

Krzysztof Loryś
Zygmunt Mazur
Rafał Nowak

konsultacja
Tomasz Greczyło
Maria Szczówka- Nowak

INFORMATYKA

**INNOWACYJNY PROGRAM WSPIERANIA UZDOLNIEN
W ZAKRESIE NAUK
MATEMATYCZNO-PRZYRODNICZYCH**

Opracowanie
Dolnośląska Szkoła Wyższa
na zlecenie Fundacji Edukacji Międzynarodowej

Krzysztof Loryś
Zygmunt Mazur
Rafał Nowak

INFORMATYKA

INNOWACYJNY PROGRAM WSPIERANIA UZDOLNIENÍ
W ZAKRESIE NAUK MATEMATYCZNO-PRZYRODNICZYCH

konsultacja
Tomasz Greczyło
Maria Szecówka-Nowak

Nakładca:
Fundacja Edukacji Międzynarodowej
ul. Zielińskiego 38
53-534 Wrocław
tel. + 48 71 782 26 27
faks +48 71 782 26 20
www.fem.org.pl

Wydawca:
DRUKARNIA KiD s.c.
A. Kisielnicki, P. Dąbkowski
ul. Sołtysowicka 26A
51-168 Wrocław
tel. +48 71 325 00 37

Publikacja powstała w ramach projektu "Szlifowanie diamentów - innowacyjne programy wsparcia uczniów uzdolnionych w zakresie nauk matematycznych i przyrodniczych"

Projekt graficzny:
Agencja Reklamowa Times

Wrocław 2013

Nakład: 1500 egz.

ISBN 978-83-63377-32-8



Szlifowanie
diamentów

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Publikacja współfinansowana przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

EGZEMPLARZ BEZPŁATNY

Spis treści

1. Wstęp	5
2. Informacja o autorach	5
3. Ogólna charakterystyka programu	6
4. Cele kształcenia	7
4.1. Cele ogólne	7
4.1.1 Wykraczające poza podstawę programową	8
4.1.2 Wynikające z diagnozy barier społecznych w dostępie do studiów wyższych	8
4.2. Cele wychowawcze	8
4.3. Cele szczegółowe	9
4.3.1 W zakresie wiadomości i umiejętności:	9
4.3.2 W zakresie aktywności intelektualnej:	9
4.3.3 W zakresie rozwijania uzdolnień:	9
5. Materiał nauczania	10
5.1. Treści nauczania	10
5.2. Zakres tematyczny	11
5.3. Grupa I	11
5.4. Grupa II	13
6. Procedury osiągnięcia celów	15
6.1. Rozwiązywanie problemów	16
6.2. Sposoby i techniki pracy	16
6.3. Przykładowy scenariusz zajęć	18
7. Warunki realizacji programu	20
7.1. Odbiorcy programu	20
7.2. Proponowany podział godzin	21
7.3. Liczebność grupy	21
7.4. Środki dydaktyczne	21
7.5. Kwalifikacje i kompetencje nauczyciela	22
7.6. Rekrutacja uczniów	22
7.7. Literatura pomocnicza dla ucznia – książki i materiały internetowe	24
8. Oczekiwane osiągnięcia ucznia	24
9. Monitorowanie osiągnięć uczestników	25
9.1. Metody sprawdzania wiedzy, umiejętności i postaw	25
9.2. Przykładowe narzędzia ewaluacji	26
9.3. Informacja zwrotna dla uczestników	28
10. Bibliografia	28

1. Wstęp

Informatyka to niewątpliwie najmłodsza spośród dyscyplin zaliczanych do grupy przedmiotów matematyczno-przyrodniczych i już choćby z tego powodu wymagająca szczególnych warunków i metod wspierania uczniów uzdolnionych informatycznie. Niniejszy program jest propozycją podstaw programu do prowadzenia zajęć, których jednym z głównych celów jest rozwijanie zainteresowania naukami matematyczno-przyrodniczymi, a w szczególności informatyką. Program jest tak skonstruowany by zajęcia były ciekawe i inspirujące dla uczniów szkół podstawowych, gimnazjalnych i ponadgimnazjalnych. Obejmuje między innymi naukę rozwiązywania problemów algorytmicznych przy użyciu komputera, co jest podstawą niemalże każdego działania. Kluczowym elementem jest zatem nauka i doskonalenie umiejętności programowania oraz twórczego podejścia do rozwiązywania problemów. Równoległe będą prowadzone działania wychowawcze i ogólnorozwojowe pozostające w korelacji ze wsparciem psychologiczno-pedagogicznym ucznia zdolnego.

Niniejszy program jest efektem prac wdrożeniowych. Został zmieniony w stosunku do wersji pierwotnej, która zakładała głównie zajęcia z algorytmów i struktur danych z równoległą nauką języka C++ i programowaniem na maszynie RAM¹. Planowane było przeprowadzanie konkursów i sparingów z programowania podczas każdej sesji oraz nagradzanie zwycięzców. Tak pomyślany program zakładał doskonalenie głównie z algorytmiki, z wybranymi elementami języka C++ i symulatorem maszyny RAM. Jednak mając na uczelni specjalistów z różnych obszarów informatyki, którzy byli chętni do poprowadzenia zajęć i podzielenia się swoją wiedzą i doświadczeniem, uznaliśmy za celowe zmniejszenie liczby zajęć z algorytmiki i zapoznanie uczniów z zagadnieniami współczesnej informatyki (np. kryptografii, modelowania danych, projektowania baz danych, programowania robotów, sieci neuronowych i algorytmów genetycznych).

Integralną częścią przedstawionego opracowania jest *Innowacyjny program wsparcia psychologiczno-pedagogicznego uczniów uzdolnionych, ich rodziców i nauczycieli* (przygotowany w odrębnym dokumencie), który należy realizować jednocześnie z niniejszym programem.

2. Informacja o autorach

dr hab. Krzysztof Loryś, Prof. UW.

Kierownik Zakładu Złożoności Obliczeniowej i Analizy Algorytmów w Instytucie Informatyki Uniwersytetu Wrocławskiego. Od wielu lat organizuje i koordynuje współpracę Instytutu z gimnazjami i szkołami ponadgimnazjalnymi w zakresie nauczania informatyki. Organizator warsztatów

¹ Program symulujący bardzo prosty model komputera.

informatycznych, obozów naukowych dla młodzieży szkolnej, półkolonii informatycznych, konkursów i sparingów w programowaniu. Społecznie uczy w XIV LO. Członek komitetu głównego Olimpiady Informatycznej, współorganizator Wrocławskiego Portalu Informatycznego.

dr hab. Zygmunt Mazur, Prof. PWR.

Kierownik Zakładu Systemów Baz Danych w Instytucie Informatyki Politechniki Wrocławskiej. Prodziekan Wydziału Informatyki i Zarządzania Politechniki Wrocławskiej w latach 1999-2005 i 2012-2016. Dyrektor Policealnego Studium Informatyki dla Dorosłych w latach 1992-2005. Organizator i przewodniczący jury Ogólnopolskiego Konkursu Polskiego Towarzystwa Informatycznego na najlepszą pracę magisterską z informatyki (organizowanego od 1984 roku). Wieloletni prezes Dolnośląskiego Oddziału PTI, obecnie wiceprezes. Członek Komitetu redakcyjnego miesięcznika „Delta”. Członek Rady Nagrody konkursu im. Witolda Lipskiego dla młodych naukowców za dorobek naukowy w dziedzinie informatyki i jej zastosowań.

dr Rafał Nowak

Adiunkt w Instytucie Informatyki Uniwersytetu Wrocławskiego. Od 2004 roku zatrudniony jako nauczyciel informatyki w XIV LO we Wrocławiu, w którym prowadzi zajęcia według autorskiego programu nauczania informatyki w szkołach gimnazjalnych i ponadgimnazjalnych. Organizator wielu obozów naukowych dla młodzieży szkolnej, konkursów i sparingów w programowaniu. Prowadzi także warsztaty informatyczne dla uczniów Gimnazjum nr 49. Wychowawca wielu finalistów Olimpiady Informatycznej oraz Olimpiady Informatycznej Gimnazjalistów.

