłynąć konkluzja, iż proces zapewnienia integralności danych jest złożony, a pierwszym etapem rozwiązania problemu jest wyszukanie zagrożeń.

Lekcja 2. Definiowanie ograniczeń. Czas: 60 min.

Zajęcia powinny rozpocząć się od przypomnienia problemów, jakie zidentyfikowano w przykładowych tabelach na pierwszej lekcji. Rozwiązanie kolejnych problemów będzie sprowadzało się do zdefiniowania odpowiednich ograniczeń, które zapewnią wymuszenie reguł gwarantujących integralność danych w omawianych tabelach.

Ograniczenie UNIQUE

Ograniczenie typu UNIQUE zapewnia wymuszenie unikalności danych w obrębie tabeli dla kolumn nie będących kluczem podstawowym. Ograniczenia dla danej tabeli definiujemy w języku SQL poleceniem **ALTER TABLE**. Przykładowe polecenia wymuszające unikalność numeru Pesel będzie miało postać:

```

ALTER TABLE Osoby
ADD CONSTRAINT UnikalnoscPesel UNIQUE NONCLUSTERED (pesel)

```

Polecenie dodaje do bazy danych ograniczenie typu UNIQUE o nazwie UnikalnoscPesel, które będzie powiązane z tabelą Osoby i wymusi przechowywanie w tabeli wierszy o unikatowych wartościach numeru Pesel. Przy próbie modyfikacji wiersza lub zapisania nowego z wartością Pesel, która już w tabeli istnieje, zostanie wygenerowany następujący komunikat o błędzie:

Problem zapewnienia unikalności nazwy miasta w tabeli miasta rozwiązujemy podobnie a przykładowe polecenie może mieć następującą postać:

```
ALTER TABLE Miasta
ADD CONSTRAINT UnikalnoscNazwyMiastal UNIQUE NONCLUSTERED (nazwa).
```

```
Error Message: Violation of UNIQUE KEY constraint 'UnikalnoscPesel'.
Cannot insert duplicate key in object 'dbo.Osoby'.
The statement has been terminated.
```

Ograniczenie CHECK

Ograniczenie typu CHECK jest wyrażeniem logicznym, które testuje modyfikowane dane, jeżeli wyrażenie w wyniku daje wartość false, to modyfikacja nie może być wykonana i generowany jest odpowiedni błąd. Ograniczenia dla danej tabeli definiujemy w języku SQL poleceniem ALTER TABLE. Przykładowe polecenia, wymuszające by numer Pesel składał się dokładnie z jedenastu cyfr, będzie miało postać:

```
ALTER TABLE Osoby
ADD CONSTRAINT PeselWarunek1
CHECK (pesel like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]').
```

Polecenie dodaje do bazy danych ograniczenie typu CHECK o nazwie PeselWarunek1, które będzie powiązane z tabelą Osoby i wymusi zapisywanie w tabeli wartości numeru Pesel zgodne z podanym wzorcem (11 cyfr). Przy próbie modyfikacji wiersza lub zapisania nowego z wartością Pesel, która nie spełnia zdefiniowanego ograniczenia, zostanie wygenerowany następujący komunikat o błędzie:

Ograniczenie PeselWarunek1 odnosi się jedynie do samej wartości tej kolumny, niezależnie od wartości danych w pozostałych kolumnach. Należy pamiętać, że numer Pesel musi być zgodny z datą urodzenia i płcią danej osoby. W celu zapewnienia tej zgodności należy zdefiniować kolejne ograniczenie:

```
Error Message: The UPDATE statement conflicted with the CHECK
constraint "PeselWarunek1". The conflict occurred in database
"Przyklady", table "dbo.Osoby", column 'Pesel'.
```

```
ALTER TABLE Osoby
ADD CONSTRAINT PeselWarunek2
CHECK (
    DataUrodzenia=cast('19'+substring(pesel,1,6) as date) and
CASE
    WHEN CzyKobieta=1 and substring(pesel,10,1) % 2=0 THEN 1
    WHEN CzyKobieta=0 and substring(pesel,10,1) % 2=1 THEN 1
    ELSE 0
END=1
)
```



Polecenie dodaje do bazy danych ograniczenie typu CHECK o nazwie PeselWarunek2, które będzie powiązane z tabelą Osoby i wymusi zapisywanie w tabeli wartości numeru Pesel zgodne z datą urodzenia i płcią danej osoby. Przy próbie modyfikacji wiersza lub zapisania nowego z wartością Pesel, która nie spełnia zdefiniowanego ograniczenia, zostanie wygenerowany następujący komunikat o błędzie:

Error Message: The UPDATE statement conflicted with the CHECK constraint "PesleWarunek2". The conflict occurred in database "Przykłady", table "dbo.Osoby".

Ograniczenie FOREIGN KEY

Ograniczenie typu FOREIGN KEY definiuje powiązanie klucza obcego tabeli z wartością klucza podstawowego innej tabeli. W omawianym przykładzie kluczem obcym jest kolumna idmiasta w tabeli Osoby, a powinna zostać powiązana z kluczem podstawowym idmiasta w tabeli Miasta. Zdefiniowanie ograniczenia FOREIGN KEY zapewnia, że wartość idmiasta w tabeli Osoby zawsze znajdzie odpowiadający jej wiersz w tabeli Miasta. Ograniczenia dla danej tabeli definiujemy w języku SQL poleceniem ALTER TABLE. Przykładowe polecenia, definiujące ograniczenie klucza obcego (integralność referencyjna), ma postać;

```
ALTER TABLE Osoby
ADD CONSTRAINT KluczObcyIdMiasta
FOREIGN KEY(idmiasta)
REFERENCES Miasta (idmiasta)
```

Polecenie dodaje do bazy danych ograniczenie typu FOREIGN KEY o nazwie KluczObcyIdMiasta, które będzie powiązane z tabelą Osoby i wymusi przyjmowanie tylko takich wartości dla kolumny idmiasta, które mają odpowiednik w tabeli Miasta. Przy próbie wykonania dowolnej operacji, która nie spełnia tego ograniczenia, zostanie wygenerowany następujący komunikat o błędzie:

Error Message: The DELETE statement conflicted with the REFERENCE constraint "FK_Osoby_Miasta". The conflict occurred in database "Przykłady", table "dbo.Osoby", column 'idmiasta'.

Komentarz: W opisanym przykładzie zgodność daty urodzenia z numerem Pesel jest zapewniona dla osób urodzonych w XX wieku, można zaproponować uczniom rozbudowę ograniczenia CHECK w taki sposób, żeby sprawdzało także poprawność dla osób urodzonych w XXI wieku.

Lekcja 3. Testowanie rozwiązań problemów integralności danych.

Czas: 30 min.

Po zdefiniowaniu ograniczeń dla przykładowych tabel uczniowie powinni zademonstrować ich działanie. W tym celu powinni przygotować zestaw poleceń modyfikujących dane (INSERT, UPDATE i DELETE) i zademonstrować reakcje bazy danych na próby wprowadzenia zmian, które są niezgodne z założonym poziomem integralności danych. Należy omówić z uczniami znaczenie problemów integralności danych w rzeczywistych systemach. Uczniom szczególnie zainteresowanym omawianymi problemami można zaproponować stworzenie ograniczenia, które sprawdzi poprawność liczby kontrolnej numeru Pesel. Opis algorytmu obliczającego liczbę kontrolną można znaleźć na stronie:

<http://www.mswia.gov.pl/portal/pl/381/32>

WYKORZYSTANIE XML W RELACYJNYCH BAZACH DANYCH

Andrzej Ptasznik

Warszawska Wyższa Szkoła Informatyki

aptaszni@wwsi.edu.pl

1. Zagadnienia programowe

XML (ang. *Extensible Markup Language*), w wolnym tłumaczeniu *Rozszerzalny Język Znaczników*, jest obecnie powszechnie stosowanym standardem wymiany danych. W relacyjnych bazach danych wprowadzono typ danych XML, udostępniający wiele metod działających na danych XML, oraz wprowadzono dodatkowe klauzule w poleceniu SELECT języka SQL, które umożliwiają przekształcanie wyniku zapytania do postaci dokumentu XML. W podstawie programowej język XML nie jest uwzględniony, ale ze względu na jego bardzo powszechne stosowanie warto zapoznać uczniów uzdolnionych z aspektami jego wykorzystania w kontekście relacyjnych baz danych.

Informatyka, IV etap edukacyjny, zakres podstawowy

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) znajduje dokumenty i informacje w udostępnianych w Internecie bazach danych (np. bibliotecznych, statystycznych, w sklepach internetowych), ocenia ich przydatność i wiarygodność i gromadzi je na potrzeby realizowanych projektów z różnych dziedzin;
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 6) tworzy bazę danych, posługuje się formularzami, porządkuje dane, wyszukuje informacje stosując filtrowanie;
 - 7) wykonuje podstawowe operacje modyfikowania, i wyszukiwania informacji na relacyjnej bazie danych;

Informatyka, IV etap edukacyjny, zakres rozszerzony

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
 - 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
 - 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;
 - 5) opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora ogniowa.

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe i powinien być adresowany do uczniów z lepszym przygotowaniem informatycznym i wykazujących większe zainteresowania pogłębieniem wiedzy z zakresu baz danych.

2. Temat zajęć

Wykorzystanie XML w relacyjnych bazach danych

Zajęcia są poświęcone różnym aspektom wykorzystania języka XML w relacyjnych bazach danych. Omówione zostaną:

- klauzula FOR XML w poleceniu SELECT;
- wykorzystanie metod typu XML do realizacji zapytań;
- przechowywanie danych w kolumnach typu XML;
- wykorzystanie operatora złączeń CROSS APPLY do pobierania danych z wielu dokumentów XML.

Komentarz. Jest to trudny temat, ponieważ łączy w sobie problemy przechowywania i wyszukiwania danych w relacyjnej bazie danych z przechowywaniem i wyszukiwaniem danych w dokumentach XML.



3. Cele zajęć

W wyniku realizacji tych zajęć uczeń powinien umieć:

- przekształcać wyniki zapytań SQL do postaci dokumentów XML;
- wykorzystywać metody typu XML do pozyskiwania danych;
- stosować operator złączeń CROSS APPLY w zapytaniach.

Komentarz. Cele tych zajęć wykraczają poza podstawę programową.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów;
- znać podstawowe pojęcia z zakresu informatyki i baz danych;
- umieć pisać proste zapytania w języku SQL.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania,
 - implementacja (realizacja) rozwiązania w postaci polecenia w języku SQL,

- testowanie i ewaluacja rozwiązania,
- prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- przygotowanie przez uczniów listy problemów związanych z realizacją zadania;
- samodzielne sporządzenie przez uczniów opisów sposobów rozwiązania poszczególnych problemów;
- zapisanie odpowiednich poleceń w języku SQL i ich testowanie;
- samodzielne testowanie poprawności zapisanych poleceń;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

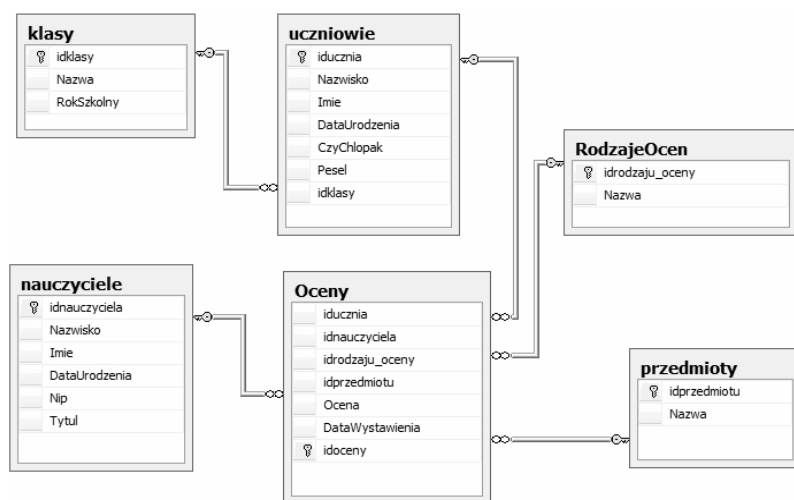
Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania sposobu rozwiązania problemu;
- przygotowywanie opisów rozwiązania może odbywać się indywidualnie lub w parach uczniów;
- polecenia SQL uczniowie piszą samodzielnie;
- testowanie poleceń może odbywać się w grupach uczniów lub samodzielnie (w zależności od stopnia zaawansowania);
- końcowym efektem pracy nad danym problemem jest prezentacja napisanego polecenia oraz wyników jego działania.

Uwaga: W trakcie lekcji należy podkreślać fakt, że dokument XML jest pewną formą bazy danych.

7. Materiały pomocnicze

Do przeprowadzenia zajęć niezbędne jest zainstalowanie oprogramowania SQL Server 2008 R2 Express Edition (darmowe oprogramowanie możliwe do pobrania ze stron firmy Microsoft). Po zainstalowaniu oprogramowania serwera baz danych należy utworzyć przykładową bazę danych, a w niej zdefiniować tabele, według schematu przedstawionego na rysunku 1.



Rysunek 1. Schemat bazy danych ElektronicznyDziennikOcen

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- materiały dotyczące baz danych, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, w tym oprogramowanie MS SQL2008 R2 Express Edition oraz MS SQL Server Management Studio.

Komentarz. Oprogramowanie MS SQL Server 2008 R2 Express Edition jest darmowe, można je pobrać ze strony <http://www.microsoft.com/express/Database>.

9. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenie do wykorzystania dokumentów XML w relacyjnych bazach danych mogą być rozłożone na kilka lekcji, niekoniecznie kolejnych. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel objaśnia budowę dokumentu XML i omawia podstawowe elementy składni klauzuli FOR XML wykorzystywanej w poleceniu SELECT (Lekcja 1);
- na kolejnym etapie wyjaśniane jest wykorzystywanie metod typu XML (Lekcja 2);
- na kolejnym etapie uczniowie wykonują przykłady zapytań w których wykorzystywane są jednocześnie, dane z tabel relacyjnych i dane z XML(Lekcja 3);
- rozwiązywanie każdego z zagadnień składa się z sześciu etapów, wyżej wymienionych (w Metodach pracy).

Lekcja 1. Przekształcanie wyników zapytań do postaci dokumentów XML.

Czas: 60 min.

W tej części zajęć uczniowie zapoznają się z działaniem klauzuli FOR XML polecenia SELECT języka SQL, która umożliwi przekształcanie wyników zapytania do postaci dokumentu XML. Pierwsze próby wykują na bazie następującej postaci zapytania;

```
SELECT TOP 3 Nazwisko,
           Imie,
           Pesel,
           DataUrodzenia,
           CASE CzyChlopak
            WHEN 1 THEN 'Mężczyzna'
            ELSE 'Kobieta'
           END AS Płeć
FROM Uczniowie
```

Przykładowy wynik tego zapytania jest przedstawiony na rysunku 2.

Nazwisko	Imie	Pesel	DataUrodzenia	Płeć
Kotek	Kasia	92031275446	1992-03-12	Kobieta
Piesek	Jan	92051587746	1992-05-15	Mężczyzna
Lisek	Kasia	92022277654	1992-02-22	Kobieta

Rysunek 2. Przykładowa postać wyniku zapytania

W kolejnym kroku należy zademonstrować uczniom najprostszą postać przekształcenia wyniku zapytania do postaci XML. Osiągniemy to dodając do przykładowego zapytania klauzulę FOR XML RAW. Zapytanie będzie miało następującą postać;

```
SELECT TOP 3 Nazwisko,
       Imie,
       Pesel,
       DataUrodzenia,
       CASE CzyChlopak
         WHEN 1 THEN 'Mężczyzna'
         ELSE 'Kobieta'
       END AS Płeć
FROM Uczniowie
FOR XML RAW.
```

Wynik tego zapytania ma postać przedstawioną na rysunku 3.

```
<row Nazwisko="Kotek" Imie="Kasia" Pesel="92031275446" DataUrodzenia="1992-03-12" Płeć="Kobieta" />
<row Nazwisko="Piesek" Imie="Jan" Pesel="92051587746" DataUrodzenia="1992-05-15" Płeć="Mężczyzna" />
<row Nazwisko="Lisek" Imie="Kasia" Pesel="92022277654" DataUrodzenia="1992-02-22" Płeć="Kobieta" />
```

Rysunek 3. Wynik zapytania w postaci XML

Omawiając ten przykład należy zwrócić uczniom uwagę na budowę tego wyniku. Każdy wiersz zwracany przez zapytanie został przekształcony w element o nazwie **row** a kolejne kolumny są umieszczone wewnątrz elementu jako atrybuty. Prawidłowy dokument XML powinien zawierać element główny (ang. *root*) a w pokazanym przykładzie takiego elementu nie ma.

W kolejnych krokach należy przedstawić modyfikacje klauzuli FOR XML, w wyniku których będzie można otrzymać różne postaci dokumentu XML. Ostateczna postać klauzuli FOR XML będzie miała następującą postać:

```
FOR XML RAW('Uczen'), ELEMENTS, ROOT('ListaUczniow'),
```

a wynik zapytania jest pokazany na rysunku 4.

```
<ListaUczniow>
  <Uczen>
    <Nazwisko>Kotek</Nazwisko>
    <Imie>Kasia</Imie>
    <Pesel>92031275446</Pesel>
    <DataUrodzenia>1992-03-12</DataUrodzenia>
    <Płeć>Kobieta</Płeć>
  </Uczen>
  <Uczen>
    <Nazwisko>Piesek</Nazwisko>
    <Imie>Jan</Imie>
    <Pesel>92051587746</Pesel>
    <DataUrodzenia>1992-05-15</DataUrodzenia>
    <Płeć>Mężczyzna</Płeć>
  </Uczen>
  <Uczen>
    <Nazwisko>Lisek</Nazwisko>
    <Imie>Kasia</Imie>
    <Pesel>92022277654</Pesel>
    <DataUrodzenia>1992-02-22</DataUrodzenia>
    <Płeć>Kobieta</Płeć>
  </Uczen>
</ListaUczniow>
```

Rysunek 4. Przykładowa postać wyniku zapytania



Doprowadzenie wyniku do postaci pokazanej na rysunku 4 można realizować sekwencyjnie, pokazując działanie poszczególnych opcji klauzuli FOR XML:

- RAW('Uczen') – zastosowanie tej opcji powoduje nadanie nazwy elementowi dokumentu XML (zamiast domyślnej nazwy row);
- ELEMENTS – zastosowanie tej opcji powoduje budowanie wyniku w postaci elementów a nie atrybutów, jak pokazano na rysunku 3;
- ROOT('ListaUczniow') – opcja powodująca dodanie do wyniku elementu głównego dokumentu XML.