3. Ogólna charakterystyka programu

Program wsparcia uczniów uzdolnionych w zakresie informatyki przeznaczony jest dla dwóch grup wiekowych:

- grupa I – obejmująca uczniów z klas V i VI ze szkoły podstawowej oraz z klas I i II ze szkoły gimnazjalnej,
- grupa II – obejmująca uczniów z klasy III ze szkoły gimnazjalnej oraz z klas I i II szkoły ponadgimnazjalnej.

Ze względu na znaczne zróżnicowanie predyspozycji w zakresie informatyki uczniów z każdego z poziomów nauczania szkolnego oraz w obrębie danej grupy wiekowej (uczniowie są z różnych szkół, z różnych miejscowości) przygotowanie oferty wsparcia jest niezwykle wyzwaniem i wymaga przygotowania wielu nowych materiałów oraz narzędzi edukacyjnych. Niniejszy program to próba sprostania temu zadaniu.

W programie zaakcentowane są także psychologiczno-pedagogiczne aspekty rozwoju uczniów zdolnych. Zaproponowane oddziaływania mają na

celu wspieranie rozwoju w sferze poznawczej, emocjonalnej, motywacyjnej i społecznej ułatwiając w ten sposób uczniom dążenie do samorealizacji.

Podczas spotkań z uczniami szkół podstawowych, gimnazjalnych i ponadgimnazjalnych można zaobserwować ogromny potencjał jaki drzemie w uczniach uzdolnionych informatycznie do przyswajania wiedzy i umiejętności wykraczających poza podstawę programową nauczania szkolnego. Aby lepiej przygotować takie osoby do podjęcia studiów informatycznych, samodzielnego rozwijania zainteresowań i podejmowania różnych aktywności (np. uczęszczania na kursy informatyczne, organizowania informatycznych kół zainteresowań, zdobywania certyfikatów, podejmowania własnych działalności badawczych itp.), konieczne jest współdziałanie uczelni i szkół.

4. Cele kształcenia

Szczególny rodzaj zajęć, spotkania z uczniami podczas sześciu sesji w określonym wymiarze godzin oraz skład grup uczniowskich (uczestnicy zajęć pochodzą z różnych klas i szkół, z różnych miejscowości, są w różnym wieku) wymaga szczególnego doboru celów kształcenia: ogólnych, szczegółowych i wychowawczych.

Z obserwacji wynika, że uczeń zdolny nie zawsze jest uczniem idealnym – pomimo ogromnych możliwości nie zawsze wykonuje wszystkie zadania, może sprawiać trudności wychowawcze, mieć różne problemy wynikające z jego cech osobowościowych i niezadowolonia ze swoich osiągnięć, nie radzić sobie z komunikacją w grupie itd.

Określenie celów kształcenia pozwala wyznaczyć kierunki zmian, jakie chcemy osiągnąć podczas zajęć. Cele ogólne określają zamiar i misję projektu. Są trudne do zmierzenia i oceny, czy zostały osiągnięte. Cele szczegółowe precyzyjniej określają cele ogólne.

Cele wychowawcze przedstawiają wizję pożądaných cechy osobowości i zachowania ucznia uzdolnionego.

4.1. Cele ogólne

Ukierunkowanie i stymulowanie rozwoju intelektualnego uczestników zajęć – uczniów ponadprzeciętnie uzdolnionych z informatyki.

Prezentowanie informatyki jako dyscypliny szczególnie ważnej w kontekście wyboru ścisłego kierunku studiów wyższych.

Wyrównywanie szans edukacyjnych uczniów uzdolnionych informatycznie pochodzących z różnych środowisk oraz obu płci.

Integracja różnych środowisk wokół problematyki wspierania uzdolnień uczniów.

4.1.1 Wykraczające poza podstawę programową

Doskonalenie umiejętności rozwiązywania problemów algorytmicznych przy użyciu komputera.

Umiejętność zapisywania algorytmów jako programów w imperatywnych językach programowania wyższego rzędu programowania np. w języku C++ czy Java (a dokładniej – w pewnych podzbiorach tych języków).

Nauczenie podstawowych metod konstrukcji algorytmów.

Zaznajomienie z wieloma klasycznymi algorytmami.

Zapoznanie z różnymi zagadnieniami współczesnej informatyki (np. kryptografia, sieci neuronowe, modelowanie danych, projektowanie baz danych).

4.1.2 Wynikające z diagnozy barier społecznych w dostępie do studiów wyższych

Rozpoznawanie mocnych stron ucznia i ich wzmacnianie.

Stwarzanie warunków, w których uczeń poznaje i poszerza swoje możliwości, zainteresowania i predyspozycje.

Rozwój zainteresowań i refleksyjna praca nad własnym potencjałem aspiracjami.

Przeciwdziałanie funkcjonowaniu stereotypu roli zawodowej kobiet.

4.2. Cele wychowawcze

Budowanie poczucia własnej wartości, akceptacji siebie oraz racjonalnej oceny możliwości.

Kształtowanie i wzmacnianie refleksyjności oraz pobudzanie wyobraźni.

Stymulowanie autokreacji, stwarzanie możliwości samopoznania; umożliwienie samooceny własnych celów poznawczych i szacowania możliwości ich realizacji.

Wdrażanie do samodzielności poprzez wspieranie tworzenia własnych koncepcji, pobudzanie kreatywności i sumienną pracę.

Doskonalenie umiejętności współpracy, wymiany myśli i poglądów oraz kultury dyskusji.

Rozwijanie kompetencji społecznych uczniów i umiejętności współpracy w zespole.

Eliminowanie uprzedzeń i skłonności do powierzchownego oceniania; rozwijanie szacunku dla odrębności i indywidualności innych osób.

Rozwijanie umiejętności radzenia sobie z porażką, przyjmowania uwag i krytyki, jako elementu niezbędnego do rozwoju, postępu i doskonalenia.

Stwarzanie warunków do kontaktu z kulturą wysoką i przygotowanie do jej odbioru.

Przygotowanie uczniów do samodoskonalenia i samodzielnego poszerzania swojej wiedzy informatycznej.

Rozbudzanie aspiracji edukacyjnych.

4.3. Cele szczegółowe

Cele szczegółowe określone są z punktu widzenia osoby prowadzącej zajęcia w zakresie wiedzy, umiejętności, aktywności intelektualnej i rozwijania uzdolnień. W przypadku uczniów uzdolnionych informatycznie celem jest przekazanie im wiedzy zarówno z zakresu podstaw informatyki jak i najnowszych trendów, gdyż to pozwala spojrzeć całościowo na rozważane zagadnienia. Jednak wszelkie działania prowadzących zmierzają do pobudzenia ich kreatywności, myślenia twórczego, koncepcyjnego, nabywania i rozwijania umiejętności pracy w zespole z jednoczesnym zachowaniem umiejętności samodzielnego myślenia i rozwiązywania problemów.

4.3.1 W zakresie wiadomości i umiejętności:

1. Przystwojenie wybranych pojęć i opanowanie wybranych umiejętności informatycznych.
2. Opanowanie treści i zdobycie kompetencji dodatkowych pozwalających na indywidualny rozwój uzdolnień.
3. Zdobycie umiejętności przydatnych w życiu codziennym oraz służących poszerzaniu zainteresowań.
4. Poznanie różnorodnych technik i metod poszukiwania, przetwarzania i tworzenia nowych informacji.
5. Opanowanie umiejętności planowania, organizowania i ewaluacji własnej pracy.