W kolejnym przykładzie można pokazać przekształcenie do postaci dokumentu o budowie mieszanej (część danych w atrybutach a część danych w elementach), co możemy osiągnąć stosując w klauzuli FOR XML opcję PATH zamiast opcji RAW. Zapytanie będzie miało postać:

```
SELECT TOP 3 Nazwisko as „@Nazwisko”,
      Imie as “@Imie”,
      Pesel,
      DataUrodzenia,
      CASE CzyChłopak
      WHEN 1 THEN ‘Mężczyzna’
      ELSE ‘Kobieta’
      END AS Płeć
FROM Uczniowie
FOR XML PATH('Uczen'), ROOT('ListaUczniow').
```

Wynik tego zapytania jest pokazany na rysunku 5. Zastosowana w zapytaniu opcja PATH klauzuli FOR XML, umożliwia umieszczanie wybranych danych w atrybutach, co osiągamy nadając zastępczą kolumnę (np. **Imie as „@Imie”**) umieszczoną w cudzysłowach i poprzedzoną znakiem @.

```
<ListaUczniow>
  <Uczen Nazwisko="Kotek" Imie="Kasia">
    <Pesel>92031275446</Pesel>
    <DataUrodzenia>1992-03-12</DataUrodzenia>
    <Płeć>Kobieta</Płeć>
  </Uczen>
  <Uczen Nazwisko="Piesek" Imie="Jan">
    <Pesel>92051587746</Pesel>
    <DataUrodzenia>1992-05-15</DataUrodzenia>
    <Płeć>Mężczyzna</Płeć>
  </Uczen>
  <Uczen Nazwisko="Lisek" Imie="Kasia">
    <Pesel>92022277654</Pesel>
    <DataUrodzenia>1992-02-22</DataUrodzenia>
    <Płeć>Kobieta</Płeć>
  </Uczen>
</ListaUczniow>
```

Rysunek 5. Wynik zapytania – dokument XML o budowie mieszanej

Kolejny przykład dotyczy budowy dokumentu XML z wykorzystaniem podzapytań. W tym przykładzie chcemy rozbudować wynikową postać dokumentu XML o informacje o dwóch ostatnich ocenach otrzymanych przez danego ucznia. Zapytanie będzie miało następującą postać:

```

SELECT TOP 2 Nazwisko as „@Nazwisko, Imie as „@Imie“,
    Pesel, DataUrodzenia,
    CASE CzyChlopak
      WHEN 1 THEN 'Mężczyzna'
      ELSE 'Kobieta'
    END AS Płeć,
    (
      SELECT TOP 2 Nazwa as Przedmiot,
        Ocena,
        DataWystawienia
      FROM Oceny join Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
      WHERE iducznia=Uczniowie.iducznia
      ORDER BY DataWystawienia DESC
      FOR XML RAW('Ocena'),ELEMENTS, TYPE, ROOT('OstatnieOceny')
    )
FROM Uczniowie
FOR XML PATH('Uczen'), ROOT('ListaUczniow')

```

Wynik zapytania jest przedstawiony na rysunku 6. Należy omówić z uczniami nowe elementy wykorzystane w tym zapytaniu. Uczniom szczególnie zainteresowanym i uzdolnionym można zlecić wyjaśnienie znaczenia opcji TYPE wykorzystanej w zapytaniu wewnętrznym omawianego przykładu.

```

<ListaUczniow>
  <Uczen Nazwisko="Kotek" Imie="Kasia">
    <Pesel>92031275446</Pesel>
    <DataUrodzenia>1992-03-12</DataUrodzenia>
    <Płeć>Kobieta</Płeć>
    <OstatnieOceny>
      <Ocena>
        <Przedmiot>Literatura</Przedmiot>
        <Ocena>4.00</Ocena>
        <DataWystawienia>2009-02-15</DataWystawienia>
      </Ocena>
      <Ocena>
        <Przedmiot>Literatura</Przedmiot>
        <Ocena>3.00</Ocena>
        <DataWystawienia>2009-02-12</DataWystawienia>
      </Ocena>
    </OstatnieOceny>
  </Uczen>
  <Uczen Nazwisko="Piesek" Imie="Jan">...
</ListaUczniow>

```

Rysunek 6. Przykładowy wynik zapytania

Lekcja 2. Pobieranie danych z dokumentu XML. Czas: 60 min.

Celem tej lekcji jest zapoznanie uczniów ze sposobami wykorzystania metod typu danych XML do realizacji zapytań. Typ danych XML w środowisku SQL Server udostępnia 5 metod:

- metoda `value()` – zwraca z dokumentu XML wskazaną wartość skalarną;
- metoda `query()` – zwraca wskazane elementy dokumentu XML (wynik jest typu XML);
- metoda `nodes()` – zwraca tabelę;

- metoda exist() – zwraca wartość logiczną informującą, czy wskazany w parametrze metody element istnieje w dokumencie XML;
- metoda modify() – modyfikuje dokument XML.

W ramach lekcji prezentowane jest działanie metod: query(), value() i nodes().

Do realizacji ćwiczeń należy przygotować przykładowy skrypt, w którym jest zadeklarowana zmienna typu XML. Przypisujemy jej dokument XML otrzymany w czasie lekcji 1 (lista uczniów z dwoma ostatnimi ocenami). Skrypt może mieć następującą postać:

```

DECLARE @dane xml='<ListaUczniow>
<Uczen Nazwisko="Kotek" Imie="Kasia">
<Pesel>92031275446</Pesel>
<DataUrodzenia>1992-03-12</DataUrodzenia>
<Płeć>Kobieta</Płeć>
<OstatnieOceny>
<Ocena>
<Przedmiot>Literatura</Przedmiot>
<Ocena>4.00</Ocena>
<DataWystawienia>2009-02-15</DataWystawienia>
</Ocena>
<Ocena>
<Przedmiot>Literatura</Przedmiot>
<Ocena>3.00</Ocena>
<DataWystawienia>2009-02-12</DataWystawienia>
</Ocena>
</OstatnieOceny>
</Uczen>
<Uczen Nazwisko="Piesek" Imie="Jan">
<Pesel>92051587746</Pesel>
<DataUrodzenia>1992-05-15</DataUrodzenia>
<Płeć>Mężczyzna</Płeć>
<OstatnieOceny>
<Ocena>
<Przedmiot>Fizyka</Przedmiot>
<Ocena>5.00</Ocena>
<DataWystawienia>2009-02-09</DataWystawienia>
</Ocena>
<Ocena>
<Przedmiot>Informatyka</Przedmiot>
<Ocena>4.00</Ocena>
<DataWystawienia>2009-01-21</DataWystawienia>
</Ocena>
</OstatnieOceny>
</Uczen>
</ListaUczniow>';

```



Pierwszy przykład – wykorzystanie metody query()

Zadaniem jest wybranie wszystkich elementów DataUrodzenia zawartych w przykładowym dokumencie XML. W najprostszej postaci, jako parametr metody query() przekazujemy adres węzłów dokumentu XML. Wynikiem metody jest zbiór wskazanych elementów znajdujących się w dokumencie. Postać skryptu do realizacji tego zadania jest pokazana na rysunku 7.

```

+ DECLARE @dane XML='<ListaUczniow>...
|
|
| Select @dane.query('ListaUczniow/Uczen/DataUrodzenia')

```

Rysunek 7. Postać skryptu wykorzystującego metodę query()

Wynik zapytania zawiera elementy DataUrodzenia znalezione w dokumencie zapisanym w zmiennej @dane. W naszym przykładzie są to dwa następujące:

```

<DataUrodzenia>1992-03-12</DataUrodzenia>
<DataUrodzenia>1992-05-15</DataUrodzenia>

```

Omiawiając wykonanie tego przykładu należy zwrócić uwagę, że wynik metody jest fragmentem wyjściowego dokumentu XML i nie zawiera elementu głównego.

Drugi przykład – wykorzystanie metody value()

Metoda value() zwraca wskazaną wartość skalarną z dokumentu XML. Metoda wymaga dwóch parametrów: pierwszy – dokładnie adresuje wybierany element, a drugi – określa typ danych, do którego zostanie przekształcona odczytana wartość. Na rysunku 8 jest pokazana postać skryptu, który zwraca datę urodzenia zapisaną w drugim elemencie DataUrodzenia znanym w dokumencie XML.

```

+ DECLARE @dane XML='<ListaUczniow>...
|
|
| Select @dane.value('(ListaUczniow/Uczen/DataUrodzenia)[2]', 'date')

```

Rysunek 8. Postać skryptu wykorzystującego metodę value()

W wyniku działania skryptu pokazanego na rysunku 8 otrzymamy datę 1992-05-15. Szczególną uwagę należy zwrócić na konieczność jednoznacznego adresu wyszukiwanej wartości, ponieważ standard XML zezwala na wielokrotne wystąpienie elementu o danej nazwie w konkretnym kontekście. Dla zapewnienia jednoznacznej adresacji stosujemy indeks określający, o które wystąpienie szukanego elementu chodzi. Konstrukcja (ListaUczniow/Uczen/DataUrodzenia)[2] określa drugie wystąpienie elementu DataUrodzenia zawartego w elemencie Uczen, który jest zawarty w elemencie ListaUczniow.

Trzeci przykład – wykorzystanie metody nodes()

Metoda nodes() jest szczególnie ważna, ponieważ pozwala przekształcić dokument XML do postaci tabeli, co umożliwia wykorzystanie wyniku tego przekształcenia w zapytaniach SQL. Skrypt, wykorzystujący metodę nodes, jest pokazany na rysunku 9.


```

+ DECLARE @dane XML='<ListaUczniow>[...]'
- SELECT k.value('@Nazwisko', 'varchar(64)') as Nazwisko,
      k.value('@Imie', 'varchar(64)') as Imie,
      k.value('DataUrodzenia[1]', 'date') as DataUrodzenia,
      k.value('Pesel[1]', 'char(11)') as Nazwisko
FROM @dane.nodes('ListaUczniow/Uczen') t(k)

```

Rysunek 9. Postać skryptu wykorzystującego metode nodes()

Parametrem metody nodes() jest adres węzła dokumentu XML (w naszym przykładzie ListaUczniow/Uczen) i zwraca tyle wierszy, ile razy wskazany element występuje w dokumencie. Ponieważ w klauzuli FROM zapytania SELECT wykorzystujemy wyrażenie budujące tabelę, to wymogiem składniowym języka SQL jest, by nadać tej tabeli nazwę. Wykorzystana w omawianym przykładzie postać nazwy zastępczej t(k) oznacza, że tabela powstała z dokumentu XML będzie miała nazwę t a kolumna będąca wynikiem działania metody nodes() będzie miała nazwę k. Kolumna wynikowa, zwracana przez metodę nodes, nie może być wykorzystana bezpośrednio lecz jedynie z użyciem metod typu XML. W przykładzie z kolumny k pobierane są wartości z atrybutów @Nazwisko i @Imie oraz dane z elementów DataUrodzenia i Pesel. Wynik przykładowego skryptu jest pokazany na rysunku 10.

Nazwisko	Imie	DataUrodzenia	Nazwisko
Kotek	Kasia	1992-03-12	92031275446
Piesek	Jan	1992-05-15	92051587746

Rysunek 10. Wynik zastosowania metody nodes()

Zainteresowanym uczniom można polecić wykonanie zapytania, które zwróci nazwisko i imie ucznia oraz nazwę przedmiotu i wystawioną ocenę.

Lekcja 3. Zapytania wykorzystujące dane relacyjne i XML. Czas: 60 min.

Wprowadzenie typu XML do standardu SQL spowodowało, że projektując bazę danych, w tabelach relacyjnych są umieszczane kolumny zawierające dane XML. Zapytania języka SQL muszą korzystać z różnych postaci danych. W ramach tej lekcji są przedstawiane dwa przykłady takich zapytań. Przed przystąpieniem do ćwiczenia należy przygotować odpowiednią tabelę, która będzie zawierała kolumnę typu XML. W tym celu należy wykonać następujące zapytanie:

```

SELECT Nazwisko,
       Imie,
       Pesel,
       DataUrodzenia,
       CzyChlopak,
       (
         SELECT Nazwa as Przedmiot,
                Ocena,
                DataWystawienia
        FROM Oceny join Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu

```

```

WHERE iducznia=Uczniowie.iducznia
ORDER BY DataWystawienia DESC
FOR XML RAW('Ocena'),ELEMENTS, TYPE, ROOT('ListaOcen')
) as OcenyUcznia
INTO UczniowieXML
FROM Uczniowie

```

Klauzula INTO spowoduje, że w bazie danych zostanie utworzona tabela o nazwie UczniowieXML, będąca wynikiem tego zapytania. Przykładowa zawartość tej tabeli jest pokazana na rysunku 11.

Nazwisko	Imie	Pesel	DataUrodzenia	CzyChlopak	OcenyUcznia
Kotek	Kasia	92031275446	1992-03-12	0	<ListaOcen><Ocena><Przedmiot>Literatura</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-15</DataWystawienia></Ocena><Ocena><Przedmiot>Fizyka</Przedmiot><Ocena>3.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena><Ocena><Przedmiot>Literatura</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Piesek	Jan	92051587746	1992-05-15	1	<ListaOcen><Ocena><Przedmiot>Fizyka</Przedmiot><Ocena>5.00</Ocena><DataWystawienia>2009-02-15</DataWystawienia></Ocena><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>2.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Lisek	Kasia	92022277654	1992-02-22	0	<ListaOcen><Ocena><Przedmiot>Fizyka</Przedmiot><Ocena>3.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>1.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Kurka	Jola	92060288788	1992-06-02	0	<ListaOcen><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>2.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena><Ocena><Przedmiot>Literatura</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Gąska	Wacek	91031199123	1991-03-11	0	<ListaOcen><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>1.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena><Ocena><Przedmiot>Literatura</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Krówka	Rysio	92051577646	1992-05-15	1	<ListaOcen><Ocena><Przedmiot>Informatyka</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-15</DataWystawienia></Ocena><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>1.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>
Zebra	Wojtek	93031399846	1993-03-13	1	<ListaOcen><Ocena><Przedmiot>Geografia</Przedmiot><Ocena>1.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena><Ocena><Przedmiot>Literatura</Przedmiot><Ocena>4.00</Ocena><DataWystawienia>2009-02-12</DataWystawienia></Ocena></ListaOcen>

Rysunek 11. Przykładowa zawartość tabeli UczniowieXML

W kolumnie OcenyUcznia jest zapisany dokument XML w postaci, której przykład jest pokazany na rysunku 12.

```

<ListaOcen>
  <Ocena>
    <Przedmiot>Literatura</Przedmiot>
    <Ocena>4.00</Ocena>
    <DataWystawienia>2009-02-15</DataWystawienia>
  </Ocena>
  <Ocena>
    <Przedmiot>Literatura</Przedmiot>
    <Ocena>3.00</Ocena>
    <DataWystawienia>2009-02-12</DataWystawienia>
  </Ocena>
  <Ocena>
    <Przedmiot>Literatura</Przedmiot>
    <Ocena>4.00</Ocena>
    <DataWystawienia>2009-02-12</DataWystawienia>
  </Ocena>
</ListaOcen>

```

Rysunek 12. Przykładowa postać dokumentu XML

Pierwszy przykład – obliczenie średniej oceny ucznia

W tym przykładzie chcemy uzyskać tabelę wyników zawierającą nazwisko, imię i Pesel ucznia oraz jego średnią ocen. Zwracamy uczniom uwagę na fakt, że część danych możemy pobrać bezpośrednio z tabeli UczniowieXML, natomiast średnią ocenę możemy uzyskać wykonując zapytanie odwołujące się do kolumny OcenyUcznia (dane XML). Polecenie realizujące to zadanie ma następującą postać:

```

SELECT Nazwisko,
       Imie,
       Pesel,

```

```
(
    SELECT AVG(k.value('Ocena[1]';decimal(5,2)))
    FROM OcenyUcznia.nodes('ListaOcen/Ocena') t(k)
) as Średnia
FROM UczniowieXML.
```

Problem sprowadził się do wykonania dla każdego wiersza tabeli UczniowieXML zapytania obliczającego średnią z dokumentu XML, zapisanego w kolumnie OcenyUcznia. Przykładowy wynik zapytania jest przedstawiony na rysunku 13.

Nazwisko	Imie	Pesel	Średnia
Kotek	Kasia	92031275446	3.000000
Piesek	Jan	92051587746	2.980000
Lisek	Kasia	92022277654	3.229508
Kurka	Jola	92060288788	3.030769
Gąska	Wacek	91031199123	2.857142
Krówka	Rysio	92051577646	2.781818
Zebra	Wojtek	93031399846	3.075757
Gazela	Basia	92111177446	2.932432
Sarenka	Rysio	92121278766	2.563636
Konik	Kasia	93031275446	2.686274
Ryba	Jan	93051587746	3.123076
Kura	Kasia	93022277654	3.035087

Rysunek 13. Przykładowa postać wyniku zapytania

Drugi przykład – obliczenie średniej oceny z poszczególnych przedmiotów

W tym przykładzie, do wykonania zadania jest potrzebne pobranie danych ze wszystkich dokumentów XML zapisanych w kolumnie OcenyUcznia tabeli UczniowieXML. Do realizacji zadania zastosujemy operator złączenia tabel CROSS APPLY, który dla każdego wiersza tabeli umieszczonej po lewej stronie operatora, tworzy dynamicznie tabelę zgodnie z wyrażeniem umieszczonym po prawej stronie operatora i łączy ten wiersz z każdym wierszem otrzymanym w wyniku wykonania wyrażenia. Operator złączeń CROSS APPLY umożliwi połączenie wiersza opisującego danego ucznia z jego ocenami uzyskanymi z zapytania odwołującego się do kolumny OcenyUcznia. Zapytanie będzie miało postać:

```
SELECT Przedmiot,
    AVG(Ocena) as Średnia
FROM UczniowieXML CROSS APPLY
(
    SELECT k.value('Przedmiot[1]';'varchar(64)') as Przedmiot,
        k.value('Ocena[1]';'decimal(5,2)') as Ocena
    FROM OcenyUcznia.nodes('ListaOcen/Ocena') as t(k)
) as Oceny
GROUP BY Przedmiot.
```

W wyniku otrzymujemy tabelę zawierającą nazwę przedmiotu oraz średnią ocenę z danego przedmiotu – przykładowy wynik jest pokazany na rysunku 14.

Przedmiot	Średnia
Infomatyka	2.862745
Literatura	3.125000
Fizyka	2.891891
Geografia	3.058823
Matematyka	3.024590

Rysunek 14. Przykładowy wynik zapytania wykorzystującego operator CROSS APPLY

W ramach lekcji można polecić uczniom wykonanie zmodyfikowanych zapytań, dodając dodatkowe warunki. (np. średnia ocena z poszczególnych przedmiotów uzyskanych w bieżącym roku).



ZAPYTANIA REKURENCYJNE Z WYKORZYSTANIEM WYRAŻEŃ CTE W SQL SERVER

Andrzej Ptasznik

Warszawska Wyższa Szkoła Informatyki

aptaszni@wwsi.edu.pl

1. Zagadnienia programowe

Rekurencja pojawia się na lekcjach matematyki i informatyki (patrz rozdz. w tym opracowaniu), a bazy danych i język SQL – na lekcjach informatyki, zgodnie z poniższymi zapisami w podstawie programowej.

Informatyka, IV etap edukacyjny, zakres podstawowy

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) znajduje dokumenty i informacje w udostępnianych w Internecie bazach danych (np. bibliotecznych, statystycznych, w sklepach internetowych), ocenia ich przydatność i wiarygodność i gromadzi je na potrzeby realizowanych projektów z różnych dziedzin;
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 6) tworzy bazę danych, posługuje się formularzami, porządkuje dane, wyszukuje informacje stosując filtrowanie;
 - 7) wykonuje podstawowe operacje modyfikowania, i wyszukiwania informacji na relacyjnej bazie danych;

Informatyka, IV etap edukacyjny, zakres rozszerzony

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
 - 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
 - 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
 - 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;
 - 5) opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zaporą ogniową.

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe i powinien być adresowany do uczniów z lepszym przygotowaniem informatycznym i wykazujących większe zainteresowania pogłębieniem wiedzy z zakresu baz danych.

2. Temat zajęć

Zapytania rekurencyjne z wykorzystaniem wyrażeń CTE w SQL Server

Temat jest wprowadzeniem do stosowania wyrażeń CTE (ang. *Common Table Expressions*) w poleceniach języka SQL oraz użycia tych wyrażeń do zapytań rekurencyjnych.

Komentarz. Jest to trudny temat, ponieważ łączy w sobie problemy wyszukiwania danych w relacyjnej bazie danych z myśleniem rekurencyjnym.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- tworzyć zapytania w języku SQL z wykorzystaniem wyrażeń CTE,
- pisać zapytania rekurencyjne.

Komentarz. Cele tych zajęć wykraczają poza podstawę programową.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów;
- znać podstawowe pojęcia z zakresu informatyki i baz danych;
- umieć pisać proste zapytania w języku SQL;
- znać pojęcie rekurencji.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania,
 - implementacja (realizacja) rozwiązania w postaci polecenia w języku SQL,
 - testowanie i ewaluacja rozwiązania,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania.
- przygotowanie przez uczniów listy problemów związanych z realizacją zadania;
- samodzielne sporządzenie przez uczniów opisów sposobów rozwiązania poszczególnych problemów;
- zapisanie odpowiednich poleceń w języku SQL i ich testowanie;
- samodzielne testowanie poprawności zdefiniowanych zapisanych poleceń;
- prezentacja otrzymanych rozwiązań.



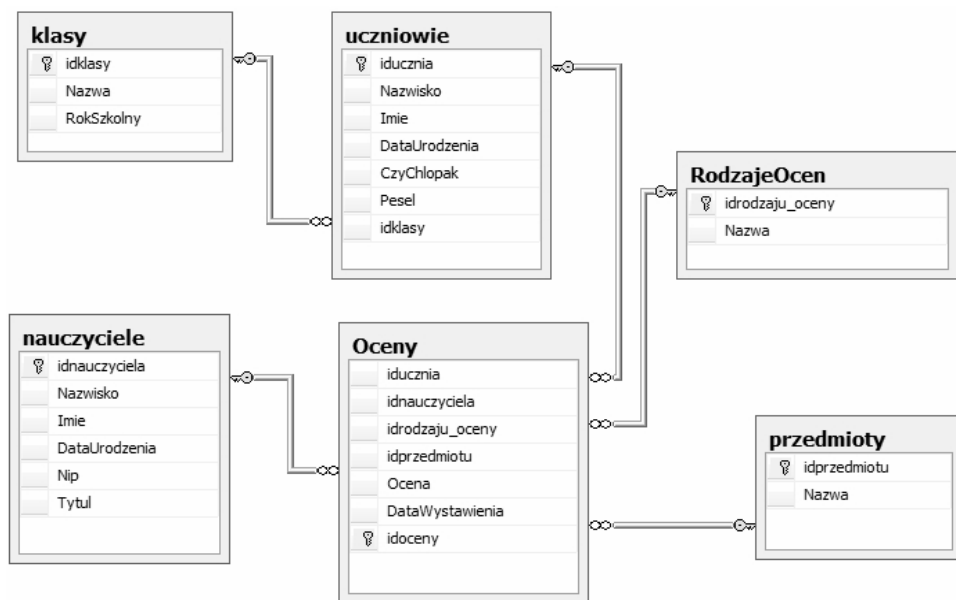
6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania sposobu rozwiązania problemu;
- przygotowywanie opisów rozwiązania może odbywać się indywidualnie lub w parach uczniów;
- polecenia SQL uczniowie piszą samodzielnie;
- testowanie poleceń może odbywać się w grupach uczniów lub samodzielnie (w zależności od stopnia zaawansowania);
- końcowym efektem pracy nad danym problemem jest prezentacja napisanego polecenia oraz wyników jego działania.

7. Materiały pomocnicze

Do przeprowadzenia zajęć niezbędne jest zainstalowanie oprogramowania SQL Server 2008 R2 Express Edition (darmowe oprogramowanie możliwe do pobrania ze stron firmy Microsoft). Po zainstalowaniu oprogramowania serwera baz danych należy utworzyć przykładową bazę danych a w niej zdefiniować tabele według schematu pokazanego na rysunkach 1 i 2.



Rysunek 1. Schemat bazy danych ElektronicznyDziennikOcen

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- podręcznik, np.: E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, *Informatyka, zakres rozszerzony, WSiP, Warszawa 2004.*

- materiały dotyczące baz danych, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, w tym oprogramowanie MS SQL2008 R2 Express Edition oraz MS SQL Server Management Studio.

Komentarz. Oprogramowanie MS SQL Server 2008 R2 Express Edition jest darmowe; można pobrać ze strony <http://www.microsoft.com/express/Database>.



Rysunek 2. Schemat bazy danych Pracownicy

9. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest wykorzystanie wyrażeń CTE do realizacji zapytań rekurencyjnych mogą być rozłożone na kilka lekcji, niekoniecznie kolejne. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel objaśnia składnię i zastosowanie wyrażeń CTE (Lekcja 1);
- na kolejnym etapie są realizowane złożone zapytania z wykorzystaniem wyrażeń CTE (Lekcja 2);
- następnie uczniowie zajmują się problemami, które mają naturę rekurencyjną w tym sensie, że ich sformułowania i wynikające z nich algorytmy są rekurencyjne (Lekcja 3);
- rozwiązywanie każdego z zagadnień składa się z sześciu etapów, wyżej wymienionych (w Metodach pracy).

Lekcja 1. Wyrażenia CTE w języku SQL. Czas: 30 min.

Wyrażenia **CTE** (ang. *Common Table Expression*) są nowym elementem języka SQL. Podstawowym celem wprowadzenia wyrażeń CTE było umożliwienie pisania zapytań rekurencyjnych. Dodatkowym zastosowaniem tych wyrażeń jest tworzenie nazwanych zbiorów danych, które mogą być wykorzystywane do realizacji jednego z poleceń manipulacji danymi (SELECT, INSERT, UPDATE, DELETE,

MERGE). Nazwane zbiory są tworzone dla konkretnego polecenia i przestają istnieć po jego wykonaniu. Podstawowa składnia tworzenia wyrażeń CTE jest następująca:

```
WITH Nazwa_zbioru AS
(
    Polecenie_Select
)
Polecenie SQL
```

Wykorzystanie wyrażeń CTE można wyjaśnić na prostych przykładach.

Pierwszy przykład: Na podstawie bazy danych ElektronocznDziennikOcen wybrać uczniów, którzy w rankingu według średniej ocen zajmują pozycje od 3 do 8.

Uwaga. W rozwiązaniu tego przykładowego zadania należy wykorzystać funkcję szeregującą Row_Number(). Problem można rozwiązać za pomocą następującego polecenia:

```
WITH RankingSrednich AS
(
    SELECT ROW_NUMBER() OVER (ORDER BY AVG(ocena) DESC ) as Lp,
           Nazwisko, Imie, Pesel, AVG(ocena) as Srednia
    FROM Uczniowie JOIN Oceny
         ON Uczniowie.iducznia=Oceny.iducznia
    GROUP BY Nazwisko, Imie, Pesel
)
SELECT Nazwisko, Imie, Pesel, Srednia
FROM RankingSrednich
WHERE Lp BETWEEN 3 and 8
```

Wyjaśniając istotę wyrażeń CTE należy w pierwszym kroku pokazać wynik zapytania wykorzystanego do tworzenia nazwanego zbioru danych (RankingSrednich).

Drugi przykład: Na podstawie bazy danych ElektronicznyDziennikOcen wybrać tych uczniów, których średnia ocena jest wyższa od średniej oceny jego klasy.

W rozwiązaniu tego problemu utworzymy dwa zbiory (średnie uczniów i średnie klas), a następnie połączymy te zbiory i wybierzemy te wiersze, dla których spełniony jest założony warunek. Problem można rozwiązać za pomocą następującego polecenia;

```
WITH SrednieKlas AS
(
    SELECT idklasy, AVG(ocena) AS SredniaKlasy
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
    GROUP BY idklasy
), SrednieUczniow AS
(
    SELECT Nazwisko, Imie, Pesel, idklasy, AVG(ocena) AS SredniaUcznia
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
    GROUP BY Nazwisko, Imie, Pesel, idklasy
)
```

```
SELECT Nazwisko, Imie, Pesel
FROM SrednieKlas JOIN SrednieUczniow ON SrednieKlas join SrednieUczniow
WHERE SredniaUcznia>SredniaKlasy
```

W omawianym przykładzie zdefiniowaliśmy dwa nazwane zbiory a następnie umieściliśmy zapytanie, które z tych zbiorów korzysta. Na zakończenie zajęć należy omówić z uczniami sposoby wykorzystania wyrażeń CTE zwracając uwagę na fakt, że dla jednego polecenia w języku SQL możemy zdefiniować wiele nazwanych zbiorów, dzięki czemu ostateczna postać polecenia jest prosta i oddaje istotę realizowanego zadania.

Lekcja 2. Wykorzystanie wyrażeń CTE do złożonych zapytań. Czas: 60 min.

Celem tej lekcji jest zilustrowanie, w jaki sposób wyrażenia CTE można wykorzystać do pisania poleceń realizujących złożone wybieranie danych. Definiowanie nazwanych zbiorów danych powinno ułatwić pisanie takich poleceń. W trakcie lekcji należy zwracać szczególną uwagę na prostą postać ostatecznego polecenia, uzyskaną dzięki wcześniejszemu zdefiniowaniu nazwanych zbiorów, które są częścią rozwiązania.

Pierwszy przykład: Na podstawie bazy danych ElektronicznyDziennikOcen wybrać tych uczniów, których średnia ocena z fizyki jest wyższa od ich średniej oceny z matematyki.

Przykład można rozwiązać za pomocą następującego polecenia;

```
WITH SrednieZFizyki AS
(
    SELECT Nazwisko, Imie, Pesel, Iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Fizyka'
    GROUP BY iducznia Nazwisko, Imie, Pesel,iducznia
),
SrednieZMatematyki AS
(
    SELECT Nazwisko, Imie, Pesel, iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Matematyka'
    GROUP BY Nazwisko, Imie, Pesel, iducznia
)
SELECT F.Nazwisko, F.Imie, F.Pesel
FROM SrednieZFizyki as F JOIN SrednieZMatematyki as M
    ON F.iducznia=M.iducznia
WHERE SredniaFizyka>SredniaMatematyka
```

Przy omawianiu tego przykładu należy zwrócić uwagę na fakt, że ostateczna postać polecenia jest prosta, natomiast wcześniej zdefiniowano nazwane zbiory, do których się odwołujemy.

Dla utrwalenia zasad stosowania wyrażeń CTE można formułować kolejne problemy, w rozwiązaniach których będzie można wykorzystać mechanizm wyrażeń CTE.



Drugi przykład: Na podstawie bazy danych ElektronicznyDziennikOcen wybrać uczniów, których średnia ocena z fizyki jest większa od 4.00 za wyjątkiem tych, których średnia ocena z matematyki jest niższa od 3.50

```
WITH SrednieZFizyki AS
(
    SELECT Nazwisko, Imie, Pesel, Iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Fizyka'
    GROUP BY iducznia Nazwisko, Imie, Pesel,iducznia
    HAVING AVG(ocena)>4.00
),
SrednieZMatematyki AS
(
    SELECT Nazwisko, Imie, Pesel, iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Matematyka'
    GROUP BY Nazwisko, Imie, Pesel, iducznia
    HAVING AVG(ocena) <3.50
)
SELECT Nazwisko, Imie, Pesel
FROM SrednieZFizyki
EXCEPT
SELECT Nazwisko, Imie, Pesel
FROM SrednieZMatematyki
```

W omawianym przykładzie dodatkowo należy zwrócić uwagę na wykorzystanie klauzuli HAVING do filtrowania zapytania w stosunku do wyniku działania funkcji agregującej oraz wykorzystanie operatora EXCEPT, który zwraca różnicę dwóch zbiorów (zbiory przygotowane zapytaniem). Zmieniając treść zadania można pokazać działanie operatora UNION (suma zbiorów) oraz INTERSECT (część wspólna zbiorów).

Lekcja 3. Zapytania rekurencyjne z wykorzystaniem wyrażeń CTE.

Czas: 90 min.

Dla odwzorowania hierarchii w relacyjnych bazach danych stosuje się klucze obce, które odwołują się do innego wiersza w tej samej tabeli. Przykład takiego rozwiązania jest pokazany na rysunku 2. W tabeli Pracownicy jest umieszczony klucz obcy o nazwie idprzełożonego, który wskazuje na inny wiersz w tej samej tabeli (przełożony jest też pracownikiem i ma swojego przełożonego). Takie konstrukcje tabel stwarzają problemy przy zapytaniach, których istotą jest wybranie danych powiązanych z konkretnych wierszem na wszystkich poziomach hierarchii (wszyscy podwładni konkretnej osoby na wszystkich poziomach podległości). Ponieważ problem przeglądania hierarchii jest w swej istocie problemem rekurencyjnym, to jego rozwiązanie wymaga również rekurencyjnego podejścia. Klasyczne polecenie SELECT języka SQL nie umożliwi realizacji zapytań rekurencyjnych. Wyrażenia

CTE mogą być użyte do zdefiniowania takiego zbioru nazwanego, który posłuży do rozwiązania zadań rekurencyjnych.

Przykład pierwszy: wygenerowanie tabeli z kolumną będącą sekwencją kolejnych liczb całkowitych w zakresie od 0 do 10.

Polecenie realizujące ten przykład może mieć następującą postać;

```
WITH Sekwencja AS
(
SELECT 0 AS liczba
UNION ALL
SELECT Liczba+1
FROM Sekwencja
WHERE liczba<10
)
SELECT Liczba FROM Sekwencja
```

Liczba
0
1
2
3
4
5
6
7
8
9
10

Rysunek 3. Wynik zapytania generującego sekwencję liczb

Na tym prostym przykładzie można omówić istotę wyrażania rekurencji w zapytaniach wykorzystujących wyrażenia CTE. Polecenie tworzące sekwencję liczb można opisać jako następujące czynności:

- definicja nazwanego zbioru Sekwencja(wyrażenie CTE) – WITH Sekwencja AS;
- zapytanie tworzące ten zbiór składa się z dwóch zapytań połączonych operatorem UNION ALL (dodaje do siebie zbiory bez eliminowania powtarzających się wartości):
 - pierwsze zapytanie (element wyjściowy) pobiera dane, od których rozpoczynamy rekurencję – SELECT 0 AS liczba – tworzy jeden wiersz w tabeli zawierającej jedną kolumnę z wartością 0 (zero);
 - drugie zapytanie – SELECT Liczba+1 FROM Sekwencja WHERE liczba<10 – (element rekurencyjny) pobiera dane ze zbioru Sekwencja zwiększając odczytaną kolumnę Liczba o jeden a operator UNION ALL dodaje ten wynik do wcześniej wybranych wierszy w zbiorze Sekwencja (w zbiorze Sekwencja znajdują się wiersze ostatnio wybrane);
 - po zdefiniowaniu wyrażenia CTE tworzymy zapytanie pobierające wyniki – SELECT Liczba FROM Sekwencja.

Na tym etapie należy poinformować uczniów, że SQL Server ma domyślne ograniczenie dla liczby wywołań rekurencyjnych wynoszące 100, ale można tę ilość zmienić używając opcji MAXRECURSION.

W naszym przykładzie, gdybyśmy chcieli wygenerować sekwencję 500 kolejnych liczb całkowitych – polecenie miałyby następującą postać:

```
WITH Sekwencja AS
(
SELECT 0 AS Liczba
UNION ALL
SELECT Liczba+1
FROM Sekwencja
WHERE Liczba<10
)
SELECT Liczba FROM Sekwencja
OPTION (MAXRECURSION 500)
```

Przykład drugi: wybranie wszystkich podwładnych (na dowolnym poziomie) konkretnego pracownika.

Odwołując się do tabel pokazanych na rysunku 2 możemy przystąpić do realizacji zadania. Dla lepszego zrozumienia przykładu przyjmijmy, że tabele Pracownicy i Funkcje mają zawartość jak na rysunku 4 i 5.

IdPracownika	Nazwisko	Imie	Pesel	IdPrzełożonego	IdFunkcji
1	Kot	Jan	87090812321	1	1
2	Pies	Janina	81110987654	1	2
3	Okoń	Stanisław	76120785609	1	2
4	Płotka	Zofia	59122188723	2	3
5	Leszcz	Piotr	76120976781	2	3
6	Szczupak	Maria	82061209834	4	4
7	Lin	Karol	87062187654	4	4
8	Myszka	Szara	44091187654	6	5
9	Szczurek	Paweł	69121189876	6	5
10	Królik	Violetta	90111298723	8	6

Rysunek 4. Przykładowa zawartość tabeli Pracownicy

idfunkcji	Nazwa
1	Prezes
2	Dyrektor Departamentu
3	Dyrektor Działu
4	Kierownik Zespołu
5	Referent
6	Asystent

Rysunek 5. Przykładowa zawartość tabeli Funkcje

Polecenie, które dla przykładowych danych zwróci dane wszystkich podwładnych (na wszystkich poziomach) pracownika o IdPracownika=1 może mieć następującą postać;

```
WITH Podwładni AS
(
SELECT Imie+' '+Nazwisko as Pracownik,
```

```

0 as Poziom,
idpracownika
FROM Pracownicy
WHERE IdPracownika=1
UNION ALL
SELECT Pr.Imie+''+Pr.Nazwisko as Pracownik ,
       Po.Poziom+1 as Poziom,
       Pr.idpracownika
FROM Pracownicy Pr JOIN Podwladni Po
ON Pr.IdPrzelozonego=Po.idpracownika
WHERE Pr.idPrzelozonego !=Pr.IdPracownika
)
SELECT * FROM Podwladni
    
```

Wynik tego zapytania dla przykładowych danych pokazanych na rysunkach 4 i 5 jest przedstawiony na rysunku 6.

Pracownik	Poziom	idpracownika
Jan Kot	0	1
Janina Pies	1	2
Stanisław Okoń	1	3
Zofia Płotka	2	4
Piotr Leszcz	2	5
Maria Szczupak	3	6
Karol Lin	3	7
Szara Myszka	4	8
Paweł Szczurek	4	9
Violetta Królik	5	10

Rysunek 6. Wynik zapytania

W omawianym zapytaniu utworzyliśmy kolumnę o nazwie Poziom, która określa „odległość” w hierarchii danego pracownika od tego, który jest początkowym elementem hierarchii.

Uczniom można zaproponować modyfikację tego zapytania, aby otrzymać wyniki pokazane na rysunku 7.

Pracownik	Funkcja	Poziom	idpracownika	ścieżka
Jan Kot	Prezes	0	1	
Janina Pies	Dyrektor Departamentu	1	2	Prezes-Jan Kot/
Stanisław Okoń	Dyrektor Departamentu	1	3	Prezes-Jan Kot/
Zofia Płotka	Dyrektor Działu	2	4	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies/
Piotr Leszcz	Dyrektor Działu	2	5	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies/
Maria Szczupak	Kierownik Zespołu	3	6	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka/
Karol Lin	Kierownik Zespołu	3	7	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka/
Szara Myszka	Referent	4	8	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak/
Paweł Szczurek	Referent	4	9	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak/
Violetta Królik	Asystent	5	10	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak./Referent-Szara Myszka/

Rysunek 7. Wyniki zapytania

W wynikach zapytania na rysunku 7 jest utworzona kolumna, która opisuje hierarchie podwładności dla każdego pracownika.

ZARZĄDZANIE SIECIAMI LAN

Dariusz Chaładyniak

Warszawska Wyższa Szkoła Informatyki

dchalad@wwsi.edu.pl

1. Zagadnienia programowe

Sieci komputerowe pojawiają się na lekcjach informatyki zarówno jako medium komunikacyjne, umożliwiające komunikację i dostęp do zasobów Internetu, jak i jako sieć zbudowana z fizycznych elementów i zarządzana odpowiednim oprogramowaniem.

Informatyka, IV etap edukacyjny – zakres podstawowy

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Bezpieczne posługiwanie się komputerem, jego oprogramowaniem i korzystanie z sieci komputerowej. Uczeń:
 - 3) korzysta z podstawowych usług w sieci komputerowej, lokalnej i rozległej, związanych z dostępem do informacji, wymianą informacji i komunikacją, przestrzega przy tym zasad netykiety i norm prawnych, dotyczących bezpiecznego korzystania i ochrony informacji oraz danych w komputerach w sieciach komputerowych.

Informatyka, IV etap edukacyjny – zakres rozszerzony

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:
 - 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;
3. Komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych. Uczeń:
 - 1) wykorzystuje zasoby i usługi sieci komputerowych w komunikacji z innymi użytkownikami, w tym do przesyłania i udostępniania danych;
 - 2) bierze udział w dyskusjach w sieci (forum internetowe, czat).

2. Temat zajęć

Zarządzanie sieciami LAN

3. Cele zajęć

Do prowadzenia jakiegokolwiek działalności związanej w wymianą informacji niezbędne jest prawidłowe funkcjonowanie sieci LAN (*Local Area Network*). Zrozumiane muszą być zasady projektowania, budowania i utrzymania architektury sieciowej. Celem zajęć jest przedstawienie sposobów budowy i utrzymania, hierarchicznej i w pełni konwergentnej sieci komputerowej.

4. Przygotowanie słuchaczy

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać podstawowe zasady posługiwania się komputerem,
- znać podstawowe pojęcia z zakresu informatyki i technologii sieciowych.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie rozważanego zagadnienia (problemu) przebiega w etapach, które składają się na skuteczną metodę uruchomienia lokalnej sieci komputerowej;
- posłużenie się przez nauczyciela i uczniów objaśnieniami i demonstracjami przykładowych procedur konfiguracyjnych;
- przygotowanie przez uczniów schematów działania sieci LAN;
- samodzielne sporządzenie przez uczniów funkcjonalnych modeli sieci;
- samodzielne testowanie poprawności zaimplementowanej konfiguracji;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas analizy przedprojektowej prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania koncepcji budowy sieci LAN;
- przygotowywanie modelowych projektów sieci może odbywać się indywidualnie lub w parach uczniów;
- programy do symulacji budowy sieci są instalowane i udostępnianie uczniom indywidualnie;
- każdy uczeń samodzielnie wykonuje aplikację konfiguracji w zaprojektowanym modelu;
- testowanie konfiguracji może odbywać się w grupach uczniów – uczniowie wspólnie poprawiają błędy w konfiguracji, dobierają parametry i porównują wyniki działania swoich programów;
- końcowym efektem pracy nad danym projektem jest prezentacja działającego modelu wraz z testami konfiguracji i komunikacji.