4.3.2 W zakresie aktywności intelektualnej:

1. Posługiwanie się językami, symboliką oraz regułami wnioskowania charakterystycznymi dla informatyki.
2. Stosowanie algorytmów.
3. Tworzenie algorytmów.
4. Analizowanie poprawności i efektywności algorytmów.
5. Tworzenie programów.
6. Definiowanie parametrów programu.
7. Tworzenie strategii.
8. Identyfikowanie problemów do rozwiązania.
9. Przedstawianie różnych rozwiązań problemów, ich analizowanie i porównywanie.
10. Weryfikowanie rozwiązań.

4.3.3 W zakresie rozwijania uzdolnień:

1. Kształcenie myślenia koncepcyjnego i kreatywności.
2. Nabycie biegłości w posługiwaniu się procedurami charakterystycznymi dla informatyki.

3. Wyształcenie umiejętności całościowego spojrzenia na prowadzony proces rozumowania.
4. Wdrażanie do samodzielności, zwłaszcza w rozwiązywaniu problemów.

5. Materiał nauczania

Materiał nauczania dostosowany jest do każdej z dwóch wyodrębnionych grup wiekowych uczestników. Proponowany zakres tematyczny wzbogać o pojęcia, problemy i metody wykraczające poza zakres informatyki szkolnej. W przypadku jednakowo brzmiących tematów zajęć dla obu grup, np. *Wstęp do baz danych*, przekazywane treści, tempo zajęć, omawiane przykłady i zadania do rozwiązania powinny być dostosowane do danej grupy wiekowej oraz do możliwości uczestników zajęć.

Dla rozbudzania zainteresowań uczniów oraz poszerzania ich wiedzy proponuje się przeprowadzać w każdej sesji zajęcia przybliżające wybrany ważny dział informatyki, np. z kryptografii, algorytmów genetycznych, sieci neuronowych, grafiki i animacji komputerowej, wyszukiwania informacji. Ich dobór jest dowolny – w każdej dziedzinie należy wskazywać na istotę znajomości zagadnień podstaw informatyki, algorytmiki, złożoności obliczeniowej algorytmów, podstaw teoretycznych zagadnień, aktualny stan wiedzy, prowadzone badania i zastosowania, problemy i kierunki rozwoju.

5.1. Treści nauczania

Treści nauczania podzielono tematycznie na dwie części, dla każdej z grup wiekowych uczestników. Tematy niektórych zajęć mogą być identyczne w obu grupach, ale poziom ich realizacji winien być dopasowany zaawansowaniem do grupy wiekowej. W każdej grupie wyszczególniono 6 sesji tematycznych po 40 godzin każda.

Grupa I

Uczniowie klas V i VI szkoły podstawowej oraz klas I i II gimnazjum

1. *Informatyka – informacja i algorytmika.*
2. *Internet, bazy danych i obliczenia komputerowe.*
3. *Złożone struktury danych, modelowanie danych i systemów.*
4. *Wybrane zagadnienia współczesnej informatyki.*
5. *Grafy, bazy danych, języki opisu stron WWW.*
6. *Grafika, tworzenie gier, dokumentowanie.*

Grupa II

Uczniowie klasy III gimnazjum oraz klas I i II szkoły ponadgimnazjalnej

1. *Informatyka – informacja i algorytmika.*
2. *Internet, bazy danych i obliczenia komputerowe.*
3. *Złożone struktury danych, modelowanie danych i systemów.*
4. *Wybrane zagadnienia współczesnej informatyki.*

5. *Grafy oraz języki programowania, skryptowe i opisu stron WWW.*
6. *Grafika, tworzenie gier, dokumentowanie.*

5.2. Zakres tematyczny

W ustaleniu liczby godzin poszczególnych zajęć należy kierować się wyczućciem i doświadczeniem. W nawiasach umieszczono orientacyjną liczbę godzin związanych z wybraną tematyką. W zależności od grupy i zainteresowania uczestników oraz ich opinii przekazywanych w ankietach ewaluacyjnych przeprowadzanych po każdej sesji, układ zajęć i wymiar planowanych godzin, w miarę możliwości, potrzeby i rozsądku, może i powinien być modyfikowany.

Głównym celem zajęć jest rozbudzenie i rozwijanie zainteresowań, pobudzania do logicznego, algorytmicznego i analitycznego myślenia, wskazanie różnorodności i bogactwa tematyki w zakresie nauki określanej jednym określeniem „informatyka”. Nie zakłada się natomiast, że uczeń ma opanować pewien narzucony materiał.

Planując zajęcia należy mieć na uwadze, że poziom przygotowania informatycznego, wstępnie określanego przez samych uczniów najczęściej jako wysoki, w rzeczywistości jest bardzo różny. Uczniowie pochodzą z różnych szkół i środowisk i być może wcześniej nie mieli możliwości konfrontacji swojej wiedzy z innymi uzdolnionymi osobami w tym samym wieku. Podczas zajęć można zaobserwować rzeczywisty poziom wiedzy i umiejętności poszczególnych osób, zainteresowania określoną tematyką oraz zaangażowania w wykonywanie zadań.

Układając harmonogram zajęć należy zadbać o różnorodność zajęć w odniesieniu do takich aspektów jak: treść, forma, prowadzący zajęcia. Pięć kolejnych dni zajęć z informatyki po 8 godzin dziennie i dodatkowo zajęcia z psychologami są dla młodzieży szkolnej wyczerpujące, gdyż materiał jest na ogół nowy i wymagający głębszej refleksji. Dobierając tematykę i rodzaj zajęć można oddziaływać na koncentrację uwagi uczniów w taki sposób, żeby utrzymywać ją na optymalnym poziomie – dostosowanym do wymagań zadania. Duża różnorodność tematyki zajęć stwarza większe prawdopodobieństwo, że każdy z uczestników znajdzie zagadnienia interesujące go i zachęcające do dalszego rozwijania tych zainteresowań.

5.3. Grupa I

Sesja I – grupa I

Informatyka – informacja i algorytmika

- Informacja, pozycyjne systemy liczbowe (8h)
- Sterowanie a programowanie, schematy blokowe (8h)
- Programowanie robotów Mindstorms NXT w LabView (8h)
- Elementy algorytmiki i złożoność obliczeniowa (8h)

- Wybrane algorytmy wykorzystywane w badaniach kryminalistycznych (8h)

Sesja II – grupa I

Internet, bazy danych i obliczenia komputerowe

- Internet i wyszukiwarki internetowe (10h)
- Obliczenia i symulacje w środowisku Matlab i w Excelu (12h)
- Wprowadzenie do baz danych (4h)
- Algorytmy iteracyjne (6h)
- Wprowadzenie do kryptografii (4h)
- Programowanie w LabView (4h)

Sesja III – grupa I

Złożone struktury danych, modelowanie danych i systemów

- Złożone struktury danych – tablica, stos, kolejka (8h)
- Algebra Boole'a (2h)
- Podstawy relacyjnych baz danych (6h)
- Podstawy SQL (2h)
- Skład tekstu w systemie LaTeX (4h)
- Modelowanie systemów w Matlab (6h)
- Systemy sterowania w technice, medycynie i sporcie (4h)
- Podstawy kryptografii (8h)

Sesja IV – grupa I

Wybrane zagadnienia współczesnej informatyki

- Modelowanie danych (4h)
- Modelowanie i symulacja systemów (4h)
- Metody optymalizacji numerycznej (2h)
- Algorytmy sortowania i zalewowy (implementacja i wizualizacja najkrótszej ścieżki do wyjścia) (8h)
- Sieci neuronowe (4h)
- Teoria chaosu (2h)
- Rozpoznawanie obiektów (4h)
- Cyfrowe przetwarzanie obrazów (8h)
- Rozszerzona rzeczywistość w systemach mobilnych (2h)

Sesja V – grupa I

Grafy, bazy danych, języki opisu stron WWW

- Grafy – droga Eulera, najkrótsza ścieżka, minimalne drzewo rozpinające (6h)
- Myślenie obrazowe w matematyce (5h)
- Algorytmy genetyczne i fraktale (10h)
- Podstawy języka HTML5 (8h)