7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z pracowni komputerowej wyposażonej:

- komputery klasy PC z systemem operacyjnym Windows2000/XP Professional/Vista Business/Windows 7 Professional, Ultimate (uprawnienia administratora dla użytkowników);
- dostęp wszystkich komputerów do sieci LAN oraz sieci Internet;
- przeglądarka internetowa;
- zainstalowane oprogramowanie Packet Tracer (aktualna wersja);
- rzutnik multimedialny + ekran;
- tablica biała, markery (czarny, zielony, niebieski).

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie zajęć następujące opracowania:

1. Empson S., *Akademia sieci Cisco. CCNA Pełny przegląd poleceń*, WN PWN, Warszawa 2008,
2. Krysiak K., *Sieci komputerowe. Kompendium*, Helion, Gliwice 2005.
3. Lewis W., *Akademia sieci Cisco. CCNA Exploration. Przelączanie sieci LAN i sieci bezprzewodowe*, WN PWN, Warszawa 2009.
4. Lewis W., *CCNA semestr 3. Podstawy przelączania oraz routing pośredni*, WN PWN, Warszawa 2007.
5. Mucha M., *Sieci komputerowe. Budowa i działanie*, Helion, Gliwice 2003.
6. *Vademecum teleinformatyka*, IDG Poland SA, Warszawa 1999.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenie do problematyki związanej z zarządzaniem sieciami LAN powinny trwać kilka lekcjach z zachowaniem poniższej kolejności tematów, a każda lekcja powinna składać się z następujących etapów pracy (moduł 2x45min):

- A. Stworzenie opisu sytuacji problemowej, przekazanie założeń i wytycznych do projektu przez nauczyciela – ok. 15 min.
- B. Opracowanie i akceptacja koncepcji – ok. 15 min.
- C. Projektowanie i aplikacja konfiguracji – ok.45 min.
- D. Testowanie i weryfikacja poprawności implementacji – ok.10 min.
- E. Wskazanie zagadnień do przeanalizowania w ramach pracy domowej – ok. 5 min.

1. Zasady projektowania sieci LAN

Cechy dobrego projektu sieci LAN

- Funkcjonalność – sieć musi działać, to znaczy musi umożliwiać użytkownikom sprawne wykonywanie ich zadań. Sieć musi zapewniać połączenie użytkownik-użytkownik oraz użytkownik-aplikacja przy rozsądnej prędkości i niezawodności.
- Skalowalność – sieć musi mieć możliwość wzrostu, to znaczy musi rozwijać się zgodnie z pierwotnym planem, bez konieczności wprowadzania większych zmian do już istniejącego projektu.

- Adaptacyjność – sieć musi być projektowana z myślą o technologiach jutra i nie powinna zawierać żadnych elementów, które mogą w przyszłości ograniczać możliwość zastosowania najnowszych rozwiązań technicznych.
- Możliwość zarządzania – sieć powinna być zaprojektowana w sposób sprzyjający jej nadzorowi i utrzymaniu, co zapewni jej stałe i niezawodne działanie.

Podstawowe etapy planowania i instalacji sieci

- Etap wstępnego rozpoznania
- Etap projektowania
- Etap instalacji i wdrożenia
- Etap utrzymania i administracji

Kluczowe składniki projektu topologii sieci LAN mogą być podzielone na trzy odrębne kategorie, zgodnie z modelem odniesienia OSI:

- warstwę sieci,
- warstwę łącza danych,
- warstwę fizyczną.

Projektowanie topologii warstwy 1 (fizycznej)

- topologia gwiazdy,
- topologia rozszerzonej gwiazdy.

Projektowanie topologii warstwy 2 (łącza danych)

Cele stawiane przed projektem topologii warstwy 2 sieci lokalnej:

- aby ograniczyć rozmiar domen kolizji, należy zastosować urządzenia przełączające sieci LAN,
- w oparciu o grupy robocze użytkowników, należy utworzyć sieci VLAN i unikatowe domeny rozgłoszeniowe.

Projektowanie topologii warstwy 3 (sieci)

Cele stawiane przed projektem topologii warstwy 3 sieci lokalnej:

- tworzenie między segmentami sieci LAN, ścieżki filtrującej przepływ pakietów danych,
- odseparowanie transmisji rozgłoszeniowych protokołu ARP,
- odseparowanie kolizji między segmentami,
- wprowadzenie między segmentami filtrowania usług warstw wyższych.

2. Podstawy systemów operacyjnych stosowanych w aktywnych urządzeniach sieciowych, konfiguracja podstawowych parametrów

Model topologii

Interaktywny model zostanie stworzony indywidualnie przez uczniów z wykorzystaniem oprogramowania Packet Tracer (firmy Cisco Systems) według założeń przekazanych przez nauczyciela.

Konfiguracja podstawowych parametrów przełącznika – podłączenie terminala

Sekwencja startowa systemu operacyjnego przełącznika

Konfiguracja podstawowych parametrów przełącznika – podłączenie stacji zarządzającej



```

Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z
Switch(config)#line console 0
Switch(config-line)#password test
Switch(config-line)#login
Switch(config-line)#^Z
%SYS-5-CONFIG_I: Configured from console by console

```

Tabela adresacji

Urządzenie	Interfejs	Adres	Maska	Brama domyślna
Router1	Fa0/0	192.168.50.1	255.255.255.0	N/A
	Fa0/1.10	192.168.10.1	255.255.255.0	N/A
	Fa0/1.20	192.168.20.1	255.255.255.0	N/A
	Fa0/1.30	192.168.30.1	255.255.255.0	N/A
	Fa0/1.99	192.168.99.1	255.255.255.0	N/A
Switch1	VLAN 99	192.168.99.31	255.255.255.0	192.168.99.1
Switch2	VLAN 99	192.168.99.32	255.255.255.0	192.168.99.1
Switch3	VLAN 99	192.168.99.33	255.255.255.0	192.168.99.1
PC1	NIC	192.168.10.21	255.255.255.0	192.168.10.1
PC2	NIC	192.168.20.22	255.255.255.0	192.168.20.1
PC3	NIC	192.168.30.23	255.255.255.0	192.168.30.1
PC4	NIC	192.168.10.24	255.255.255.0	192.168.10.1
PC5	NIC	192.168.20.25	255.255.255.0	192.168.20.1
PC6	NIC	192.168.30.26	255.255.255.0	192.168.30.1

3. Konfiguracja i weryfikacja działania sieci VLAN

Schemat topologii

Schemat topologii tworzony jest indywidualnie według wskazówek nauczyciela.

Utworzenie VLAN-ów

```

Switch(config)#vlan 10
Switch(config-vlan)#name Student
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name Wykładowca
Switch(config-vlan)#vlan 30
Switch(config-vlan)#name Administracja
Switch(config-vlan)#vlan 99
Switch(config-vlan)#name Zarządzanie

```

Przydzielenie portów do VLAN-ów

```
Switch(config)#interface fastethernet 0/5
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#interface fastethernet 0/10
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#interface fastethernet 0/15
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 30
Switch(config-if)#^Z
```

4. Konfiguracja i weryfikacja działania protokołu VTP

Konfiguracja serwera VTP

```
Switch3(config)#vtp domain kurs1
Changing VTP domain name from NULL to kurs1
Switch3(config)#vtp version 1
VTP mode already in V1.
Konfiguracja klientów VTP:
Switch1(config)#vtp mode client
Setting device to VTP CLIENT mode
Switch2(config)#vtp mode client
Setting device to VTP CLIENT mode
```

5. Routing pomiędzy sieciami VLAN

Routing pomiędzy sieciami VLAN jest procesem przekazywania ruchu sieciowego z jednej sieci VLAN do innej z wykorzystaniem routera lub przełącznika warstwy 3.

Konfiguracja przełącznika

```
Switch1(config)#int f0/24
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#end
%SYS-5-CONFIG_I: Configured from console by console
%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to up
```



%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to up

Konfiguracja routera:

Router1(config)#interface f0/0.10 (kreowanie subinterfejsów)

Router1(config-subif)#encapsulation dot1Q 10

Router1(config-subif)#ip address 192.168.10.1 255.255.255.0

Router1(config-subif)#interface f0/0.20

Router1(config-subif)#encapsulation dot1Q 20

Router1(config-subif)#ip address 192.168.20.1 255.255.255.0

Router1(config-subif)#interface f0/0.30

Router1(config-subif)#encapsulation dot1Q 30

Router1(config-subif)#ip address 192.168.30.1 255.255.255.0

Router1(config-subif)#int f0/0

Router1(config-if)#no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/0.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.10, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.20, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/0.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.30, changed state to up



ZARZĄDZANIE SIECIAMI WAN

Dariusz Chaładyniak

Warszawska Wyższa Szkoła Informatyki

dchalad@wwsi.edu.pl

1. Zagadnienia programowe

Sieci komputerowe pojawiają się na lekcjach informatyki zarówno jako medium komunikacyjne, umożliwiające komunikację i dostęp do zasobów Internetu, jak i jako sieć zbudowana z fizycznych elementów i zarządzana odpowiednim oprogramowaniem.

Informatyka, IV etap edukacyjny – zakres podstawowy

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Bezpieczne posługiwanie się komputerem, jego oprogramowaniem i korzystanie z sieci komputerowej. Uczeń:
 - 3) korzysta z podstawowych usług w sieci komputerowej, lokalnej i rozległej, związanych z dostępem do informacji, wymianą informacji i komunikacją, przestrzega przy tym zasad netykiety i norm prawnych, dotyczących bezpiecznego korzystania i ochrony informacji oraz danych w komputerach w sieciach komputerowych.

Informatyka, IV etap edukacyjny – zakres rozszerzony

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:
 - 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;
3. Komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych. Uczeń:
 - 1) wykorzystuje zasoby i usługi sieci komputerowych w komunikacji z innymi użytkownikami, w tym do przesyłania i udostępniania danych;
 - 2) bierze udział w dyskusjach w sieci (forum internetowe, czat).



2. Temat zajęć

Zarządzanie sieciami WAN

3. Cele zajęć

Wraz ze wzrostem różnego typu działalności związanej z przesyłaniem informacji, zwiększaniem się ilości usług sieciowych, konieczna jest komunikacja pomiędzy sieciami odległymi od siebie i korzystającymi z różnych protokołów. Istnienie sieci WAN (*Wide Area Network*) stwarza dodatkowe obszary zainteresowań i odpowiedzialności, takie jak: bezpieczeństwo sieciowe, zarządzanie adresacją, sterowanie ruchem. Celem warsztatów jest wprowadzenie do technologii WAN wraz z praktyczną implementacją usług sieciowych.

4. Przygotowanie słuchaczy

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać podstawowe zasady posługiwania się komputerem,
- znać podstawowe pojęcia z zakresu informatyki i technologii sieciowych.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie rozważanego zagadnienia (problemu) przebiega w etapach, które składają się na skuteczną metodę uruchomienia lokalnej sieci komputerowej;
- posłużenie się przez nauczyciela i uczniów objaśnieniami i demonstracjami przykładowych procedur konfiguracyjnych;
- przygotowanie przez uczniów schematów działania sieci WAN;
- samodzielne sporządzenie przez uczniów funkcjonalnych modeli sieci;
- samodzielne testowanie poprawności zaimplementowanej konfiguracji;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas analizy przedprojektowej prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania koncepcji budowy sieci WAN;
- przygotowywanie modelowych projektów sieci może odbywać się indywidualnie lub w parach uczniów;
- programy do symulacji budowy sieci są instalowane i udostępnianie uczniom indywidualnie;
- każdy uczeń samodzielnie wykonuje aplikację konfiguracji w zaprojektowanym modelu;
- testowanie konfiguracji może odbywać się w grupach uczniów – uczniowie wspólnie poprawiają błędy w konfiguracji, dobierają parametry i porównują wyniki działania swoich programów;
- końcowym efektem pracy nad danym projektem jest prezentacja działającego modelu wraz z testami konfiguracji i komunikacji.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z pracowni komputerowej wyposażonej:

- komputery klasy PC z systemem operacyjnym Windows2000/XP Professional/Vista Business/Windows 7 Professional, Ultimate (uprawnienia administratora dla użytkowników),
- dostęp wszystkich komputerów do sieci LAN oraz sieci Internet,
- przeglądarka internetowa,
- zainstalowane oprogramowanie Packet Tracer (aktualna wersja),
- rzutnik multimedialny + ekran,
- tablica biała, markery (czarny, zielony, niebieski).

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie zajęć następujące opracowania:

1. Dye M. A., McDonald R., Rufi A. W., *Akademia sieci Cisco. CCNA Exploration. Semestr 1*, WN PWN, Warszawa 2008.
2. Graziani R., Vachon B., *Akademia sieci Cisco. CCNA Exploration. Semestr 4*, WN PWN, Warszawa 2009.
3. Krysiak K., *Sieci komputerowe. Kompedium*, Helion, Gliwice 2005.
4. Reid A., *CCNA semestr 4. Sieci rozległe – technologie WAN*, WN PWN, Warszawa 2007.
5. *Vademecum teleinformatyka*, IDG Poland SA, Warszawa 1999.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenie do problematyki związanej z zarządzaniem sieciami WAN powinny trwać kilka lekcjach z zachowaniem poniższej kolejności tematów, a każda lekcja powinna składać się z następujących etapów pracy (moduł 2x45min):

- A. Stworzenie opisu sytuacji problemowej, przekazanie założeń i wytycznych do projektu przez nauczyciela – ok. 15 min.
- B. Opracowanie i akceptacja koncepcji – ok. 15 min.
- C. Projektowanie i aplikacja konfiguracji – ok.45 min.
- D. Testowanie i weryfikacja poprawności implementacji – ok.10 min.
- E. Wskazanie zagadnień do przeanalizowania w ramach pracy domowej – ok. 5 min.

1. Projektowanie sieci WAN

Urządzenia sieci WAN

- modem,
- CSU/DSU (Channell Service Unit/Data Service Unit) linie cyfrowe,
- serwer dostępowy,
- przełącznik WAN,
- router.

Protokoły enkapsulacji w sieciach WAN

- HDLC,
- PPP,



- X.25,
- Frame Relay (FR),
- ATM.

2. Praktyczne aspekty implementacji protokołów WAN na urządzeniach sieciowych

Interaktywny model zostanie stworzony indywidualnie przez uczniów z wykorzystaniem oprogramowania Packet Tracer (firmy Cisco Systems) wg wskazówek nauczyciela.

3. Mobilne sieci WAN

Mobilne sieci WAN, nazywane także WWAN (ang. *Wireless Wide Area Network*). Jest to połączenie zasięgu sieci GSM oraz prędkości transferu z sieci WiFi, stwarzające możliwość mobilnego dostępu do transmisji danych (także Internetu).

Konfiguracja sieci WWAN w notebooku

1. Uruchamiamy narzędzie do zarządzania komunikacją i połączeniami
2. Tworzymy nowy profil
3. Wybieramy operatora
4. Edytujemy parametry profilu
5. Konfigurujemy parametry protokołu TCP/IP
6. Wykonujemy diagnostykę połączenia

Konfiguracja sieci WWAN w telefonie komórkowym (Windows Mobile Professional 6.1)

1. W panelu ustawień przechodzimy do zakładki połączenia
2. Wybieramy dodaj nowe połączenie modemowe
3. Tworzymy nowe połączenie wraz z nowym punktem dostępowym
4. Wpisujemy parametry konta
5. Konfigurujemy parametry protokołu IP oraz serwera DNS
6. Ustawiamy parametry dodatkowe połączenia
7. Finalizujemy tworzenie połączenia

4. Implementacja mechanizmów bezpieczeństwa

SSH (ang. *secure shell*) – standard protokołów komunikacyjnych używanych w sieciach komputerowych TCP/IP, w architekturze klient-serwer.

Konfiguracja protokołu SSH dla konsoli wirtualnego terminala routera:

```
INFORMATYKA+_01(config)#ip domain-name wwsi.pl
```

```
INFORMATYKA+_01(config)#crypto key generate rsa
```

```
The name for the keys will be: INFORMATYKA+_01.wwsi.pl
```

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

```
How many bits in the modulus [512]: 1024
```

```
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```

INFORMATYKA+_01(config)#username student password 12345678
INFORMATYKA+_01(config)#line vty 0 4
INFORMATYKA+_01(config-line)#transport input ssh
INFORMATYKA+_01(config-line)#login local
INFORMATYKA+_01(config-line)#end
%SYS-5-CONFIG_I: Configured from console by console
Konfiguracja protokołu PPP z autentykacją PAP (pomiędzy routerami R00 i R01)
INFORMATYKA+_00(config)#interface serial 0/0
INFORMATYKA+_00(config-if)#encapsulation ppp (włączenie enkapsulacji PPP)
INFORMATYKA+_00(config-if)#ppp authentication pap
INFORMATYKA+_00(config-if)# ppp sent-username l_00 password test
INFORMATYKA+_00(config-if)#end

INFORMATYKA+_01(config)#interface serial 0/0
INFORMATYKA+_01(config-if)#encapsulation ppp (włączenie enkapsulacji PPP)
INFORMATYKA+_01(config-if)#ppp authentication pap
INFORMATYKA+_01(config-if)# ppp sent-username l_01 password test
INFORMATYKA+_01(config-if)#end

```

Konfiguracja protokołu PPP z autentykacją CHAP (pomiędzy routerami R00 i R02)

CHAP (ang. *Challenge Handshake Authentication Protocol*) – to jeden z dwóch obok PAP sposobów uwierzytelniania w PPP. CHAP zapewnia węzłom zgłaszanie swojej tożsamości za pomocą trójfazowego uzgadniania. CHAP jest bezpiecznym protokołem uwierzytelniania, zapewnia ochronę przed atakami wykorzystującymi podsłuch transmisji, wykorzystuje MD5. Jest preferowany jako uwierzytelnianie w PPP

```

INFORMATYKA+_00(config)#username INFORMATYKA+_02 password 12345678
INFORMATYKA+_00(config)#interface serial 0/1
INFORMATYKA+_00(config-if)#encapsulation ppp (włączenie enkapsulacji PPP)
INFORMATYKA+_00(config-if)#ppp authentication chap
INFORMATYKA+_00(config-if)#end

INFORMATYKA+_02(config)#username INFORMATYKA+_00 password 12345678
INFORMATYKA+_02(config)#interface serial 0/1
INFORMATYKA+_02(config-if)#encapsulation ppp (włączenie enkapsulacji PPP)
INFORMATYKA+_02(config-if)#ppp authentication chap
INFORMATYKA+_02(config-if)#end

```



5. Konfiguracja i korzystanie z wybranych usług sieciowych

DNS (ang. *Domain Name System, system nazw domenowych*) – to system serwerów oraz protokół komunikacyjny zapewniający zamianę adresów znanych użytkownikom Internetu na adresy zrozumiałe dla urządzeń tworzących sieć komputerową np. IP).

DNS to także złożony system komputerowy oraz prawny. Zapewnia z jednej strony rejestrację nazw domen internetowych i ich powiązanie z numerami IP. Z drugiej strony realizuje bieżącą obsługę komputerów odnajdujących adresy IP odpowiadające poszczególnym nazwom.

Korzystanie z przeglądarki domen

Konfiguracja protokołu DNS w aktywnym urządzeniu sieciowym

DHCP (ang. *Dynamic Host Configuration Protocol*) – protokół dynamicznego konfigurowania węzłów – protokół komunikacyjny umożliwiający komputerom uzyskanie od serwera danych konfiguracyjnych, np. adresu IP hosta, adresu IP bramy sieciowej, adresu serwera DNS, maski podsieci.

Konfiguracja protokołu DHCP w aktywnym urządzeniu sieciowym

NAT (ang. *Network Address Translation*), tłumaczenie adresów sieciowych; znane również jako maskowanie sieci lub adresu IP – technika przesyłania ruchu sieciowego poprzez router, która wiąże się ze zmianą źródłowych lub docelowych adresów IP, zwykle również numerów portów TCP/UDP pakietów IP podczas ich przepływu. Zmieniane są także sumy kontrolne (tak IP jak i TCP/UDP), aby potwierdzić wprowadzone zmiany.

Większość systemów korzystających z NAT ma na celu umożliwienie dostępu wielu hostom w sieci prywatnej do Internetu przy wykorzystaniu pojedynczego publicznego adresu IP (brama sieciowa).



KONFIGURACJA PROTOKOŁÓW ROUTINGU STATYCZNEGO I DYNAMICZNEGO

Dariusz Chaładyniak

Warszawska Wyższa Szkoła Informatyki

dchalad@wwsi.edu.pl

1. Zagadnienia programowe

Sieci komputerowe pojawiają się na lekcjach informatyki zarówno jako medium komunikacyjne, umożliwiające komunikację i dostęp do zasobów Internetu, jak i jako sieć zbudowana z fizycznych elementów i zarządzana odpowiednim oprogramowaniem.

Informatyka, IV etap edukacyjny – zakres podstawowy

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Bezpieczne posługiwanie się komputerem, jego oprogramowaniem i korzystanie z sieci komputerowej. Uczeń:
 - 3) korzysta z podstawowych usług w sieci komputerowej, lokalnej i rozległej, związanych z dostępem do informacji, wymianą informacji i komunikacją, przestrzega przy tym zasad netykiety i norm prawnych, dotyczących bezpiecznego korzystania i ochrony informacji oraz danych w komputerach w sieciach komputerowych.

Informatyka, IV etap edukacyjny – zakres rozszerzony

W rozdziale pierwszym oczekuje się od ucznia umiejętności bezpiecznego posługiwania się komputerem i jego oprogramowaniem, wykorzystania sieci komputerowej; komunikowania się za pomocą komputera i technologii informacyjno-komunikacyjnych

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:
 - 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;
3. Komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych. Uczeń:
 - 1) wykorzystuje zasoby i usługi sieci komputerowych w komunikacji z innymi użytkownikami, w tym do przesyłania i udostępniania danych;
 - 2) bierze udział w dyskusjach w sieci (forum internetowe, czat).



2. Temat zajęć

Konfiguracja protokołów routingu statycznego i dynamicznego

3. Cele zajęć

Dynamika zmian o sieciach komputerowych wymusza stosowanie w pełni skalowalnych i wysoce wydajnych protokołów routingu, służących do wymiany informacji pomiędzy urządzeniami oraz do określenia optymalnej ścieżki do sieci docelowej. Zajęcia są poświęcone prezentacji różnych rodzajów protokołów wraz z praktycznymi sposobami ich implementacji i weryfikacji działania.

4. Przygotowanie słuchaczy

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać podstawowe zasady posługiwania się komputerem,
- znać podstawowe pojęcia z zakresu informatyki i technologii sieciowych,
- orientować się w problematyce związanej z działaniem sieci LAN i WAN.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie rozważanego zagadnienia (problemu) przebiega w etapach, które składają się na skuteczną metodę konfiguracji i uruchomienia protokołów routingu statycznego i dynamicznego;
- posłużenie się przez nauczyciela i uczniów objaśnieniami i demonstracjami przykładowych procedur konfiguracyjnych;
- przygotowanie przez uczniów schematów działania protokołów routingu statycznego i dynamicznego;
- samodzielne sporządzenie przez uczniów funkcjonalnych modeli sieci;
- samodzielne testowanie poprawności zaimplementowanej konfiguracji;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas analizy przedprojektowej prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania koncepcji działania protokołów routingu statycznego i dynamicznego;
- przygotowywanie modelowych projektów sieci może odbywać się indywidualnie lub w parach uczniów;
- programy do symulacji budowy sieci są instalowane i udostępniane uczniom indywidualnie;
- każdy uczeń samodzielnie wykonuje aplikację konfiguracji w zaprojektowanym modelu;



- testowanie konfiguracji może odbywać się w grupach uczniów – uczniowie wspólnie poprawiają błędy w konfiguracji, dobierają parametry i porównują wyniki działania swoich programów;
- końcowym efektem pracy nad danym projektem jest prezentacja działającego modelu wraz z testami konfiguracji i komunikacji.

7. Materiały pomocnicze

Nauczyciel i uczniowie korzystają z pracowni komputerowej wyposażonej:

- komputery klasy PC z systemem operacyjnym Windows2000/XP Professional/Vista Business/Windows 7 Professional, Ultimate (uprawnienia administratora dla użytkowników),
- dostęp wszystkich komputerów do sieci LAN oraz sieci Internet,
- przeglądarka internetowa,
- zainstalowane oprogramowanie Packet Tracer (aktualna wersja),
- rzutnik multimedialny + ekran,
- tablica biała, markery (czarny, zielony, niebieski).

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie zajęć następujące opracowania:

1. Empson S., *Akademia sieci Cisco. CCNA Pełny przegląd poleceń*, WN PWN, Warszawa 2008,
2. Graziani R., Johnson A., *Akademia sieci Cisco. CCNA Exploration. Semestr 2. Protokoły i koncepcje routingu*, WN PWN, Warszawa 2008.
3. Józefiak A., *Budowa sieci komputerowych na przełącznikach i routerach Cisco*, Helion, Gliwice 2009.
4. Krysiak K., *Sieci komputerowe. Kompendium*, Helion, Gliwice 2005.
5. Mucha M., *Sieci komputerowe. Budowa i działanie*, Helion, Gliwice 2003.
6. Odom W., McDonald R., *CCNA semestr 2. Routery i podstawy routingu*, WN PWN, Warszawa 2007.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest wprowadzenie do problematyki związanej z konfiguracją protokołów routingu statycznego i dynamicznego, powinny trwać kilka lekcjach z zachowaniem poniższej kolejności tematów, a każda lekcja powinna składać się z następujących etapów pracy (moduł 2x45min):

- A. Stworzenie opisu sytuacji problemowej, przekazanie założeń i wytycznych do projektu przez nauczyciela – ok. 15 min.
- B. Opracowanie i akceptacja koncepcji – ok. 15 min.
- C. Projektowanie i aplikacja konfiguracji – ok.45 min.
- D. Testowanie i weryfikacja poprawności implementacji – ok.10 min.
- E. Wskazanie zagadnień do przeanalizowania w ramach pracy domowej – ok. 5 min.

1. Różne sposoby adresowania w sieciach komputerowych

Działania na przestrzeni adresowej IPv4

Projektowanie sieci o określonej liczbie hostów

Założenia:

- przestrzeń adresowa 172.16.0.0/16,
- sieć LAN1 – 400 hostów,
- sieć LAN2 – 200 hostów,
- sieć LAN3 – 100 hostów,
- sieć LAN4 – 100 hostów,
- sieć LAN5 – 60 hostów,
- sieć LAN6 – 20 hostów,
- sieć WAN1, 2, 3 – są to sieci *point-to-point*.

Zadanie do wykonania: Zaprojektować schemat adresacji zaczynając od sieci największej a kończąc na najmniejszej. Przy tym należy zachować zasadę, że powinniśmy zachować jak najwięcej adresów na przyszły rozwój sieci. Określić adresy sieci, maski oraz zakresy dla adresów użytecznych

Projektowanie wymaganej ilości sieci przy opisanej przestrzeni adresowej

Założenia:

- przestrzeń adresowa 192.168.1.0/24,
- 5 maksymalnie dużych sieci LAN,
- 4 *point-to-point* sieci WAN.

Zadanie do wykonania: Zaprojektować schemat adresacji zgodnie z wymaganiami. Określić adresy sieci, maski oraz zakresy dla adresów użytecznych

Działania na przestrzeni adresowej IPv6

Budowanie statycznej tablicy routingu

Model topologii

Interaktywny model zostanie stworzony indywidualnie przez uczniów z wykorzystaniem oprogramowania Packet Tracer (firmy Cisco Systems).

Schemat adresacji

LAN1: 192.168.1.0/24

LAN2: 172.16.0.0/16

WAN1: 10.10.10.0/30

WAN2: 10.10.10.4/30

2. Konfiguracja routingu statycznego

```
INFORMATYKA+_01(config)#ip route 172.16.0.0 255.255.0.0 10.10.10.2
```

```
INFORMATYKA+_01(config)#ip route 10.10.10.4 255.255.255.252 10.10.10.2
```

```
INFORMATYKA+_00(config)#ip route 192.168.1.0 255.255.255.0 10.10.10.1
```

```
INFORMATYKA+_00(config)#ip route 172.16.0.0 255.255.0.0 10.10.10.6
```

```
INFORMATYKA+_02(config)#ip route 192.168.1.0 255.255.255.0 10.10.10.5
```

```
INFORMATYKA+_02(config)#ip route 10.10.10.0 255.255.255.252 10.10.10.5
```

3. Konfiguracja i weryfikacja działania protokołu routingu dynamicznego RIP

Schemat topologii i adresacji

LAN1: 192.168.1.0/24

LAN2: 172.16.0.0/16

WAN1: 10.10.10.0/30

WAN2: 10.10.10.4/30

Konfiguracja protokołu RIP

```
INFORMATYKA+_01(config)#router rip
INFORMATYKA+_01(config-router)#version 2
INFORMATYKA+_01(config-router)#network 192.168.1.0
INFORMATYKA+_01(config-router)#network 10.10.10.0
```

```
INFORMATYKA+_00(config)#router rip
INFORMATYKA+_00(config-router)#version 2
INFORMATYKA+_00(config-router)#network 10.10.10.0
```

```
INFORMATYKA+_02(config)#router rip
INFORMATYKA+_02(config-router)#version 2
INFORMATYKA+_02(config-router)#network 172.16.0.0
INFORMATYKA+_02(config-router)#network 10.10.10.0
```

Weryfikacja tablicy routingu

Weryfikacja działania protokołu routingu

4. Konfiguracja i weryfikacja działania protokołu routingu dynamicznego EIGRP

Schemat topologii i adresacji

LAN1: 192.168.1.0/24

LAN2: 172.16.0.0/16

WAN1: 10.10.10.0/30

WAN2: 10.10.10.4/30

Konfiguracja protokołu EIGRP

```
INFORMATYKA+_01(config)#router EIGRP 100
INFORMATYKA+_01(config-router)#no auto-summary
INFORMATYKA+_01(config-router)#network 192.168.1.0 0.0.0.255
INFORMATYKA+_01(config-router)#network 10.10.10.0 0.0.0.3
```

```
INFORMATYKA+_02(config)#router EIGRP 100
INFORMATYKA+_02(config-router)#no auto-summary
INFORMATYKA+_02(config-router)#network 172.16.1.0 0.0.255.255
INFORMATYKA+_02(config-router)#network 10.10.10.4 0.0.0.3
```




```
INFORMATYKA+_00(config)#router EIGRP 100
INFORMATYKA+_00(config-router)#no auto-summary
INFORMATYKA+_00(config-router)#network 10.10.10.0 0.0.0.3
INFORMATYKA+_02(config-router)#network 10.10.10.4 0.0.0.3
```

Weryfikacja działania protokołu EIGRP

5. Konfiguracja i weryfikacja działania protokołu routingu dynamicznego OSPF

Schemat topologii i adresacji

LAN1: 192.168.1.0/24
LAN2: 172.16.0.0/16
WAN1: 10.10.10.0/30
WAN2: 10.10.10.4/30

Konfiguracja protokołu OSPF

```
INFORMATYKA+_01(config)#router ospf 200
INFORMATYKA+_01(config-router)#network 10.10.10.0 0.0.0.3 area 0
INFORMATYKA+_01(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

```
INFORMATYKA+_02(config)#router ospf 200
INFORMATYKA+_02(config-router)#network 172.16.0.0 0.0.255.255 area 0
INFORMATYKA+_02(config-router)#network 10.10.10.4 0.0.0.3 area 0
```