- Projektowanie i implementacja aplikacji bazodanowych (7h)
- Automatyczna identyfikacja obiektów (4h)

Sesja VI – grupa I

Grafika, tworzenie gier, dokumentowanie

- Programowanie gier – tworzenie GUI, obsługa myszki i klawiatury w Javie (10h)
- Internetowe gry komputerowe – grafika, interakcja i fabuła, dźwięk i synchronizacja (8h)
- Metody uczenia sieci neuronowych (2h)
- L systemy i grafika fraktalna (2h)
- Przetwarzanie obrazu i przekształcenia geometryczne w Matlab (8h)
- Wyszukiwanie informacji (2h)
- Zaawansowana analiza danych (2h)
- Dokumentowanie – redagowanie tekstów, raportowanie, prezentacje multimedialne (6h)

5.4. Grupa II

Sesja I – grupa II

Informatyka – informacja i algorytmika

- Informacja, sterowanie a programowanie, schematy blokowe (8h)
- Programowanie robotów Mindstorms NXT w LabView (10h)
- Elementy algorytmiki i złożoność obliczeniowa (4h)
- Technika „dziel i zwyciężaj” (4h)
- Sesja programistyczna (4h)
- Algorytmy całkowania numerycznego (4h)
- Wybrane algorytmy wykorzystywane w badaniach kryminalistycznych (6h)

Sesja II – grupa II

- Internet, bazy danych i obliczenia komputerowe
- Internet i wyszukiwarki internetowe (10h)
- Obliczenia i symulacje w środowisku Matlab i w Excelu (10h)
- Rozwiązywanie problemów za pomocą programowania dynamicznego (4h)
- Podstawy baz danych. Relacyjne bazy danych (6h)
- Wprowadzenie do kryptografii (4h)
- Programowanie w LabView (6h)

Sesja III – grupa II

Złożone struktury danych, modelowanie danych i systemów

- Liniowe struktury danych (4h)
- Wykorzystanie metod sortowania do rozwiązywania problemów (2h)

- Algebra Boole'a (2h)
- Projektowanie i implementacja prostej aplikacji bazodanowej (6h)
- Podstawy SQL (2h)
- Skład tekstu w systemie LaTeX (4h)
- Modelowanie systemów, proste algorytmy sterowania i obliczenia statystyczne w MatLab (8h)
- Systemy sterowania w technice, medycynie i sporcie (4h)
- Podstawy kryptografii (8h)

Sesja IV – grupa II

Wybrane zagadnienia współczesnej informatyki

- Modelowanie danych (4h)
- Modelowanie i symulacja systemów (4h)
- Metody optymalizacji numerycznej (2h)
- Algorytmy sortowania (2h)
- Drzewa binarne (8h)
- Sieci neuronowe (4h)
- Teoria chaosu (2h)
- Rozpoznawanie obiektów (4h)
- Cyfrowe przetwarzanie obrazów (8h)
- Rozszerzona rzeczywistość w systemach mobilnych (2h)

Sesja V – grupa II

Grafy oraz języki programowania, skryptowe i opisu stron WWW

- Grafy – droga Eulera, najkrótsza ścieżka, minimalne drzewo rozpinające (6h)
- Myślenie obrazowe w matematyce (3h)
- Algorytmy genetyczne i fraktale (10h)
- Skład tekstu w systemie LaTeX (2h)
- Podstawy języka HTML5 (6h)
- Języki skryptowe w aplikacjach internetowych – ECMA-262 (7h)
- Modelowanie danych (2h)
- Automatyczna identyfikacja obiektów (4h)

Sesja VI – grupa II

Grafika, tworzenie gier, dokumentowanie

- Programowanie gier – tworzenie GUI, obsługa myszki i klawiatury w Javie (6h)
- Internetowe gry komputerowe – grafika, interakcja i fabuła, dźwięk i synchronizacja (6h)
- O wymiarach i czasoprzestrzeni (2h)
- Metody uczenia sieci neuronowych (2h)
- L systemy i grafika fraktalna (2h)

- Szereg Fouriera, przetwarzanie obrazu i przekształcenia geometryczne w Matlab (6h)
- Rozwiązywanie problemów metodą zmiatania (4h)
- Ciekawe struktury danych i rozwiązywanie wybranych problemów algorytmicznych (4h)
- Wyszukiwanie informacji (4h)
- Zaawansowana analiza danych (2h)
- Dokumentowanie – redagowanie tekstów, raportowanie, prezentacje multimedialne (4h)

6. Procedury osiągnięcia celów

Podczas każdej sesji jest zaplanowana nauka programowania, którą można określić jako naukę „sterowaną” algorytmiką. Oznacza to, że w każdej fazie nauki przed uczniami będą stawiane problemy, będące punktem wyjściowych rozważań. Będą one tak dobrane, by ich rozwiązania dały się zaprogramować przy użyciu poznanych już elementów języka programowania bądź uzasadniały wprowadzenie kolejnych elementów. W ten sposób nacisk położony jest na rozwój umiejętności algorytmicznego myślenia, a nauka języków programowania jest związana z zapisem algorytmu w wybranym języku programowania i niejako odbywa się przy okazji, nie jest zaś celem nadrzędnym zajęć. Podczas zajęć należy umożliwić uczniom samodzielne dochodzenie do rozwiązania problemu, analizę i porównanie różnych rozwiązań, zwięzłość i czytelność zapisu algorytmu, testowanie rozwiązań.

Przyjęcie takiego modelu pracy pozwoli zrealizować wszystkie cele szczegółowe i wychowawcze programu. Przy czym poszczególne cele wychowawcze zostaną osiągnięte nie poprzez pojedyncze techniki pracy z uczniem czy realizację poszczególnych tematów, ale są efektem udziału ucznia w całym programie wsparcia. Realizacji celów wychowawczych zorientowanych na kształcenie kompetencji społecznych takich jak budowanie poczucia własnej wartości, akceptacji siebie oraz racjonalnej oceny możliwości, stymulowanie autokreacji, stwarzanie możliwości samopoznania; umożliwienie samooceny własnych celów poznawczych i szacowania możliwości ich realizacji, wdrażanie do samodzielności poprzez wspieranie tworzenia własnych koncepcji, pobudzanie kreatywności i sumienną pracę, doskonalenie umiejętności współpracy, wymiany myśli i poglądów oraz kultury dyskusji, rozwijanie kompetencji społecznych uczniów i umiejętności współpracy w zespole, eliminowanie uprzedzeń i skłonności do powierzchownego oceniania; rozwijanie szacunku dla odrębności i indywidualności innych osób, rozwijanie umiejętności radzenia sobie z porażką, przyjmowania uwag i krytyki, jako elementu niezbędnego do rozwoju, postępu i doskonalenia sprzyjają zwłaszcza zaplanowane metody interaktywne, oparte na dyskusji. Przygotowując realizację zajęć zaleca się wykorzystanie wytycznych dotyczących metod pracy z uczniami

zawarte w rozdziale *Wskaźówki do pracy dla członków zespołów przedmiotowych* będącego częścią *Innowacyjnego programu wsparcia psychologiczno-pedagogicznego uczniów uzdolnionych, ich rodziców i nauczycieli*.

Integralną częścią pracy z uczniami podczas sesji jest również udział w aktywnościach kulturalnych, w tym wizyty w instytucjach kultury wysokiej (filharmonia, teatr, opera), które stanowią dopełnienie działań dydaktycznych i stymulują szeroki rozwój społeczno-kulturalny uczestników programu- w myśl przyjętych założeń holistycznego rozwoju i zgodnie z wytycznymi zawartymi w *Innowacyjnym programie wsparcia psychologiczno-pedagogicznego uczniów uzdolnionych, ich rodziców i nauczycieli*. Włączenie tego typu działań wzmocni procesy osiągania celów edukacyjnych i wychowawczych.