```
INFORMATYKA+_00(config)#router ospf 200
INFORMATYKA+_00(config-router)#network 10.10.10.0 0.0.0.3 area 0
INFORMATYKA+_00(config-router)#network 10.10.10.4 0.0.0.3 area 0
```

Weryfikacja działania protokołu OSPF



OBRÓBKA I WIZUALIZACJA OBRAZÓW

Piotr Kopciał

Politechnika Warszawska

piotrkopcial@gmail.com

1. Zagadnienia programowe

Edycja i tworzenie obrazów cyfrowych pojawia się na lekcjach informatyki i plastyki.

Informatyka, III etap edukacyjny

W czwartym dziale od ucznia oczekuje się umiejętności opracowywania za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.

4. Opracowywanie, za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. Uczeń:
 - 1) przy użyciu edytora grafiki tworzy kompozycje z figur, fragmentów rysunków i zdjęć, umieszcza napisy na rysunkach, tworzy animacje, przekształca formaty plików graficznych;
 - 6) tworzy dokumenty, zawierające różne obiekty (na przykład tekst, grafikę, tabele, wykresy i tym podobne) pobrane z różnych programów i źródeł;
 - 7) tworzy i przedstawia prezentację z wykorzystaniem różnych elementów multimedialnych, graficznych, tekstowych, filmowych i dźwiękowych własnych lub pobranych z innych źródeł.



Informatyka, IV etap edukacyjny, zakres podstawowy

W pierwszym dziale od ucznia oczekuje się znajomości współpracy aparatu cyfrowego z komputerem. W czwartym dziale od ucznia oczekuje się umiejętności opracowywania za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.

1. Bezpieczne posługiwanie się komputerem, jego oprogramowaniem i korzystanie z sieci komputerowej. Uczeń:
 - 1) opisuje podstawowe elementy komputera, jego urządzenia zewnętrzne i towarzyszące (np. aparat cyfrowy) i ich działanie w zależności od wartości ich podstawowych parametrów, wyjaśnia współdziałanie tych elementów;
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 1) edytuje obrazy w grafice rastrowej i wektorowej, dostrzega i wykorzystuje różnice między tymi typami obrazów;
 - 2) przekształca pliki graficzne z uwzględnieniem wielkości plików i ewentualnej utraty jakości obrazów;
 - 3) opracowuje obrazy i filmy pochodzące z różnych źródeł, tworzy albumy zdjęć;

Informatyka, IV etap edukacyjny, zakres rozszerzony

W pierwszym dziale od ucznia oczekuje się znajomości sposobów reprezentowania obrazów i animacji w komputerze. W czwartym dziale od ucznia oczekuje się umiejętności opracowywania za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych, a w szóstym – umiejętności współpracy w ramach projektów.

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej.

Uczeń:

 - 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;
 - 4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania.
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 1) opisuje podstawowe modele barw i ich zastosowanie;
 - 2) określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;
 - 3) przetwarza obrazy i filmy, np.: zmienia rozdzielczość, rozmiar, model barw, stosuje filtry;
6. Uczeń wykorzystuje komputer oraz programy i gry edukacyjne do poszerzania wiedzy i umiejętności z różnych dziedzin:
 - 1) opracowuje indywidualne i zespołowe projekty przedmiotowe i międzyprzedmiotowe z wykorzystaniem metod i narzędzi informatyki;

Plastyka, III etap edukacyjny

W drugim dziale od ucznia oczekuje się umiejętności tworzenia wypowiedzi – ekspresji przez sztukę. Kurs obejmuje m.in. tworzenie kartki pocztowej, kalendarza ściennego, fotoksiążki, animacji oraz filmu.

2. Tworzenie wypowiedzi – ekspresja przez sztukę. Uczeń:
 - 1) podejmuje działalność twórczą, posługując się środkami wyrazu sztuk plastycznych, innych dziedzin sztuki (fotografia, film) i elementami formy przekazów medialnych, w kompozycji na płaszczyźnie oraz w przestrzeni rzeczywistej i wirtualnej (stosując określone materiały, narzędzia i techniki właściwe dla tych dziedzin sztuki i przekazów medialnych);
 - 2) realizuje projekty w zakresie sztuk wizualnych, w tym służące przekazywaniu informacji dostosowanej do sytuacji komunikacyjnej oraz uczestnictwu w kulturze społeczności szkolnej i lokalnej (stosując także narzędzia i wytwory mediów środowiska cyfrowego).

2. Temat zajęć

Obróbka i wizualizacja obrazów

Aparat cyfrowy stał się przedmiotem powszechnego użytku. Dzięki temu amatorzy fotografii mogą uwiecznić wiele sytuacji z życia. Zdarza się jednak, że niektóre zdjęcia wymagają korekty. Zajęcia są poświęcone metodom edycji i retuszu zdjęć oraz ich efektownej prezentacji. Pierwsza część zajęć jest krótkim przedstawieniem parametrów obrazu cyfrowego.

Uczniowie poznają pracę z obrazem przy zastosowaniu m.in. warstw i masek. Następnie ćwiczą techniki poprawy zdjęć: poprawianie źle naświetlonego zdjęcia, usuwanie efektu czerwonych oczu, poprawianie kolorów i ostrości, poprawianie zdjęć z tanich aparatów fotograficznych (takich, jak w komórkach), przekształcanie obrazu, zabawy ze światłem i kolorem oraz poznają narzędzia profesjonalnego fotoretuszu. W dalszej części uczniowie zapoznają się z zastosowaniem filtrów graficznych oraz zaawansowanych narzędzi edycji tekstu i tworzenia animacji w programie GIMP.

W dzisiejszych czasach duże znaczenie odgrywa umiejętność zastosowania grafiki w różnych projektach. Efektem końcowym zajęć każdego ucznia będzie własna fotoksiążka, kalendarz ścienny oraz kartka pocztowa, zawierające samodzielnie edytowane obrazy. Uczniowie nauczą się również tworzenia filmu zawierającego zdjęcia oraz nagrania audio i video.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- wykazać się zrozumieniem pojęcia obrazu wektorowego i rastrowego (wskazać różnice pomiędzy nimi), rozdzielczości obrazu;
- scharakteryzować przykładowe formaty graficzne; wykorzystać program graficzny GIMP do:
 - edycji parametrów obrazu, takich jak jasność, kontrast, nasycenie,
 - geometrycznego przekształcania obrazu,
 - pracy na warstwach,
 - edycji obrazu przy użyciu filtrów graficznych;
- zaprojektować i wykonać fotomontaż, kartkę pocztową, kalendarz, fotoksiążkę;
- utworzyć animację;
- zmontować prosty film;
- wykorzystać internetowe narzędzia do edycji i publikowania grafiki.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów (patrz następny punkt Metody pracy);
- znać przynajmniej jeden program komputerowy do edycji grafiki;
- znać podstawy fotografowania.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (projekt),
 - implementacja (realizacja) rozwiązania w postaci obrazu cyfrowego,



- ewaluacja rozwiązania,
- prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- posłużenie się przez nauczyciela i uczniów objaśnieniami przykładowych sposobów wykonywania projektów graficznych za pomocą listy kroków;
- samodzielne sporządzenie przez uczniów projektów graficznych;
- zapisywanie elementów składowych (poszczególne obrazy) oraz etapów przejściowych projektu na komputerze;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania pomysłu na temat projektu graficznego (np. kalendarz szkolny, zaproszenie na konkretne wydarzenie);
- gromadzenie elementów składowych projektu (poszczególnych obrazów) może odbywać się indywidualnie lub w parach uczniów;
- projekty graficzne uczniowie tworzą samodzielnie;
- końcowym efektem pracy jest prezentacja gotowego projektu graficznego.



7. Materiały pomocnicze

Nauczyciel może korzystać z tablicy w celu zapisania kolejnych kroków przekształcania obrazu, lub poszczególnych etapów wykonywania projektu graficznego. Dla uczniów przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami.

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- podręcznik dla szkół ponadgimnazjalnych, np.: W. Wrotek, *Technologia informacyjna*, Helion, Gliwice 2006;
- publikacje uzupełniające, np.:
 - Oberlan Ł., *GIMP Domowe studio graficzne*, Helion, Gliwice 2003,
 - Owczarz-Dadan A., *ABC fotografii cyfrowej i obróbki zdjęć*, Helion, Gliwice 2006,
- materiały dotyczące obróbki grafiki i filmów, opracowane w projekcie Informatyka +;
- filmy instruktażowe prezentujące proces wykonywania projektu graficznego – wiele takich filmów jest dostępnych w sieci Internet;
- komputer i jego podstawowe oprogramowanie, oraz program graficzny GIMP i inne dostępne bezpłatnie programy do edycji i tworzenia grafiki, animacji i filmów.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest nabycie umiejętności edycji obrazów cyfrowych oraz tworzenia projektów graficznych, animacji i filmów, mogą być rozłożone na kilka lekcji, przy czym zaleca się realizację według następującego toku:

- w części wprowadzającej, nauczyciel objaśnia potrzebę komputerowej edycji obrazów cyfrowych oraz omawia zagadnienia, takie jak: obraz wektorowy, obraz rastrowy, rozdziel-

czość obrazu, formaty graficzne. W tej części prezentowane są także praktyczne wskazówki dotyczące fotografowania;

- następnie uwaga uczniów zostaje skupiona na możliwościach, narzędziach i korzystaniu z programu graficznego GIMP;
- na kolejnym etapie uczniowie poznają podstawowe sposoby poprawy jakości zdjęć i obrazów cyfrowych;
- kolejnymi etapami, które mogą następować w pewnych odstępach czasu, jest wprowadzanie projektów graficznych (np. kartka pocztowa, kalendarz), wymagających użycia wielu narzędzi programu GIMP lub programów dedykowanych;
- ostatnim zadaniem jest tworzenie animacji i filmów.

Lekcja 1. Wprowadzenie: obrazy wektorowe i rastrowe, rozdzielczość obrazu, formaty graficzne. Czas: 90 min.

Tytułem wprowadzenia prezentujemy przydatność umiejętności edycji obrazów cyfrowych w życiu codziennym. Najczęściej edytujemy zdjęcia, aby usunąć z nich niedoskonałości. Jednakże powody mogą być też inne:

1. Zdjęcie wykonano w niesprzyjających warunkach (zbyt ciemno, zbyt jasno itp.).
2. Zdjęcie wykonane tanim aparatem cyfrowym jest kiepskiej jakości.
3. Przygotowujemy zdjęcie do publikacji w sieci WWW.
4. Przygotowujemy zdjęcie do przesłania przez Internet.
5. Przekształcamy zdjęcia, by uzyskać różne efekty graficzne (np. kolorystyka sepia).

Animacje pojawiają się dość często na stronach internetowych. Ruchomy obraz bardziej przyciąga uwagę, aniżeli obraz statyczny. Mamy do czynienia z animowanym logo, przyciskami, a także reklamami. Współczesne aparaty cyfrowe często są wyposażone w funkcję nagrywania obrazu video wraz z towarzyszącym mu dźwiękiem. Krótkie, kilkuminutowe filmy, ilustrujące ważne momenty np. z odbytej podróży, stały się popularne.

Można poprosić uczniów, by postarali się wskazać, jakie operacje na zdjęciach cyfrowych wykonywali dotychczas. Nie powinni mieć z tym specjalnego kłopotu.

Po tego typu wprowadzeniu omawiamy zagadnienia, takie jak: obrazy wektorowe i rastrowe, rozdzielczość obrazu, formaty graficzne.

Lekcja 2. Program graficzny GIMP: podstawowe narzędzia. Czas: 60 min.

Celem tej lekcji jest przybliżenie uczniom możliwości i narzędzi programu graficznego GIMP. Można w tym celu zaprezentować film instruktażowy prezentujący użycie programu GIMP w konkretnym projekcie graficznym (z racji, że program GIMP jest darmowy, wiele tego typu filmów znajdziemy w serwisie YouTube).

Uczniowie pobierają program GIMP z Internetu i instalują na swoich komputerach. Po zainstalowaniu zapoznają się z menu programu oraz zawartością przybornika.

Następnie uczniowie wykorzystują narzędzia programu GIMP, rozwiązując problemy, takie jak:

- poprawianie źle naświetlonego zdjęcia,
- usuwanie efektu czerwonych oczu,



- poprawianie kolorów,
- poprawianie ostrości,
- poprawianie zdjęć z tanich aparatów fotograficznych,
- skalowanie,
- obracanie,
- kadrowanie,
- przekształcenia geometryczne.

Lekcja 3. Program graficzny GIMP: narzędzia zaawansowane: klonowanie, maski, warstwy, filtry, tekst. Czas: 90 min.

W tej części uczniowie wykorzystują zaawansowane narzędzia programu GIMP (takie jak: klonowanie, maski, warstwy, filtry), wsparte narzędziami podstawowymi.

Uczniowie wykonują ćwiczenia opisane w zeszyte dydaktycznym, takie jak:

- usuwanie niechcianych elementów ze zdjęcia,
- fotomontaż,
- poprawianie zdjęcia portretowego,
- tworzenie małej głębi ostrości,
- tworzenie wrażenia ruchu,
- tworzenie niestandardowych efektów graficznych.

Na zakończenie uczniowie tworzą napisy, a następnie edytują je, wykorzystując wszystkie dostępne narzędzia programu GIMP

Lekcja 4. Projekty graficzne: kartka pocztowa, kalendarz ścienny, fotoksiążka. Czas: 90 min.

W tej części uczniowie realizują projekty graficzne, takie jak: kartka pocztowa, kalendarz ścienny, fotoksiążka. W tym celu mogą posłużyć się programami dedykowanymi do tego celu, dostępnymi w Internecie. Uczniowie znajdują potrzebny program w Internecie lub wyszukują program wskazany w zeszyte dydaktycznym. Następnie pobierają go i instalują na swoim komputerze.

Natomiast zdjęcia wchodzące w skład realizowanych projektów uczniowie edytują, wykorzystując poznane narzędzia programu GIMP, oraz sieciowe (internetowe) edytory grafiki.

Po wykonaniu projektu graficznego, każdy z uczniów publicznie prezentuje swoją pracę.

Lekcja 5. Tworzenie animacji, montaż filmu. Czas: 90 min.

Tworzenie filmu odbywa się według następujących kroków:

1. pobieranie materiału video, audio oraz grafiki na twardy dysk komputera,
2. dodawanie materiału do projektu w programie do tworzenia filmów,
3. przycinanie plików video,
4. dodawanie podkładu dźwiękowego,
5. dołączanie grafiki (zdjęć),
6. dodawanie napisów,
7. użycie efektów specjalnych: przejścia, nakładanie scen itp.,

8. zapis gotowego filmu do pliku.

Jeśli chcemy przygotować film DVD, należy wykonać dodatkowe kroki:

9. tworzenie menu,
10. nagranie filmu na płytę DVD.

Na zajęciach uczniowie korzystają z plików graficznych, audio i video, udostępnionych na platformie edukacyjnej Informatyka+, mogą także korzystać z własnych zdjęć i muzyki, a potrzebny materiał filmowy zrealizować za pomocą własnych aparatów cyfrowych.

Zarówno przy tworzeniu filmów jak i animacji, uczniowie postępują zgodnie z poleceniami opisanymi w ćwiczeniach w zeszycie dydaktycznym.

Po ukończeniu animacji, a następnie filmu, każdy z uczniów publicznie prezentuje swoją pracę.



TWORZENIE SERWISÓW INTERNETOWYCH

Piotr Kopciał

Politechnika Warszawska

piotrkopcial@gmial.com

1. Zagadnienia programowe

Tworzenie stron internetowych oraz baz danych pojawia się na lekcjach informatyki.

Informatyka, III etap edukacyjny

W czwartym dziale od ucznia oczekuje się umiejętności tworzenia prostej strony internetowej.

4. Opracowywanie, za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. Uczeń:
 - 8) tworzy prostą stronę internetową, zawierającą tekst, grafikę, elementy aktywne, linki, korzystając ewentualnie z odpowiedniego edytora stron, wyjaśnia znaczenie podstawowych poleceń języka HTML.

Informatyka, IV etap edukacyjny, zakres podstawowy

W czwartym dziale od ucznia oczekuje się umiejętności tworzenia strony internetowej oraz umiejętności tworzenia i znajomości obsługi bazy danych.

4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - 6) tworzy bazę danych, posługuje się formularzami, porządkuje dane, wyszukuje informacje stosując filtrowanie;
 - 7) wykonuje podstawowe operacje modyfikowania, i wyszukiwania informacji na relacyjnej bazie danych;
 - 9) projektuje i tworzy stronę internetową, posługując się stylami, szablonami i elementami programowania.

Informatyka, IV etap edukacyjny, zakres rozszerzony

W pierwszym dziale od ucznia oczekuje się znajomości architektury klient-serwer oraz umiejętności korzystania z wybranych usług sieciowych. Wiedza ta będzie potrzebna do zainstalowania i skonfigurowania serwera WWW. W drugim dziale od ucznia oczekuje się umiejętności tworzenia i znajomości obsługi bazy danych. Uczniowie zainstalują i skonfigurują system obsługi baz danych i będą tworzyć własne bazy danych.

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:

- 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;
 - 4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania.
2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
- 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
 - 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
 - 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

2. Temat zajęć

Tworzenie serwisów internetowych

Internet wkracza w coraz to nowe obszary naszego życia: e-nauczanie, elektroniczne biblioteki, wirtualne laboratoria, medycyna, usługi (bankowość, turystyka). Podstawowym elementem tych i podobnych serwisów są strony internetowe, które składają się na bardziej złożone witryny, portale i platformy internetowe.

Zajęcia są poświęcone tworzeniu i funkcjonowaniu serwisów internetowych. W pierwszej części opisano mechanizmy działania stron internetowych, w tym m.in. komunikację w standardzie klient-serwer i strony dynamiczne. Omawiane są zalety i wady stron statycznych i dynamicznych oraz mechanizmy interakcji serwisów internetowych z użytkownikiem, stosowane we współczesnych stronach internetowych. Następnie przedstawiono kilka przykładowych interaktywnych serwisów WWW w działaniu. Jednym z celów tej prezentacji jest uwrażliwienie uczniów na dobre praktyki projektowania i tworzenia stron internetowych. W części warsztatowej uczniowie poznają fundamentalny metajęzyk opisu struktury strony – HTML oraz ściśle z nim związany język CSS, odpowiedzialny za styl prezentacji treści na stronie. Przy tym uczą się korzystania z programu (edytora) do tworzenia stron internetowych.

Tematyka kursu obejmuje również metody i techniki tworzenia dynamicznych stron internetowych. Uczniowie poznają podstawy języka PHP oraz SQL, a także sposób ich współdziałania w praktycznych przykładach. Zajęcia obejmują również instalowanie i konfigurowanie serwera WWW Apache oraz relacyjnej bazy danych MySQL, a także korzystanie ze środowiska programistycznego, służącego do tworzenia dynamicznych stron internetowych. Uzupełnieniem tych treści jest metodyka tworzenia złożonych projektów informatycznych. Zaprezentowano przykład w pełni funkcjonalnego serwisu.

Wiedzę zdobytą w trakcie prezentacji materiału i ćwiczeń, uczniowie weryfikują projektując własny dynamiczny serwis sieciowy w technologii klient-serwer. Serwis ten poza stroną wizualną, zawiera takie funkcje jak: obsługa interaktywnych formularzy użytkownika, przetwarzanie danych wprowadzonych przez użytkownika, obsługa logowania użytkownika, obsługa sesji użytkownika i inne.



Wynikiem zajęć jest opanowanie umiejętności tworzenia atrakcyjnych, pełnowartościowych serwisów internetowych, a efektem końcowym – serwis internetowy oraz dokumentacja projektu, utworzone samodzielnie przez każdego ucznia.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- wyjaśnić zasadę działania statycznej i dynamicznej strony internetowej;
- wskazać różnice pomiędzy statyczną i dynamiczną stroną internetową;
- wymienić elementy składowe strony internetowej i serwisu internetowego z punktu widzenia twórcy;
- wykonać dokumentację projektu serwisu internetowego;
- utworzyć statyczną stronę internetową w kodzie HTML;
- zainstalować i skonfigurować oprogramowanie do obsługi dynamicznych stron internetowych (serwer WWW) oraz bazy danych;
- utworzyć dynamiczną stronę internetową w języku PHP;
- zaprojektować i utworzyć bazę danych w technologii MySQL;
- zrealizować obsługę bazy danych z poziomu przeglądarki internetowej poprzez utworzony serwis WWW;
- przetestować utworzony serwis i zaprezentować jego działanie.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów (patrz poniżej Metody pracy);
- znać podstawowe znaczniki kodu HTML;
- posiadać podstawową wiedzę na temat baz danych;
- znać przynajmniej w podstawowym zakresie dowolny język programowania.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (projekt),
 - implementacja (realizacja) rozwiązania w postaci strony internetowej, baz danych, serwisu WWW,
 - ewaluacja rozwiązania,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- posłużenie się przez nauczyciela i uczniów objaśnieniami sposobów działania przykładowych serwisów WWW oraz tworzenia serwisu WWW zgodnie z listą kolejnych kroków;
- samodzielne sporządzenie przez uczniów projektów oraz dokumentacji projektowej;

- zapisywanie elementów składowych (poszczególne podstrony) oraz etapów przejściowych projektu na komputerze;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania pomysłów na zastosowanie tworzonego serwisu (np. portal społecznościowy, sklep internetowy);
- przygotowywanie dokumentacji projektowej oraz tworzenie serwisów może odbywać się indywidualnie;
- testowanie utworzonych serwisów może odbywać się w grupach uczniów – uczniowie wspólnie wyszukują i poprawiają błędy i porównują wyniki działania swoich programów;
- końcowym efektem pracy jest prezentacja utworzonego serwisu przez każdego ucznia.

7. Materiały pomocnicze

Nauczyciel może korzystać z tablicy w celu zapisania kolejnych etapów tworzenia serwisu WWW oraz projektowania bazy danych. Na tablicy może również zapisywać i wyjaśniać znaczniki HTML, elementy języka PHP oraz instrukcje SQL. Dla uczniów przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami.



8. Środki dydaktyczne

Uczniowie wykorzystują w czasie zajęć:

- podręcznik dla szkół ponadgimnazjalnych, np.: W. Wrotek, *Technologia informacyjna*, Helion, Gliwice 2006;
- publikacje uzupełniające, np.:
 - Price J., Price L., *Profesjonalny serwis WWW*, Helion, Gliwice 2002,
 - Cohen J., *Serwisy WWW. Projektowanie, tworzenie, zarządzanie*, Helion, Gliwice 2004,
 - Meloni J. C., *PHP, MySQL i Apache dla każdego*, Helion, Gliwice 2007,
 - Ullman L., *PHP i MySQL. Dynamiczne strony WWW*, Helion, Gliwice 2004;
- materiały dotyczące stron internetowych, serwisów internetowych i baz danych, opracowane w projekcie Informatyka +;
- otwarte zasoby edukacyjne w Internecie dotyczące języków HTML, PHP, SQL oraz działania serwera WWW Apache i baz danych MySQL.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest nabycie umiejętności tworzenia interaktywnych serwisów WWW, powinny być rozłożone na kilka lekcji, przy czym zaleca się realizację według następującego toku:

- w części wprowadzającej, nauczyciel objaśnia zasadę działania statycznej i dynamicznej strony internetowej, wskazuje na niezbędne elementy serwisu WWW oraz prezentuje przykładowe serwisy istniejące w Internecie; po takim wprowadzeniu uczniowie formułują tematy własnych serwisów i tworzą pierwszy etap dokumentacji projektu – scenariusz działania serwisu;

- w drugiej części uczniowie instalują i konfigurują oprogramowanie do obsługi serwisu dynamicznego: serwer WWW Apache oraz system baz danych MySQL;
- na kolejnym etapie uczniowie tworzą pierwsze strony swoich serwisów przy użyciu technologii HTML, CSS, PHP;
- następnie uczniowie tworzą bazę danych, po czym wypełniają ją danymi;
- ostatnim zadaniem jest realizacja obsługi bazy danych (przeglądanie i modyfikowanie wartości) z poziomu przeglądarki internetowej za pomocą utworzonego serwisu WWW;
- podsumowaniem pracy jest utworzenie pełnej dokumentacji projektu, zawierającej wyniki testów i wnioski końcowe.

Lekcja 1. Wprowadzenie do WWW, strony statyczne i dynamiczne, przykładowe serwisy WWW. Czas: 120 min

Tytułem wprowadzenia jest prezentowany wykład „Witryna w Internecie – zasady tworzenia i funkcjonowania”. Po wykładzie nauczyciel prezentuje przykładowe serwisy (przykłady bliskie temu, co zaplanowano do realizacji przez uczniów).

Następnie uczniowie wybierają tematy i tworzą scenariusze funkcjonowania swojego serwisu – pierwszy etap dokumentacji projektu.



Lekcja 2. Mój własny serwis WWW – etap I. Czas: 120 min

Drugą część zajęć można zacząć od objaśnienia sposobów podejścia do dużego projektu, jakim jest utworzenie serwisu WWW.

Na podstawie wysłuchanych wiadomości oraz własnych scenariuszy uczniowie określają, jakie są potrzebne „cegielełki”, czyli które elementy są im niezbędne, aby powstał ich serwis WWW. Opisują to w dokumentacji jako założenia i wymagania projektowe.

Następnie wprowadza się odpowiednie narzędzia dla realizacji „cegielełki” – uczniowie instalują i konfigurują oprogramowanie MySQL, Apache i PHP, po czym każdy uczeń przystępuje do realizacji swojego projektu (tworzy poszczególne podstrony serwisu, w tym:

- treść (język HTML),
- wygląd (język CSS),
- sposób działania (język PHP).

Podstawowe pojęcia i struktura języków HTML, CSS oraz PHP uczniowie znajdą m.in. w materiałach dostępnych w Internecie: <http://kurshtml.boo.pl>, <http://phpkurs.pl/> oraz w materiałach opracowanych w ramach projektu Informatyka +.

Lekcja 3. Mój własny serwis WWW – etap II. Czas: 120 min.

W tej części uczniowie zapoznają się z metodyką tworzenia i funkcjonowania relacyjnej bazy danych. W tym celu mogą posłużyć się materiałami dostępnymi w ramach otwartych zasobów edukacyjnych, np. na portalu Ważniak, <http://wazniak.mimuw.edu.pl>.

Kolejny etap tworzenia własnego serwisu to projektowanie i tworzenie bazy danych (baza danych MySQL, język SQL). Uczniowie postępują zgodnie z ćwiczeniami w zeszyty dydaktycznym pt. „Tworzenie serwisów internetowych” Informatyka+.

W drugiej części zajęć uczniowie uczą się operowania na bazie danych MySQL z poziomu skryptów PHP. Ćwiczenia obejmują:

- łączenie się z MySQL poprzez PHP,
- operowanie na danych z bazy MySQL (dodawanie, usuwanie, modyfikacja danych).

Jest to jeden z trudniejszych etapów, wymagający zwrócenia uwagi na błędy w kodzie i spójność danych w bazie MySQL z instrukcjami języka PHP.

Lekcja 4. Mój własny serwis WWW – etap III. Czas: 90 min.

W tej części uczniowie testują, poprawiają własne serwisy internetowe. Jest to również czas na dokończenie dokumentacji projektowej, w której każdy z uczniów opisuje działanie utworzonego serwisu, wyniki testów oraz wnioski. Po ukończeniu dokumentacji, każdy z uczniów publicznie prezentuje swoją pracę.



NAGRYWANIE I OBRÓBKA FILMÓW

Andrzej Majkowski

Politechnika Warszawska


amajk@ee.pw.edu.pl

1. Zagadnienia programowe

Zagadnienia związane z umiejętnością tworzenia filmów oraz pozyskiwania i przetwarzania różnych materiałów multimedialnych pojawiają się na lekcjach informatyki.

Informatyka, III etap edukacyjny

W czwartym dziale od ucznia oczekuje się umiejętności opracowywania za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.

- 
- Opracowywanie, za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. Uczeń:
 - przy użyciu edytora grafiki tworzy kompozycje z figur, fragmentów rysunków i zdjęć, umieszcza napisy na rysunkach, tworzy animacje, przekształca formaty plików graficznych;
 - tworzy dokumenty, zawierające różne obiekty (na przykład tekst, grafikę, tabele, wykresy i tym podobne) pobrane z różnych programów i źródeł;
 - tworzy i przedstawia prezentację z wykorzystaniem różnych elementów multimedialnych, graficznych, tekstowych, filmowych i dźwiękowych własnych lub pobranych z innych źródeł.

Informatyka, IV etap edukacyjny, zakres podstawowy

W czwartym dziale od ucznia oczekuje się umiejętności opracowania różnych informacji za pomocą komputera, w tym łączenia różnych materiałów multimedialnych i tworzenia filmów.

- Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
 - opracowuje obrazy i filmy pochodzące z różnych źródeł, tworzy albumy zdjęć.

Informatyka, IV etap edukacyjny, zakres rozszerzony

W pierwszym dziale od ucznia oczekuje się znajomości sposobów reprezentowania obrazów i animacji w komputerze. W czwartym dziale od ucznia oczekuje się znajomości sposobów reprezentowania obrazów i animacji w komputerze (modele barw, grafika rastrowa i wektorowa, formaty zapisu plików multimedialnych) oraz przetwarzania obrazów i filmów.

- Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:

- 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;
 - 4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania.
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
- 1) opisuje podstawowe modele barw i ich zastosowanie;
 - 2) określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;
 - 3) przetwarza obrazy i filmy, np.: zmienia rozdzielczość, rozmiar, model barw, stosuje filtry.

Plastyka, III etap edukacyjny

W drugim dziale od ucznia oczekuje się umiejętności tworzenia wypowiedzi – ekspresji przez sztukę. Kurs obejmuje m.in. tworzenie kartki pocztowej, kalendarza ściennego, fotoksiążki, animacji oraz filmu.

2. Tworzenie wypowiedzi – ekspresja przez sztukę. Uczeń:
 - 1) podejmuje działalność twórczą, posługując się środkami wyrazu sztuk plastycznych, innych dziedzin sztuki (fotografika, film) i elementami formy przekazów medialnych, w kompozycji na płaszczyźnie oraz w przestrzeni rzeczywistej i wirtualnej (stosując określone materiały, narzędzia i techniki właściwe dla tych dziedzin sztuki i przekazów medialnych);
 - 2) realizuje projekty w zakresie sztuk wizualnych, w tym służące przekazywaniu informacji dostosowanej do sytuacji komunikacyjnej oraz uczestnictwu w kulturze społeczności szkolnej i lokalnej (stosując także narzędzia i wytwory mediów środowiska cyfrowego).

Uwaga. Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe. W tych miejscach powinien być adresowany do uczniów z lepszym przygotowaniem informatycznym i wykazujących większe zainteresowania pogłębieniem znajomości różnych technik multimedialnych, w szczególności powiązanych z tworzeniem filmów. problemy mogą pojawić się ze zrozumieniem formatów zapisu i kompresji obrazów ruchomych i dźwięku DV I MPEG.

2. Temat zajęć

Nagrywanie i obróbka filmów

Stworzenie nawet prostego filmu wymaga zastosowania wielu złożonych technik i algorytmów związanych z przetwarzaniem, zapisem, kompresją materiałów multimedialnych. W trakcie zajęć uczniowie poznają sposoby rejestracji materiału wideo z użyciem różnych narzędzi (różne typy kamer, aparat fotograficzny z funkcją nagrywania, telefon komórkowy). Poznają także podstawowe formaty zapisu sygnału wideo w postaci cyfrowej (DV, MPEG1, MPEG2, MPEG4). W dalszej części są przedstawione formy nieliniowego montażu wideo – uczniowie zapoznają się z oprogramowaniem służącym do takiego montażu, poznają techniki poprawy jakości filmów i wzbogacania ich o efekty specjalne. Na końcu jest przedstawiony sposób przygotowania materiału audio i wideo do zapisu na nośniku DVD.

3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- przedstawić zasadę działania kamery cyfrowej;
- scharakteryzować podstawowe formaty zapisu obrazów i filmów (DV, MPEG);
- wykorzystać program do nieliniowego montażu wideo do:
 - przenoszenia materiału wideo i audio do komputera,
 - edycji materiału wideo i audio,
 - dodawania efektów specjalnych do materiału filmowego,
 - zastosowania różnych filtrów do poprawy jakości materiału wideo;
- zapisać zarejestrowany materiał wideo i audio w różnych formatach;
- przygotować materiał filmowy do zapisu na płycie DVD.

4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- znać podstawy obsługi aparatu fotograficznego i cyfrowej kamery;
- znać w stopniu podstawowym przynajmniej jeden program komputerowy do edycji filmów.

5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
 - opis, dyskusja i zrozumienie sytuacji problemowej,
 - podanie specyfikacji problemów do rozwiązania,
 - zaprojektowanie rozwiązania (projekt),
 - implementacja (realizacja) rozwiązania w postaci materiału filmowego,
 - ewaluacja rozwiązania,
 - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania;
- posłużenie się przez nauczyciela i uczniów objaśnieniami przykładowych sposobów wykonywania własnych filmów za pomocą listy kroków;
- samodzielne sporządzenie przez uczniów projektów polegających na stworzeniu filmu;
- zapisywanie elementów składowych (poszczególne obrazy, klipy filmowe) oraz etapów przejściowych projektu na komputerze;
- prezentacja otrzymanych rozwiązań.

6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- burza mózgów prowadzona przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania pomysłu na temat projektu (opracowania koncepcji filmu);
- gromadzenie elementów składowych projektu (poszczególnych obrazów, klipów filmowych);
- projekty filmowe mogą być przygotowywane indywidualnie lub w parach uczniów;
- końcowym efektem pracy jest prezentacja gotowego filmu.

7. Materiały pomocnicze

Nauczyciel może korzystać z tablicy w celu przedstawienia poszczególnych etapów wykonywania projektu. Dla uczniów przydatne mogą być także nośniki informacji do przenoszenia danych między komputerami.

8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- publikacje uzupełniające, np.:
 - *Adobe Premiere Elements. Domowe studio wideo*, praca zbiorowa, Helion, Gliwice 2007,
 - **Beach A.**, *Kompresja dźwięku i obrazu wideo*, Helion, Gliwice 2009,
 - **Danowski B.**, *Komputerowy montaż wideo. Ćwiczenia praktyczne*, Helion, Gliwice 2006;
- materiały dotyczące obróbki grafiki i filmów, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, program do nieliniowego montażu wideo oraz inne dostępne bezpłatnie programy do edycji i tworzenia grafiki, animacji i filmów.

9. Przebieg zajęć (kolejnych lekcji)

Zajęcia, których celem jest nabycie umiejętności tworzenia filmów oraz przetwarzania różnych treści multimedialnych, mogą być rozłożone na kilka lekcji, przy czym zaleca się ich realizację według następującego trybu:

- W części pierwszej nauczyciel objaśnia koncepcję tworzenia filmu przy użyciu narzędzi cyfrowych. W tej części uczniowie kompletują materiały audio i wideo, które będą później wykorzystane do utworzenia filmu.
- Część druga dotyczy przenoszenia różnorodnych materiałów audio i wideo z urządzeń do komputera. Na tym etapie uczniowie powinni zapoznać się ze sposobem obsługi jednego z programów do nieliniowego montażu wideo w *zakresie opcji przechwytywania materiału wideo*. Powinni również poznać formaty zapisu materiału wideo.
- Część trzecia dotyczy edycji sekwencji wizyjnej oraz dźwięku. Uczniowie powinni zapoznać się ze sposobem obsługi jednego z programów do nieliniowego montażu w *zakresie edycji materiału wideo*.
- Część czwarta jest poświęcona metodom eksportu materiału wideo do formatu MPEG-2 oraz przygotowywaniu materiału do nagrania na płytę DVD.

Lekcja 1. Pozyskiwanie materiału wideo. Czas: 45 min.

Sprzęt potrzebny do wykonania zadania:

- kamery wideo pracujące w standardzie miniDV, kamery internetowe, telefon komórkowy z opcją nagrywania wideo,
- cyfrowy aparat fotograficzny.

Celem tej lekcji jest zapoznanie uczniów z technikami przygotowywania materiału filmowego. Jest to jednocześnie pierwszy etap przygotowania własnego filmu.

Na początku uczniowie powinni zapoznać się ze sprzętem. Szczególną uwagę należy zwrócić na sposób obsługi i funkcje kamer cyfrowych. Na etapie przygotowania materiału filmowego gromadzimy

różne media: pliki, które będą później elementami montowanego filmu: grafiki, dźwięki i klipy filmowe. Na tym etapie najpierw należy przygotować **własną koncepcję krótkiego filmu** do montażu. Film będzie montowany z różnych elementów dlatego należy przygotować: materiał filmowy nagrany różnymi kamerami, własne komentarze nagrane przy użyciu oddzielnych mikrofonów, zdjęcia wykonane aparatem cyfrowym. Wszystkie te elementy mogą być zapisywane w różnych formatach. Później będą one łączone w jeden film.

Każdy uczeń (lub zespół uczniów) powinien na tym etapie mieć:

- przygotowaną koncepcję własnego filmu do montażu (bez użycia komputera należy zastanowić się, zapisać co się chce przedstawić);
- skompletowane klipy filmowe nagrane kamerami pracującymi w standardzie miniDV, kamerami internetowymi i ewentualnie nagrane przy użyciu telefonu komórkowego z opcją rejestracji wideo;
- zdjęcia, które mają być później ilustracją do filmu (np. plansze tytułowe, przerywniki);
- muzykę mogącą służyć za podkład muzyczny do scen w filmie.

Lekcja 2. Zgrywanie sekwencji wizyjnej. Czas: 45 min.

Sprzęt potrzebny do wykonania zadania:

- komputer,
- kamery wideo pracujące w standardzie miniDV, kamery internetowe, telefon komórkowy z opcją nagrywania wideo,
- cyfrowy aparat fotograficzny,
- oprogramowanie do montażu nieliniowego, np. *Pinnacle Studio*, *Adobe Premiere Elements*.

Pierwszą czynnością, od której rozpoczyna się każdy montaż nieliniowy, jest przeniesienie zapisanego materiału wideo z kasyety na twardy dysk komputera określane potocznie przechwytywaniem. Celem tej lekcji jest zapoznanie uczniów z metodami zgrywania sekwencji wizyjnej na dysk komputera.

Na wstępie uczniowie powinni zapoznać się ze sposobem obsługi jednego z programów do nieliniowego montażu wideo (np. *Pinnacle Studio*, *Adobe Premiere Elements*) w zakresie opcji *przechwytywania materiału wideo*. Powinni również poznać formaty zapisu materiału wideo *DV*, *MPEG2*, *MPEG4*.

Sekwencje wideo możemy przechwytywać jako duże partie materiału lub używając listy cięć wybranych przez nas ujęć. Rozwinięciem tej metody jest tzw. skanowanie taśmy, podczas którego automatycznie zostają wyszukane wszystkie zmiany kodu czasowego powstałe na taśmie w wyniku zatrzymania procesu nagrywania (przez naciśnięcie przycisku Stop lub Pause). W rezultacie uzyskujemy gotową listę cięć, według której można automatycznie przechwycić materiał wideo podzielony na poszczególne ujęcia. Taki sposób przechwytywania umożliwia również pakiety *Pinnacle Studio* i *Adobe Premiere Elements*.

Na etapie zgrywania sekwencji wizyjnej uczniowie powinni:

- zapoznać się z metodami łączenia poszczególnych urządzeń (kamera, aparat fotograficzny, cyfrowy aparat fotograficzny, mikrofon i wzmacniacz mikrofonowy) z komputerem;
- zapoznać się z metodami, możliwościami, magistralami umożliwiającymi skopiowanie materiałów multimedialnych na dysk komputera (FireWire i USB);

- zapoznać się z zasadą działania i podstawowymi funkcjami programu i pakietu oprogramowania Pinnacle Studio i Adobe *Premiere Elements*;
- przygotować materiały do montażu:
 - materiały własne przechwycone z kamery,
 - materiały własne przechwycone z kamery internetowej lub telefonu komórkowego,
 - zdjęcia cyfrowe,
 - gotowe przykładowe sekwencje wizyjne;
- przygotować dodatkowe materiały graficzne (np. plansze, napisy);
- jeszcze raz przemyśleć koncepcję własnego krótkiego filmu do montażu (bez użycia komputera, należy zastanowić się, co chcemy zamieścić w filmie, które materiały będą użyte, a które nie).

Lekcja 3. Edycja sekwencji wizyjnej oraz dźwięku. Czas: 90 min.

Sprzęt potrzebny do wykonania zadania:

- komputer,
- oprogramowanie do montażu nieliniowego Pinnacle Studio, Adobe *Premiere Elements*.

Uczniowie powinni zapoznać się ze sposobami korzystania z jednego z programów do nieliniowego montażu wideo (np. Pinnacle Studio, Adobe *Premiere Elements*) w zakresie możliwości edycji materiału wideo. Mimo, że na rynku oprogramowania istnieje wiele programów dedykowanych do montażu nieliniowego, wszystkie one mają podobny charakter. Wyglądają one jak wirtualne studio montażowe, ze stołem montażowym, oknem z materiałami filmowymi oraz innymi elementami niezbędnymi w pracy montażysty. Cały proces wykonywany jest z wykorzystaniem technologii *drag and drop*, dzięki temu użytkownik w prosty sposób może wyciągać z wirtualnej biblioteki materiałów wizyjnych pliki wideo i układać je na wirtualnym stole montażowym. Pakiety Pinnacle Studio, czy Adobe *Premiere Elements* oferują niemal wszystko to, co powinno być wykonywane przez program do cyfrowego montażu nieliniowego, czyli edycję obrazu wideo, obsługę kodów czasowych, synchronizację dźwięku i obrazu. W skład pakietów wchodzi również m.in. biblioteka efektów przejścia oraz biblioteka filtrów obrazu wideo. W zasadzie wszystkie zintegrowane pakiety programowe do nieliniowego montażu wideo oferują te same opcje. Różnice mogą pojawić się w interfejsie komunikacji z użytkownikiem, tzn. poszczególne programy mogą być trochę inaczej obsługiwane. Z dwóch wymienionych, pakiet Pinnacle Studio może się wydawać uczniowi – początkującemu użytkownikowi łatwiejszy do opanowania.

Edycja sekwencji wizyjnej jest najważniejszym procesem każdego montażu materiału wideo. Wykorzystując możliwości montażu nieliniowego, możemy znacznie uatrakcyjnić i ożywić niezbyt ciekawe nagrania. Podstawowymi zabiegami edycyjnymi są precyzyjne docinanie, kopiowanie i układanie poszczególnych ujęć w żądanej kolejności. Ponadto możliwe jest tworzenie stop-klatek, regulacja tempa oraz kierunku odtwarzania klipu wideo. Wszystkie programy edycyjne wyposażone są w ścieżki wideo i audio, oś lub linię czasu oraz podstawowe narzędzia edycyjne. Nasze edytowane ujęcia umieszczane są na poszczególnych ścieżkach wideo i audio. Podstawową operacją przeprowadzaną podczas edycji jest cięcie, które ma wyodrębnić poszczególne ujęcia z materiału wideo. Do tego celu używamy zazwyczaj wirtualnych „nożyczek”. Precyzyjne dobieranie początku i końca ujęcia możliwe jest również dzięki zastosowaniu znaczników wejścia i wyjścia. Stop-klatki w montowanym materiale wideo można uzyskać na dwa sposoby. Pierwszym z nich jest zaimportowanie do programu montażowego klatki w postaci pliku graficznego np. BMP, TIFF, JPEG, jak również ustawienie



jej bezpośrednio z ujęcia umieszczonego na ścieżce wideo. Zmiana tempa odtwarzanego materiału wideo wpływa na zwolnienie lub przyspieszenie akcji filmu, natomiast zmianę kierunku odtwarzania możemy zastosować w celu uzyskania efektu specjalnego. Oprócz możliwości odpowiedniego przycięcia sekwencji na tym etapie możemy rozjaśnić bądź przyciemnić sekwencję wideo, lub też zastosować różnego rodzaju filtry.

Wszystkie programy do obróbki wideo na etapie edycji oferują dwa tryby pracy: Storyboard i Timeline. Tryb **Storyboard** jest najszybszą i najprostszą metodą dodawania klipów do tworzonego właśnie filmu, poprzez zastosowanie przenoszenia poszczególnych sekwencji. W trybie tym każdy obrazek reprezentuje pojedyncze zdarzenie w edytowanym filmie. Istnieje możliwość przenoszenia żądanych klipów filmowych i umieszczania ich w założonym miejscu. Dodatkowo pomiędzy klipami można dołączyć różne efekty przejścia. Tryb **Timeline** umożliwia dokładne dopasowywanie efektów i edycję materiału. W trybie tym pokazywane jest dokładnie chronologia pojawiania się poszczególnych elementów filmu na poszczególnych ścieżkach. Klipy filmowe w tym trybie mogą być przenieszone z bibliotek na odpowiednie ścieżki i są wyświetlane jako krótkie sekwencje.

Elementy edycji klipów filmowych obejmują:

- dodawanie klipu filmowego do ścieżki,
- przycinanie klipu filmowego,
- podział klipu filmowego na segmenty,
- podział na sceny,
- korzystanie z filtrów.



Efekty specjalne

Obecnie każdy program edycyjny ma różnego rodzaju efekty specjalne, które możemy podzielić na: efekty przejść (miks), filtry obrazu, kluczowanie obrazu, nakładanie obrazu oraz wprowadzanie napisów. Z efektów przejść korzystamy w sytuacji, gdy zamierzamy w atrakcyjny sposób połączyć ze sobą dwa różne ujęcia. Generalnie, efekty przejść można podzielić na dwuwymiarowe (2D) i trójwymiarowe (3D). Najprostszymi efektami dwuwymiarowymi są przenikania obrazu (miks) oraz tzw. kurtynki, natomiast efekty trójwymiarowe najczęściej kojarzone są z efektem odwijanej kartki lub transformacji obrazu w trójwymiarową bryłę. Rodzaj oraz liczba efektów przejść zależne są od klasy i typu programu edycyjnego.

Filtry obrazu mogą spełniać rozmaite funkcje. Pierwszą z nich jest korekcja parametrów obrazu. Do tego celu są przeznaczone filtry regulujące balans kolorów (RGB), jasność, kontrast, nasycenie itp. Drugą z nich jest wprowadzanie prostych efektów, takich jak odwracanie obrazu w pionie i poziomie, docinanie itp. Natomiast trzecią z nich jest wprowadzanie efektów specjalnych: rozmycia, wyostrenia, płaskorzeźby, kalejdoskopu, fali, deformacji, efektów świetlnych itp.