6.1. Rozwiązywanie problemów

Podczas realizacji programu stwarzane będą uczniom warunki do systematycznej pracy nad problemami, składającej się z następujących etapów:

- Zrozumienie i analiza problemu.
- Konstrukcja algorytmów rozwiązujących go (często odbywać się to będzie metodą kolejnych ulepszeń, od algorytmu naiwnego o nieakceptowanej dużej złożoności czasowej po algorytmy bardzo wydajne).
- Zapisanie algorytmu w pseudokodzie lub przedstawienie go w postaci schematu blokowego.
- Weryfikacja poprawności algorytmu.
- Analiza złożoności rozwiązania.
- Zapisanie go w języku programowania.
- Uruchomienie programu.
- Umiejętność lokalizowania i poprawiania błędów w programie.
- Testowanie.

6.2. Sposoby i techniki pracy

Cele zajęć będą realizowane z wykorzystaniem następujących form zajęć:

- Zajęcia podczas sześciu pięciodniowych sesji:
 - wykłady,
 - ćwiczenia w pracowni komputerowej, z elementami dyskusji grupowej
- Zdalne konsultacje
- Samodzielne rozwiązywanie zadań w domu i przekazywanie do sprawdzenia prowadzącemu np. za pomocą platformy Moodle.

Po zajęciach z zapisu przykładowych algorytmów za pomocą schematów blokowych i opanowaniu podstawowych konstrukcji takich jak rozgałęzienie warunkowe i pętla, można przejść do programowania w wybranym języku np. Java lub C++. Pierwsze zajęcia w nowym środowisku przeznaczone będą na poznanie podstawowych elementów wybranego języka programowania pozwalających na zapisanie poznanych do tej pory algorytmów. Po przyswojeniu przez uczniów tych elementów, można wprowadzać kolejne – niezbędne

do zapisu omawianych algorytmów i rozwiązania postawionych problemów. Jednocześnie uczniowie zdobędą wiedzę o tym, w jaki sposób korzystać z opisów składni języków programowania dostępnych w Internecie, tak by mogli samodzielnie rozwijać znajomość języków programowania i sięgać do ich internetowych opisów w miarę potrzeby. Równolegle z opanowywaniem umiejętności programistycznych będą przedstawiane wybrane zagadnienia algorytmiczne. Wybór ich został dokonany na podstawie ich przydatności do rozwiązywania problemów algorytmicznych, stopnia ich skomplikowania możliwego do opanowania przez uczniów oraz łatwości implementacyjnej.

Celem zajęć nie było przygotowanie uczniów do olimpiady informatycznej czy konkursów programistycznych. Dlatego organizowanie podczas zajęć zawodów i konkursów raczej nie jest zalecane, gdyż może być dla wielu uczniów bardzo stresujące i demotywujące. Z przeprowadzania zajęć w formie konkursów są zazwyczaj zadowolone osoby wygrywające (może się okazać, że są to zawsze ci sami uczniowie). Natomiast pozostałe osoby narażone są na przeżywanie dyskomfortu wynikającego z ponoszonych porażek – mogą czuć się gorsze i odrzucone, jako te, które nie są chwalone, nie dostają nagród.

Dużo lepszym rozwiązaniem jest wykorzystanie metody projektów, w ramach której uczniowie pracują w grupach, rzadziej indywidualnie, nad określonym zadaniem i prezentują publicznie wyniki swojej pracy poddając ją pod dyskusję. Metoda ta stosowana jest powszechnie w procesie kształcenia na kierunkach informatycznych.

W trakcie realizacji projektu następuje nie tylko przyswajanie nowej wiedzy, ale przede wszystkim efektywne jej wykorzystywanie. Rozwijane są jednocześnie tak pożądane umiejętności i postawy, jak: planowanie działań, współpraca w zespole, poczucie większej wartości i pewności siebie, odpowiedzialność i in.

W odniesieniu do materiału nauczania, przedstawionego w punkcie 5, metodę projektów warto zastosować podczas realizacji dwóch ostatnich sesji (V i VI), kiedy to uczniowie dysponując już odpowiednią wiedzą i umiejętnościami mogą zrealizować projekty w zakresie aplikacji bazodanowych, programowania gier, cyfrowego przetwarzania obrazów itp.

Sesyjny system nauki wspierany i uzupełniany będzie e-learningiem wykorzystującym np. platformę Moodle, w szczególności do publikowania i komentowania zagadnień oraz materiałów omawianych na sesjach, zamieszczania dodatkowych zadań realizowanych przez uczniów pomiędzy sesjami, udzielania porad i wyjaśnień przez prowadzących poszczególne sesje. W Internecie jest wiele informacji i należy uświadomić uczniom, że nie wszystkie są wartościowe i poprawne. Należy wskazać przykładowe portale internetowe, z których można korzystać, opracowane profesjonalnie, przez kompetentne osoby, na przykład:

- www.ii.pwr.wroc.pl/pl/instytut-i-32/dolnoslaskie-zawody-w-programowaniu-zespolowym – portal udostępniający zadania z Zawodów

w Programowaniu Zespołowym organizowanych dla uczniów i studentów przez Instytut Informatyki Wydziału Informatyki i Zarządzania Politechniki Wrocławskiej,

- <http://wazniak.mimuw.edu.pl> – portal Uniwersytetu Warszawskiego udostępniający materiały z przedmiotów informatycznych i umożliwiającą naukę na odległość,
- http://edu.i-lo.tarnow.pl/inf/alg/003_sort/index.php – serwis edukacyjny I LO w Tarnowie,
- www.informatyka.wroc.pl – Wrocławski Portal Informatyczny,
- www.oi.edu.pl – portal olimpiady informatycznej,
- pwi.edu.pl – Polska Wszechnica Informatyczna
- www.portal.pwr.wroc.pl/1811035,241.dhtml – portal Politechniki Wrocławskiej – wykłady z przedmiotu „*Analiza matematyczna*” dostępne na kanale You Tube,
- www.wykop.pl/link/1332207/animacje-algorytmow-sortowania – animacje algorytmów sortowania,
- <http://main.edu.pl> – portal edukacyjny Młodzieżowej Akademii Informatycznej (zawiera kurs z algorytmiki oraz kurs języka Pascal i C++, zadania z olimpiad informatycznych)
- www.kurshtml.edu.pl – internetowy kurs HTML.

6.3. Przykładowy scenariusz zajęć

Zajęcia wykładowo-ćwiczeniowe:

Reprezentowanie algorytmów za pomocą schematów blokowych (2 godziny).

Materiały:

- Komputery z zainstalowanym oprogramowaniem (20 stanowisk)
- Zestaw prezentacyjny (rzutnik multimedialny, laptop podłączony do rzutnika, ekran, wskaźnik)
- Tablica

Metoda pracy:

- dyskusja
- wykład
- praca grupowa
- zajęcia praktyczne
- prezentacja własnej pracy

Przebieg zajęć:

- Przedstawiamy i omawiamy podstawowe elementy schematów blokowych.
- Przedstawiamy schemat blokowy dla zadania polegającego na wczytaniu jednej liczby całkowitej **a**, obliczeniu **b = a+10** i wprowadzaniu

liczb **a** i **b**. Omawiamy rozwiązanie najpierw na użytych symbolach zmiennych a następnie na przykładowych danych. Graficznie przedstawiamy zmienne i ich zmiany wartości podczas wykonywania algorytmu. Omawiamy instrukcję podstawienia.