Kolejny efekt **kluczowania** w uproszczeniu polega na usunięciu z obrazu wideo pewnych obszarów charakteryzujących się wspólnymi cechami (np. kolorem, jasnością itp.) i nadaniu im przezroczystości. Dzięki kluczowaniu uzyskujemy możliwość nakładania na siebie poszczególnych warstw z kolejnymi obrazami wideo lub grafiką. Najczęściej stosowanymi rodzajami kluczy są:

- color key – usuwanie obszarów o określonym kolorze;
- chroma key – usuwanie obszarów o określonym kolorze wraz z jego odcieniami (działa dużo precyzyjniej od color key, ponieważ uwzględnia poziom nasycenia koloru);
- luma key – usuwanie obszarów jedynie o określonym poziomie jasności;

- blue screen – usuwanie niebieskich obszarów (często zamiast koloru niebieskiego używa się koloru zielonego oraz klucza green screen) – są to najczęściej stosowane klucze używane do usunięcia tła za filmowaną postacią;
- alpha channel – usuwanie obszarów zdefiniowanych specjalną maską zapisaną w dodatkowym kanale alfa.

Niestety często się zdarza, że efekty dalekie są od ideału. Wynika to z trudności w uzyskaniu jednolitych obszarów obrazu, które mogłyby być precyzyjnie usunięte. Profesjonalne rozwiązania posługują się specjalnie przygotowywanymi maskami umożliwiającymi bardzo precyzyjne kluczowanie.

Nakładanie scen

Kolejnym efektem jest nakładanie obrazu. Mamy tu do czynienia z efektami powstałymi w wyniku nałożenia na obraz wideo jednego lub więcej często zdeformowanych obrazów pochodzących z plików wideo lub plików graficznych. Dobrym tego przykładem mogą być efekty typu obraz w obrazie (ang. PIP – *Picture In Picture*). Deformacje mogą dotyczyć zarówno wielkości, jak i kształtu nakładanego obrazu, który dodatkowo może być statyczny lub ruchomy. Wprowadzanie napisów jest efektem polegającym na nałożeniu na obraz wideo przede wszystkim napisów, ale również grafik lub małych animacji. Nakładanie napisów jest możliwe dzięki towarzyszącemu mu efektowi kluczowania (najczęściej alpha channel). Większość programów do montażu wideo ma wbudowane proste funkcje do tworzenia napisów, jednak oferują one jedynie podstawowe możliwości.

Bardzo ważnym elementem każdego filmu jest ekran tytułowy oraz ekran z napisami końcowymi. Aby stworzyć napisy, które mają być potem wykorzystane przy sekwencji początkowej lub końcowej, należy wybrać ramkę, do której dodawać chcemy napisy tytułowe. Następnie tworzymy napis tytułowy. Do tekstu można dodać efekty, np. napisy przewijane, napisy zanikające, inne efekty animacji. Na etapie edycji wideo uczniowie powinni dokonać montażu wideo z zastosowaniem kroków przedstawionych powyżej.

Dodawanie ścieżki dźwiękowej

Produkcja materiału wizyjnego nie obędzie się bez dodatkowej ścieżki audio. W pakietach Pinnacle Studio i Adobe *Premiere Elements* możliwe jest zarówno nagrywanie własnej narracji oraz zgrzywanie muzyki z płyty CD do pliku typu wav. Istnieje również możliwość dodawania zewnętrznych plików dźwiękowych.

Podobnie jak obraz wideo, również dźwięk może być poddawany różnym zabiegom edycyjnym. Podstawową operacją wykonywaną na ścieżkach audio jest ustawianie żądanego poziomu dźwięku. Ustawień dokonuje się za pomocą krzywych z obwiedniami opisującymi żądany poziom dźwięku. Warto tu zauważyć, że przy użyciu wspomnianych wcześniej efektów przejść również konieczna jest regulacja głośności zachodzących na siebie materiałów audio, co jest niezbędne do płynnego przejścia dźwięku z jednego materiału do drugiego. Do niedawna ustawiało się to ręcznie, natomiast obecnie większość dostępnych na rynku programów ma do tego specjalne narzędzia lub też tworzy je automatycznie. Dodatkowo w zaawansowanych programach dostępna jest również regulacja panoramy dźwięku, umożliwiająca precyzyjne balansowanie i ustawianie dźwięku w kanałach.

Na etapie dodawania dodatkowej ścieżki dźwiękowej do wideo (punkt c) słuchacze powinni wykonać następujące operacje:

- nagrywanie własnej narracji,
- zgrzywanie muzyki z płyty CD,



- dodawanie zewnętrznych plików dźwiękowych,
- przycinanie plików dźwiękowych,
- dodawanie efektów wyciszania dźwięków.

Lekcja 4. Eksport materiału audio i wideo do formatu wymaganego do nagrania na płytę DVD. Czas: 45 min.

Ta lekcja jest poświęcona przygotowywaniu materiału audio i wideo do nagrania na płytę DVD. Proces ten obejmuje następujące elementy, z którymi powinni być zapoznani uczniowie:

- a) konwersja plików AVI do odpowiednich formatów zgodnych ze standardami zapisu płyt DVD,
- b) budowanie menu scen (authoring),
- c) nagrywanie przygotowanych materiałów na nośnik DVD.

Przygotowanie materiału wizyjnego do nagrania na płycie DVD bądź CD składa się z kilku etapów. Pierwszy etap to zaplanowanie zawartości płyty. Konieczne jest zastanowienie się nad ilością i standardem zapisu ścieżek dźwiękowych, stosowaniem elementów interaktywnych oraz różnymi wersjami językowymi. Konieczne jest również zaplanowanie elementów oraz formy menu. Etap tworzenia menu to wykorzystanie najczęściej gotowego szablonu, który można oczywiście modyfikować. Niektóre programy umożliwiają całkowite przemodelowanie szablonu przez zmianę układu ikon dla rozdziałów, czcionki w napisach czy tła. Kolejnym etapem jest odpowiedni montaż oraz kompresja wszystkich materiałów, które chcemy wykorzystać do nagrania płyty. Mowa tu przede wszystkim o materiale wizyjnym, ale również o przygotowaniu ścieżek dźwiękowych. Ostatnim etapem jest proces *authoringu*. Termin ten oznacza wszystkie procesy, jakie muszą być zrealizowane po zakodowaniu dźwięku i obrazu, a przed otrzymaniem końcowego produktu. Etap ten polega na połączeniu ze sobą obrazu i wszystkich ścieżek dźwiękowych, podpisów do dialogów, stworzeniu menu oraz podziału na sceny. Korzystając z programów Pinnacle Studio i Adobe *Premiere Elements* większość tych operacji jest wykonywanych automatycznie. Końcowy etap to etap generowania wypalanie płyty z gotowym filmem.

Na tym etapie uczniowie powinni wykonać następujące operacje:

- zapoznać się z zasadą działania i podstawowymi funkcjami programów Pinnacle Studio i Adobe *Premiere Elements* dotyczącymi authoringu;
- na podstawie założeń przedstawionych podczas lekcji przez prowadzącego przygotować koncepcję tworzonej płyty;
- przeprowadzić konwersję wykorzystywanych materiałów do formatu MPEG-2;
- nagrać gotowy materiał filmowy na dysk komputera lub płytę DVD.

Na zajęciach uczniowie korzystają z plików graficznych, audio i wideo, udostępnionych na platformie edukacyjnej Informatyka+ oraz z własnych zdjęć, muzyki i materiałów filmowych.

Podczas tworzenia filmów uczniowie postępują zgodnie z poleceniami opisanymi w ćwiczeniach w zeszyte dydaktycznym.

Po zakończeniu montowania filmu każdy z uczniów publicznie prezentuje swoją pracę.

BIOGRAMY AUTORÓW



Dariusz Chaładyniak, dr inż. geoinformatyk, teleinformatyk, od ponad 15 lat wykładowca akademicki. Specjalizuje się w systemach i sieciach teleinformatycznych oraz przetwarzaniu równoległym i klastrach komputerowych. Instruktor Akademii Regionalnej Cisco CCNA. Koordynator merytoryczny modułu Sieci komputerowe w projekcie Informatyka+. Interesuje się naukami ścisłymi oraz kulturą dalekiego wschodu. Jego pasje to informatyka i sport.



Marek Cygan, doktorant na wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego na kierunku informatyka. Jako student uczestnik wielu konkursów programistycznych, aktualnie członek kadry przygotowującej studentów do startów w zawodach informatycznych. W ramach działalności dydaktycznej prowadzi również kółko w Zespole Szkół Ogólnokształcących nr 6 w Bydgoszczy. W przeszłości juror Olimpiady Informatycznej.



Krzysztof Diks całe życie zawodowe związał z Wydziałem Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, gdzie ukończył studia informatyczne, obronił doktorat i uzyskał stopień naukowy doktora habilitowanego. Jego badania dotyczą algorytmiki, a w szczególności algorytmicznej teorii grafów, obliczeń równoległych i rozproszonych obliczeń odpornych na błędy. Opublikował ponad 50 prac naukowych. Jest współautorem popularnego podręcznika akademickiego *Algorytmy i struktury danych*. Wspólnie z kolegami z Wydziału przetłumaczył na język polski światowe bestsellery algorytmiczne *Wprowadzenie do algorytmów i struktur danych* oraz *Sztukę programowania*. Był promotorem ponad 70 prac magisterskich i 6 doktoratów.

Od lat zaangażowany w edukację informatyczną, zarówno na poziomie akademickim, jak i szkolnym. Jest współautorem standardów kształcenia w zakresie informatyki oraz programu studiów informatycznych na Uniwersytecie Warszawskim. Kierował projektem utworzenia portalu z elektronicznymi materiałami dydaktycznymi dla studiów informatycznych I i II stopnia. W projekcie brało udział blisko 500 pracowników z czterech uczelni: Uniwersytetu Jagiellońskiego, Uniwersytetu Warszawskiego, Politechniki Poznańskiej i Politechniki Warszawskiej. W ramach projektu przygotowano programy nauczania, sylabusy przedmiotów i materiały dydaktyczne do studiów informatycznych, zarówno inżynierskich, jak i licencjackich. Portal z materiałami (wazniak.mimuw.edu.pl) jest otwarty i cieszy się wielką popularnością wśród studentów i nauczycieli akademickich.

Popularyzator informatyki. Od roku 1993 członek, a od roku 1999 – przewodniczący Komitetu Głównego Olimpiady Informatycznej. Autor zadań olimpijskich i wychowawca pokoleń olimpijczyków. Przez Olimpiadę przewinęło się ponad 12000 młodych informatyków. Przewodniczył Międzynarodowej Olimpiadzie Informatycznej w roku 2005, która miała miejsce w Nowym Sączu (blisko 800 uczestników z ponad 60 krajów z całego świata). Wspólnie z profesorem Janem Madeyem opiekuje się drużynami Uniwersytetu Warszawskiego w programowaniu zespołowym. Opiekun i nauczyciel mistrzów świata w programowaniu zespołowym w latach 2003 i 2007. Był inicjatorem pierwszego internetowego konkursu programistycznego Pogromcy Algorytmów, który obecnie jest rozgrywany corocznie pod nazwą Potyczki Algorytmiczne (kilka tysięcy uczestników, wyzywające i kształcące zadania).





dr Grzegorz Grochowski, wykładowca, autor i organizator szkoleń dla nauczycieli z zakresu metodyki, dydaktyki i wychowania a także warsztatów dla rodziców i młodzieży. Współpracuje z organizacjami pozarządowymi i instytucjami publicznymi w zakresie realizacji projektów społecznych i edukacyjnych. Jego zainteresowania informatyką sięgają początku tego wieku, gdy jako konsultant współpracował z jednym z najpopularniejszych polskich portali – Wirtualną Polską. Jest autorem artykułów i wykładów na temat zastosowania Internetu w edukacji a także zagadnień etycznych i wychowawczych związanych z użytkowaniem tego medium. Współpracownik i członek gdyńskiego Stowarzyszenia „Talent”, zajmującego się wyłanianiem młodych uzdolnionych informatyków i kształceniem ich do poziomu mistrzowskiego. Od kilkunastu lat zaangażowany w branży wydawniczej jako autor i redaktor książek, podręczników, pomocy edukacyjnych i innych materiałów o charakterze naukowym i popularyzatorskim.



Tomasz Idziaszek, doktorant na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Od dziesięciu lat związany z konkursami informatycznymi, finalista konkursów indywidualnych (TopCoder, Google Code Jam, Potyczki Algorytmiczne) i zespołowych (Mistrzostwo Europy Środkowej w Programowaniu Zespołowym w roku 2004). Członek Jury Olimpiady Informatycznej, autor zadań w konkursie programistycznym Potyczki Algorytmiczne, członek kadry naukowej na Obozie Naukowo-Treningowym im. A. Kreczmara, organizator treningów programistycznych dla studentów Uniwersytetu Warszawskiego. Autor artykułów i redaktor działu informatyki w popularnonaukowym miesięczniku *Delta*.



Hanna Kiedewicz-Nappi, nauczyciel dyplomowany, pedagog szkolny w ZS UMK Gimnazjum i Liceum Akademickie w Toruniu. Posiada wybitne osiągnięcia w pracy dydaktycznej, wychowawczej i opiekuńczej. Ukończyła dodatkowe studia i liczne kursy z zakresu pracy z uczniem zdolnym m.in. Podyplomowe Studia z Pedagogiki Zdolności i Twórczości oraz Studia Podyplomowe w zakresie Technologii Informatycznej w Nauczaniu Niestacjonarnym, a także z Pedagogiki Korekcyjno-Kompensacyjnej. Jest inicjatorem i opiekunem Klubu Wolontariusza, a od 2000 roku organizuje coroczne obozy integracyjne dla uczniów klas pierwszych gimnazjum i liceum. Od 2004 roku współpracuje z Polską Akcją Humanitarną, zaproponowała i koordynowała realizację autorskiego programu Edukacji Humanitarnej, koordynowała projekt Szkoła Humanitarna. Współpracuje ze środowiskiem lokalnym biorąc aktywny udział w konferencjach naukowych poświęconych pracy z uczniem zdolnym: Teoretyczne i praktyczne aspekty uczniów zdolnych, Nadpobudliwość psychoruchowa dzieci i młodzieży, Zdolność i talenty w perspektywie integracji europejskiej, Kształcenie uczniów zdolnych edukacją XXI wieku. Bierze czynny udział w badaniach naukowych nad uczniem zdolnym, w ramach współpracy z Uniwersytetem Mikołaja Kopernika w Toruniu. Interesuje się kulturą regionu Morza Śródziemnego, językiem włoskim, w wolnych chwilach zajmuje się ogrodem oraz propagowaniem zdrowego trybu życia.



Piotr Kopiał, doktorant na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej, specjalność: telekomunikacja. Pracownik Ośrodka Kształcenia na Odległość Politechniki Warszawskiej (OKNO PW). Od roku 2006 zajmuje się problematyką kształcenia na odległość. Specjalista i konsultant w zakresie tworzenia i wdrażania narzędzi i technologii e-learningowych (portale i platformy edukacyjne, narzędzia dedykowane). Wykładowca na studiach inżynierskich i podyplomowych, prowadzonych w formule mieszanej (ang. *blended learning*). Autor i współautor podręczników multimedialnych do studiów na odległość. Od 2009 r. główny administrator platformy edukacyjnej Fronter w OKNO PW. Przewodniczący Uniwersyteckiego Centrum Kompetencyjnego Technologii Oprogramowania IBM na Politechnice Warszawskiej. Członek IBM Professional Community.



Bolesław Kulbabiński, student Jednoczesnych Studiów Informatyczno-Matematycznych na Uniwersytecie Warszawskim. Laureat XV Olimpiady Informatycznej. Od trzech lat prowadzi koło informatyczne przygotowujące gimnazjalistów i licealistów do startów w konkursach programistycznych. Jego zainteresowania obejmują algorytmikę, optymalizację kodu oraz systemy operacyjne. Pracował jako programista m.in. dla firm Google i NVIDIA.



Anna Beata Kwiatkowska, wykładowca na Wydziale Matematyki i Informatyki Uniwersytetu Mikołaja Kopernika, nauczyciel informatyki w eksperymentalnej szkole uniwersyteckiej Gimnazjum i Liceum Akademickim przy UMK w Toruniu. Wychowała wielu finalistów i laureatów Olimpiady Informatycznej i innych konkursów. Aktywnie zaangażowana w problemy edukacji informatycznej w swojej szkole, na Uczelni i w skali kraju. Organizatorka cyklicznej konferencji dla nauczycieli „Informatyka w Edukacji” i wielu inicjatyw dla uczniów i nauczycieli, m.in. Konkursu Informatycznego „Bóbr”. Członek Rady ds. Edukacji Informatycznej i Medialnej przy Ministrze Edukacji Narodowej, Komitetu Głównego Olimpiady Informatycznej, Zarządu Głównego i Prezes Oddziału Kujawsko-Pomorskiego Polskiego Towarzystwa Informatycznego.



Jakub Łacki, doktorant informatyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Medalista międzynarodowych olimpiad informatycznych. Członek jury Olimpiady Informatycznej oraz, w 2008 roku, Bałtyckiej Olimpiady Informatycznej. Od 2006 roku współorganizator Obozu Naukowo-Treningowego im. Antoniego Kreczmar, przygotowującego najzdolniejszych uczniów do startów w zawodach międzynarodowych. W swojej pracy zajmuje się głównie dynamicznymi algorytmami grafowymi oraz algorytmami na grafach planarnych.



Jan Madey, absolwent Wydziału Matematyki i Fizyki Uniwersytetu Warszawskiego. Od początku swej kariery zawodowej związany z Uniwersytetem Warszawskim, ale spędził też blisko 10 lat w czołowych uczelniach Ameryki Północnej i Europy Zachodniej. Autor i współautor pierwszych w Polsce podręczników poświęconych językom programowania Algol 60 i Pascal oraz ponad 100 publikacji. W Uniwersytecie Warszawskim, wieloletni dyrektorem Instytutu Informatyki, prodziekan Wydziału Matematyki, Informatyki i Mechaniki, a następnie prorektor. Obecnie jest Dyrektorem Centrum Otwartej i Multimedialnej Edukacji oraz profesorem zwyczajnym w Instytucie Informatyki. Pełni wiele funkcji, m.in. jest Przewodniczącym Europejskiego Stowarzyszenia EUNIS (*European University Information Systems*), wiceprzewodniczącym Rady Głównej Szkolnictwa Wyższego, przewodniczącym Rady ds. Edukacji Informatycznej i Medialnej przy Ministrze Edukacji Narodowej, przewodniczącym Zarządu Krajowego Funduszu na rzecz Dzieci, członkiem Komitetu Informatyki PAN, członkiem Komitetu Głównego Olimpiady Informatycznej, Organizatorem Krajowym Konkursu Prac Młodych Naukowców Unii Europejskiej (EUCYS) oraz opiekunem drużyn studentów uczestniczących w Akademickich Mistrzostwach Świata w Programowaniu Zespołowym (od 1994 roku). Laureat wielu prestiżowych nagród. Otrzymał m.in. Krzyż Oficerski Orderu Odrodzenia Polski, Medal Komisji Edukacji Narodowej, dyplom Ministra Spraw Zagranicznych za wybitne zasługi dla promocji Polski w świecie, Medal PAN im. Mikołaja Kopernika za wybitne osiągnięcia w działalności na rzecz nauki, liczne nagrody Ministra Edukacji oraz Ministra Nauki i Szkolnictwa Wyższego, nagrodę Złoty Procesor i tytuł Człowieka Roku 2003 redakcji Teleinfo, wyróżnienie firmy IBM Award of Excellence, nagrodę im. Marka Cara, nagrodę Polskiej Izby Informatyki i Telekomunikacji, nagrodę Infostar 2006 oraz nagrodę tygodnika Przekrój Fenomen 2006.





Andrzej Majkowski, absolwent Wydziału Elektrycznego Politechniki Warszawskiej. Od 1995 roku pracownik Instytutu Elektrotechniki Teoretycznej i Systemów Informatyko-Pomiarowych Wydziału Elektrycznego. Stopień doktora nauk technicznych uzyskał w roku 2000. Zainteresowania naukowe: informatyka, cyfrowe przetwarzanie sygnałów, projektowanie i oprogramowanie systemów pomiarowych.



Błażej Osiński, student informatyki i matematyki na Uniwersytecie Warszawskim. Interesuje się algorytmiką i biologią obliczeniową. W szkole z sukcesami brał udział w konkursach i olimpiadach, był też stypendystą Krajowego Funduszu na Rzecz Dzieci. Od kilku lat zaangażowany w prowadzenie zajęć z programowania i algorytmiki w Warszawie oraz w VI Liceum Ogólnokształcącym w Bydgoszczy, którego jest absolwentem. Bierze też udział w przygotowaniu Olimpiady Informatycznej i Olimpiady Informatycznej Gimnazjalistów.



Andrzej Ptasznik, informatyk, specjalista baz danych. Wykładowca w Warszawskiej Wyższej Szkole Informatyki. Autor i współautor projektów systemów informatycznych wdrożonych w wielu firmach. Prowadzi autorskie szkolenia pod nazwą Akademia Baz Danych organizowane przez WWSI. Opiekun kół naukowych studentów. Konsultant w zakresie wdrożeń rozwiązań Business Intelligence. Interesuje się historią.



Jakub Radoszewski, doktorant informatyki w Instytucie Informatyki Uniwersytetu Warszawskiego, kierownik Jury Olimpiady Informatycznej, autor dziesiątków zadań olimpijskich, redaktor czasopisma *Delta*, opiekun witryny Młodzieżowej Akademii Informatycznej, trener reprezentacji Uniwersytetu Warszawskiego w konkursach programistycznych.



Maciej M. Sysło, profesor matematyki, informatyk. Zainteresowania w matematyczne i informatyczne: teoria grafów, kombinatoryka, matematyka dyskretna, optymalizacja kombinatoryczna, edukacja informatyczna – autor ponad 150 prac naukowych.

Od ponad ćwierćwiecza zaangażowany w rozwijanie edukacji informatycznej w szkołach i w uczelniach. Współzałożyciel krajowej olimpiady informatycznej, powołał również do życia międzynarodowy konkurs informatyczny „Bóbr”. W obszarze edukacji informatycznej, autor koncepcji edukacyjnych, programów nauczania, podręczników dla uczniów i poradników dla nauczycieli na wszystkich etapach edukacyjnych. Autor książek z algorytmiki: *Algorytmy optymalizacji dyskretnej z programami w języku Pascal* (pierwsze wydania po angielsku), *Algorytmy, Piramidy, szyszki i inne konstrukcje algorytmiczne*. Współautor oprogramowania edukacyjnego: Pakiet EI, Pakiet TI, e-podręcznik. Wykładowca w wielu uczelniach w kraju i za granicą. Zaangażowany w podnoszenie poziomu nauczania informatyki w szkołach i zwiększenia roli tego przedmiot w kształceniu. Koordynator merytoryczny w projekcie Informatyka +. Ekspert w instytucjach rządowych i niepublicznych. Przedstawiciel Polski w Komitecie TC 3 on Education, IFIP. Organizator konferencji edukacyjnych – „Informatyka w Szkole” (przez 20 lat), ISSEP 2008, WCCE 2013. Popularyzator powszechnej edukacji informatycznej i w zakresie technologii informatyko-komunikacyjnej. Wyróżniony nagrodami: NOT (1968), im Hugona Steinhausa (PTM,

1986), ministra edukacji narodowej (1993, 1998, 2002) oraz nagrodą im. Marka Cara (2010). Interesuje się historią informatyki, kolekcjonuje mechaniczne urządzenia i maszyny do liczenia oraz maszyny do pisania. <http://mmsyslo.pl/>



Katarzyna Śliwińska ukończyła z wyróżnieniem studia na Wydziale Pedagogiki i Psychologii UKW Bydgoszcz, kierunek Psychologia, specjalność Psychologia kliniczna, oraz studia na Wydziale Nauk Historycznych UMK w Toruniu, kierunek Historia, specjalność Historia społeczna. Absolwentka studiów podyplomowych i kursów doszkalających: Psychologia transportu i bezpieczeństwa ruchu drogowego na SWPS w Warszawie i kurs Terapii poznawczo-behawioralna dzieci i młodzieży w CTPB w Warszawie. Obecnie pracuje jako psycholog szkolny w Zespole Szkół UMK Gimnazjum i Liceum Ak-

demickiego w Toruniu, współpracuje z UMK w Toruniu w ramach badań naukowych z zakresu psychologii zdolności, współautorka publikacji z tego obszaru.



Człowiek – najlepsza inwestycja

ISBN 978-83-921270-3-1