- Formułujemy zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania sumy dwóch liczb. Uczniowie rysują schematy na kartkach papieru.
- Po rozwiązaniu zadania przez uczniów, wybieramy ochotnika, który przedstawia na tablicy swoje rozwiązanie (tak postępujemy zawsze po każdym zadaniu, które uczniowie wykonują podczas zajęć). Warto zachęcać również uczniów nieśmiałych, którzy być może nie mają odwagi zgłosić się samodzielnie do przedstawienia swojego rozwiązania przed audytorium.
- Prosimy uczestników zajęć o konstruktywną opinię na temat przedstawionego rozwiązania i uzasadnienie swojej oceny.
- Kolejne zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania iloczynu dwóch liczb – zwracamy uwagę na podobieństwo algorytmu i wymaganą niewielką modyfikację w stosunku do poprzedniego zadania.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania ilorazu dwóch liczb – zwracamy uwagę na pozorne podobieństwo algorytmu i wymaganą istotną modyfikację w stosunku do poprzedniego zadania. Wprowadzamy instrukcję warunkową w celu sprawdzenia, czy wartość mianownika jest różna od zera.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm sprawdzania, czy podana liczba całkowita jest dodatnia. Zwracamy uwagę na poprawność zapisu warunku porównania.
- Łączymy uczniów w 2-osobowe grupy i zadajemy zadanie: za pomocą schematu blokowego przedstawić algorytm sprawdzania, czy podana liczba całkowita jest dodatnia, ujemna, czy ma wartość zero.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania sumy trzech liczb.
- Wprowadzamy pojęcie pętli, licznika pętli.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania sumy dziesięciu danych liczb. Jeśli w trakcie prezentacji rozwiązania okaże się, że uczniowie zapisali schematy bez użycia pętli, to polecamy modyfikację schematów i zapis algorytmu z użyciem pętli.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczenia iloczynu 20 liczb. W przedstawionych przez uczniów rozwiązaniach uczniowie powinni pamiętać, że wartość początkowa iloczynu musi wynosić jeden lub być równa wartości pierwszego elementu. Ponadto powinien być uwzględniony warunek zakończenia obliczeń, jeśli element ma wartości zero.

- Zapis schematu blokowego w wybranym oprogramowaniu narzędziowym, np. w Excelu.
- Omówić algorytm Euklidesa obliczania największego wspólnego dzielnika dwóch liczb naturalnych. Przedstawić algorytm za pomocą schematu blokowego w określonym oprogramowaniu narzędziowym.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania liczby elementów dodatnich, ujemnych i równych 0 dla danych n liczb całkowitych.
- Zadanie dla uczniów: za pomocą schematu blokowego przedstawić algorytm obliczania liczby elementów dodatnich i ujemnych dla podawanych liczb całkowitych. Warunkiem zakończenia algorytmu jest podanie liczby o wartości zero.

7. Warunki realizacji programu

Program winien być realizowany w formie zajęć stacjonarnych. Do jego realizacji niezbędne są pracownie komputerowe z dostępem do Internetu. Pożądany jest sprzęt multimedialny np. rzutnik, tablica interaktywna. Wykorzystywane oprogramowanie zależy od tematu zajęć. Podczas zajęć w latach 2011-2013 wykorzystywano m.in.:

- Microsoft Office 2010 PL
- Matlab 2011a
- Octave
- Program do kompresji i dekompresji plików 7-Zip
- Adobe Reader
- Przeglądarka internetowa Mozilla Firefox
- Java
- LabVIEW
- LaTeX.

Wprowadzenie do tematów ze współczesnej informatyki dobrze jest rozpocząć od sformułowania problemu i wskazania etapów rozwoju nauki w danej dziedzinie. Przedstawiając aktualny stan badań, należy wskazać zarówno osiągnięcia jak i trudności oraz problemy dotychczas nie rozwiązane, nad którymi nadal prowadzone są prace badawcze.

Omawiając zagadnienia z informatyki należy podkreślać, gdzie tylko to możliwe, interdyscyplinarność tej nauki i powiązanie z innymi naukami przyrodniczo-matematycznymi, technicznymi i humanistycznymi.

7.1. Odbiorcy programu

Odbiorcami programu są uczniowie ponadprzeciętnie uzdolnieni informatycznie ze szkoły podstawowej (klas V i VI), gimnazjum oraz szkół ponadgimnazjalnych (klas I i II) wyłonieni w drodze przyjętej procedury rekrutacji uwzględniającej diagnozę ich uzdolnień oraz ocenę osiągnięć.

Uczniowie zakwalifikowani do zajęć powinni wykazać się umiejętnością algorytmicznego myślenia i znajomością obsługi komputera. Wśród zadań rekrutacyjnych powinny być zadania dotyczące zapisu algorytmów w różnych postaciach (np. w języku naturalnym, graficznie, pseudokodzie) oraz zadania wykonane na komputerze (np. w edytorze, arkusza kalkulacyjnym). Uczniowie powinni napisać list motywacyjny i napisać w nim m.in. o sobie i swoich zainteresowaniach, dlaczego chcieliby zostać zakwalifikowania na zajęcia, jakie są ich oczekiwania odnośnie zajęć.

7.2. Proponowany podział godzin

Zaplanowane 240 godzin zajęć odbywa się w 6 sesjach po 40 godzin realizowanych w ciągu 6 dni. Każda z 6 sesji obejmuje zajęcia praktyczne, które winny być realizowane w pracowniach uczelni wyższych.

Układ zajęć często podyktowany jest dostępnością sal, niemniej jednak zajęcia powinny być rozłożone w miarę równomiernie, z uwzględnieniem czasu na przerwy i posiłki, najlepiej w godz. 9-18.

Podczas każdej sesji przedmiotowej należy uwzględnić od 2 do 4 godzin warsztatów rozwojowych z psychologami i/lub pedagogami realizowanych w oparciu o *Innowacyjny program wsparcia psychologiczno-pedagogicznego uczniów uzdolnionych, ich rodziców i nauczycieli*. Wymiar i tematykę tych warsztatów rekomenduje dany zespół przedmiotowy na podstawie bieżącej ewaluacji postępów i potrzeb każdej grupy uczniowskiej.

7.3. Liczebność grupy

Grupa 40 uczniów – 20 uczniów w każdej z dwóch podgrup – I podgrupa uczniowie klas V i VI szkoły podstawowej oraz klas I i II gimnazjum, II podgrupa uczniowie klasy III gimnazjum oraz klas I i II szkoły ponadgimnazjalnej.

W przypadku zajęć, podczas których uczniowie pracują w zespołach 2-5 osobowych, dobór grup można pozostawić uczniom, chyba, że zadanie wymaga szczególnego doboru członków zespołu (w takim przypadku kryteria doboru należy przedstawić grupie).

W przypadku zajęć z informatyki wielkość grupy podyktowana jest głównie liczbą komputerów w laboratorium. W wielu przypadkach niezbędna jest praca indywidualna i każdy uczeń powinien mieć zapewnione miejsce przy komputerze. Często jednak można uczniów łączyć w zespoły 2 osobowe, co zawsze przyjmują z radością.

7.4. Środki dydaktyczne

W realizacji poszczególnych sesji wykorzystywane są zarówno tradycyjne jak i nowoczesne środki dydaktyczne:

- wzrokowe – tablica, prezentacje, rysunki, klocki.
- wzrokowo-słuchowe – prezentacje multimedialne
- automatyzujące prace – komputery.

W szczególności intensywnie będą wykorzystywane programy wspierające naukę programowania:

- Oprogramowanie narzędziowe umożliwiające zapis algorytmów w formalizmie schematów blokowych.
- Wszelkie programy wspierające algorytmiczne myślenie. Programy te pozwalają obniżyć stopień abstrakcji wymagany podczas tworzenia algorytmów poprzez wizualizację ich kolejnych kroków. To umożliwia weryfikację własnych decyzji i uczy przewidywania ich skutków.

7.5. Kwalifikacje i kompetencje nauczyciela

Prowadzącymi zajęcia powinni być pracownicy naukowo-dydaktyczni lub dydaktyczni wyższych uczelni cechujący się umiejętnością pracy z uczniami

- w wieku szkolnym. Nauczyciel realizujący zajęcia merytoryczne powinien posiadać:
- wiedzę w zakresie prowadzonego tematu,
- umiejętność precyzyjnego formułowania wypowiedzi,
- umiejętność doboru zakresu tematycznego,
- umiejętność i doświadczenie w doborze przykładów ilustrujących omawiane zagadnienia,
- umiejętność prowokowania uczniów do analizowania zagadnień, samodzielnego rozwiązywania problemów oraz dyskusji nad rozwiązaniami,
- umiejętność formułowania zadań do samodzielnego rozwiązywania,
- wysoki poziom kompetencji społecznych (w kontekście skuteczności w interakcjach i komunikacji z uczniami),
- doświadczenie w pracy z młodzieżą szkolną.

7.6. Rekrutacja uczniów

Informacja o naborze uczniów powinna być umieszczona na określonych stronach WWW i ogólnie dostępna.

Dla kandydatów do obu grup wiekowych należy przygotować ok. 5-7 zadań do samodzielnego wykonania w domu. Rozwiązania zadań powinny być wykonane na komputerze, wydrukowane i przesłane na adres organizatorów zarówno w wersji papierowej jak i w postaci elektronicznej na płytach CD.

Zadania kwalifikacyjne do projektu powinny być z zakresu algorytmów, schematów blokowych, elementów programowania, znajomości oprogramowania narzędziowego. Poziom trudności zadań powinien być zróżnicowany dla obu grup i odpowiednio dobrany do wieku uczniów. Każdy uczeń powinien rozwiązać zadania samodzielnie, bez pomocy rodziców, nauczycieli, rodzeństwa, kolegów. Rozwiązania powinny być przedstawione w czytelnej formie, przedstawiającej tok myślenia a nie tylko ostateczny wynik. Rozwiązanie każdego zadania powinno być przedstawione na osobnej kartce, zawierającej nazwisko, imię i numer zadania – ułatwia to sprawdzanie zadań. W przypadku

braku rozwiązania jakiegoś zadania, uczeń powinien na osobnej podpisanej kartce obok numeru zadania napisać „Brak rozwiązania”.

Przykładowe zadania dla grupy I:

1. W arkuszu kalkulacyjnym opracuj wzór dzienniczka ocen ucznia.
2. W arkuszu kalkulacyjnym opracuj szablon do rozliczania miesięcznego budżetu rodziny.
3. Opracuj prezentację związaną z miejscowością, w której mieszkasz.
4. Przepisz w edytorze tekstowym wybrany tekst np. wiersz lub fragment z podręcznika do matematyki lub chemii. Tekst może zawierać wzory, tabelki, rysunki (długość tekstu ok. 1000 znaków bez spacji).
5. Zaprojektuj zaproszenie na zabawę mikołajkową w Twojej szkole.
6. Opracuj, jako szablon dokumentu, papier firmowy dla Twojej szkoły.
7. Przedstaw na schemacie blokowym algorytm obliczania sumy n liczb.

Przykładowe zadania dla grupy II:

1. W arkuszu kalkulacyjnym opracuj tabliczkę mnożenia w zakresie liczb całkowitych od 1 do 10 z możliwością szybkiego rozszerzenia na większy zakres (wpisz formułę raz a potem ją tylko kopiuj na pozostały obszar). Przedstaw tabliczkę mnożenia na wykresie powierzchniowym.
2. W arkuszu kalkulacyjnym przygotuj odpowiednie dane dotyczące ocen na koniec roku uczniów trzech oddziałów klasowych kończących gimnazjum (w każdej klasie jest nie mniej niż 20 uczniów) a następnie policz liczbę uczniów w każdej klasie ze średnią obliczoną z ocen końcowych powyżej 4,5 (w każdej klasie tych uczniów jest nie mniej niż 15% składu każdej klasy).
3. Opracuj schemat blokowy dla algorytmu obliczania wartości największej dla danych n liczb.
4. Opracuj jako korespondencję seryjną etykiетки adresowe i listy wysyłane do rodziców przez wychowawcę 30-osobowej klasy np. z informacją o liczbie godzin nieobecnych ucznia od początku roku szkolnego.
5. Opracuj schemat blokowy dla algorytmu obliczania sumy liczb położonych nad główną przekątną w tablicy dwuwymiarowej o n wierszach i n kolumnach.

W przypadku projektu zakładającego zajęcia głównie z algorytmiki i nauki programowania, których celem byłoby przygotowanie uczniów do konkursów i olimpiad informatycznych, zadania rekrutacyjne powinny być dotyczyć opracowania programów w wybranym (lub określonym) języku programowania. Ponadto uczniowie przystępujący do rekrutacji powinni być poinformowani o planowanym charakterze zajęć, ponieważ uczeń niezainteresowany wyłącznie programowaniem może być rozczarowany, jeśli 240 godz. zajęć podczas 6 sesji będzie związanych głównie z doskonaleniem umiejętności programowania.

Rozwiązania zadań powinny być przesłane w wersji elektronicznej na CD oraz wydrukowane, w wersji papierowej. Umożliwia to szybsze sprawdzanie prac – włożenie każdej płyty CD i odczytanie jej zawartości trwa o wiele dłużej, niż sprawdzenie rozwiązań wydrukowanych. Z kolei wersja elektroniczna na CD potwierdza umiejętności pracy z komputerem – co prawda minimalne, ale jednak pożądane na zajęciach. Ponadto w przypadku uszkodzenia płyty CD (co się zdarza, jeśli przesyłka jest nieodpowiednio zabezpieczona) jest wersja papierowa, która raczej nie ulega zniszczeniu przez pocztę. Wersje papierowe ułatwiają również porównywanie kilka prac jednocześnie, co jest potrzebne np. w celu ustalenia ich podobieństwa lub wybrania pracy lepszej wówczas, gdy różnice między nimi są minimalne.

Każdy uczeń powinien również napisać list motywacyjny zawierający krótką autoprezentację ucznia oraz uzasadnienie i powody chęci uczestnictwa w zajęciach z informatyki.

7.7. Literatura pomocnicza dla ucznia – książki i materiały internetowe

1. M.M. Sysło, Algorytmy, WSiP,1997.
2. M.M. Sysło, Piramidy, szyszki i inne konstrukcje algorytmiczne, WSiP 1998, <http://mmsyslo.pl/Materialy/Ksiazki-i-podreczniki/Ksiazki/Ksiazka-Piramidy-szyszki-i>.
3. Wirtualny kurs języka HTML, www.kurshtml.edu.pl.
4. Wrocławski Portal Informatyczny, www.informatyka.wroc.pl.
5. Wykłady z informatyki, <http://wazniak.mimuw.edu.pl>.

8. Oczekiwane osiągnięcia ucznia

Oczekiwane osiągnięcia uczniów można ująć w następujących standardach:

A. Sprawne wykonywanie typowych czynności

Standard związany jest z uzyskaniem biegłości w posługiwaniu się narzędziami na takim poziomie, by nie stanowiły dodatkowej trudności przy wykonywaniu czynności wyższego rzędu. Do tych typowo technicznych umiejętności zaliczyć można: umiejętność korzystania z kompilatora wybranego języka programowania (C++, Java) sprawne korzystanie z edytorów tekstu i arkuszy kalkulacyjnych w podstawowym zakresie, sprawne korzystanie z automatycznych programów sprawdzających (np. Spoj), wyszukiwanie niezbędnych informacji na dany temat (w tym zwłaszcza informacji zamieszczonych w Internecie).

B. Umiejętne prowadzenie rozumowania

W standardzie tym dominuje strona techniczna, ale w obrębie całościowego spojrzenia na prowadzony proces i dobieranie pod tym kątem określonych procedur. Standard już tylko nieznacznie wiąże się z treściami i dotyczy

umiejętności: wykorzystywania znanych algorytmów i zabiegów programistycznych, tworzenia przejrzystego i logicznego ciągu działania, wykorzystywania i przetwarzania informacji o różnych formach.

C. Samodzielne rozwiązywanie problemów

Standard dotyczy obszaru aktywności ucznia i charakteryzuje go przede wszystkim zdolność do zmagania się z problemami. Umiejętności w tym standardzie to przede wszystkim: umiejętność zlokalizowania i określenia problemu, określenia strategii jego rozwiązywania, podjęcie adekwatnych działań.

D. Kreatywność

To najwyższy ze szczebli rozwoju ucznia, dlatego jest to niewątpliwie standard najtrudniejszy do osiągnięcia. Stanowi wyzwanie dla nauczyciela, który winien szukać takich zabiegów dydaktycznych, by pozwoliły uczniom na skuteczne podjęcie prób takich działań, jak: znajdowanie nowatorskich rozwiązań, optymalizacja działań.

9. Monitorowanie osiągnięć uczestników

Monitorowanie osiągnięć będzie realizowane poprzez metody, dzięki którym uczeń będzie miał możliwość poddania autorefleksji przebieg procesu rozwiązywania problemów i uczenia się.

Na wszystkich zajęciach należy odnotowywać obecność uczniów. Zaleca się by podczas zajęć uczniowie mieli identyfikatory, co umożliwi prowadzącym ich szybką identyfikację i odnotowywania aktywności podczas zajęć.

9.1. Metody sprawdzania wiedzy, umiejętności i postaw

Problemy często będą sformułowane w postaci pewnej historii „bliskiej życia” tak jak ma to miejsce w przypadku zadań rozwiązywanych na olimpiadach informatycznych.

Nie zaleca się przeprowadzania konkursów, testów, sprawdzianów itp. które wpływałyby stresująco i blokująco na uczniów. Monitorowanie osiągnięć uczniów nie powinno być dla nich obciążające i deprymujące. Sprawdzanie wiedzy powinno być przeprowadzane np. w formie gier lub zabaw, gdzie uczestnikami są grupy uczniów lub indywidualni uczniowie.

Monitorowanie osiągnięć uczestników może być również przeprowadzane w formie testów czy sprawdzianów, ale wyniki nie powinny być podawane do ogólnej wiadomości tylko indywidualnie danym uczniom.

W okresie między sesjami uczniowie mogą rozwiązywać dodatkowe zadania, w celu utrwalenia i pogłębienia wiedzy lub uzupełnienia ewentualnych braków. Zadania powinny być dostępne w ustalony sposób, np. poprzez przeglądarkę internetową na platformie Moodle. Po sprawdzeniu rozwiązań zadań prowadzący powinien przekazać uczniom informację zwrotną na temat przedstawionego rozwiązania.

Wskazane jest również na początku każdej sesji (o ile to możliwe) krótkie nawiązanie do poprzednich zajęć i sprawdzenie, na ile ostatnio omawiany materiał został przez uczniów zrozumiany i opanowany.

9.2. Przykładowe narzędzia ewaluacji

Umiejętności uczniów będą ewaluowane m.in. poprzez zadania, których rozwiązanie jest przedstawiane w postaci programu komputerowego. Będą one miały różny stopień trudności i będą dotyczyły różnych problemów algorytmicznych. Oto przykłady takich zadań.

Przykład 1. (zadanie: Zakochany Żab)

Pan Żab siedzi na kamieniu na środku jeziora. Na innym kamieniu siedzi pani Żaba, którą pan Żab chciałby odwiedzić. Ponieważ Żab chciałby dobrze zaprezentować się przed panią Żabą (co wyklucza możliwość przepłynięcia do jej kamienia), więc przymierza się, by dotrzeć do niej przeskakując po kamieniach. Ten plan może udać się jedynie wtedy, gdy największa odległość między kamieniami na drodze wybranej przez pana Żaba jest mniejsza niż zasięg jego skoku.

Zadanie

Napisz program, który pomoże panu Żabowi stwierdzić, czy jego plan dotarcia suchą nogą do pani Żaby jest realny.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna n (nie większa od 1000) określająca liczbę kamieni wystających ponad powierzchnię jeziora. W każdym z kolejnych n wierszy znajduje się para liczb całkowitych określających współrzędne jednego z kamieni. Pierwsza z tych par określa współrzędne kamienia, na którym siedzi pan Żab, a druga – współrzędne kamienia, na którym siedzi pani Żaba.

Wyjście

W pierwszym i jedynym wierszu wyjścia znajduje się jedna liczba rzeczywista (podana z dokładnością do setnych części) określająca, jak daleko co najmniej pan Żab musi umieć skakać, by móc dotrzeć do pani Żaby, nie zamoczywszy nogi.

Przykład

Dla danych wejściowych

4

0 0

4 4

0 3

2 1

poprawną odpowiedzią jest 3.61

Przykład 2. (Zadanie: Drogi)

W Bajtoci jest n miast. Miasta są połączone jednokierunkowymi drogami. Każda droga łączy tylko dwa miasta i nie przechodzi przez żadne inne. Niestety, nie zawsze z każdego miasta da się dojechać do każdego innego. Król Bajtazar postanowił rozwiązać ten problem. Król ma świadomość, że budowanie nowych dróg jest bardzo kosztowne, a budżet Bajtoci nie jest zbyt zasobny. Dlatego też poprosił Cię o pomoc. Trzeba obliczyć minimalną liczbę jednokierunkowych dróg, które trzeba zbudować, żeby z każdego miasta dało się dojechać do każdego innego miasta.

Zadanie

Napisz program, który:

- wczyta opis istniejącej sieci dróg,
- obliczy minimalną liczbę dróg, które trzeba dobudować tak, aby z każdego miasta w Bajtoci dało się dojechać do każdego innego,
- wypisze wynik.

Wejście

Pierwszy wiersz zawiera dwie liczby całkowite n i m ($2 \leq n \leq 10\,000$, $0 \leq m \leq 100\,000$) oddzielone pojedynczym odstępem i oznaczające, odpowiednio, liczbę miast i liczbę dróg w Bajtoci. Miasta są ponumerowane od 1 do n . W każdym z kolejnych m wierszy znajdują się dwie liczby całkowite oddzielone pojedynczym odstępem. W $i+1$ -szym wierszu znajdują się liczby a_i i b_i ($1 \leq a_i, b_i \leq n$ dla $1 \leq i \leq m$), reprezentują one jednokierunkową drogę prowadzącą z miasta a_i do b_i .

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać dokładnie jedną nieujemną liczbę całkowitą – minimalną liczbę dróg, które trzeba zbudować w Bajtoci tak, aby z każdego miasta dało się dojechać do każdego innego miasta.

Przykład

Dla danych wejściowych:

```
7 11
1 3
3 2
2 1
2 2
3 4
4 5
5 4
3 4
1 6
6 7
7 6
```

poprawną odpowiedzią jest 2.

9.3. Informacja zwrotna dla uczestników

Zadania rozwiązywane przez uczniów będą sprawdzane przez prowadzącego. Po sprawdzeniu prowadzący:

- przekazuje uczniowi informacje dotyczące błędów w rozwiązaniu, ale akcentuje też to, co zostało poprawnie wykonane,
- podaje informacje o liczbie poprawnie wykonanych testów,
- przedstawia aktualny ranking uczniów (sporządzany względem liczby poprawnych rozwiązań).

Wykorzystanie platformy elektronicznej (np. Moodle), znacznie ułatwia kontakt z uczniami, umożliwia umieszczenie dodatkowych materiałów do zajęć (przed lub po zajęciach), komunikację z całą grupą lub z wybranymi osobami, wysyłanie wiadomości, rozwiązań zadań oraz informacji zwrotnych na temat przesłanych rozwiązań. Pozwala obserwować aktywność uczniów poza zajęciami na uczelni. Zamieszczanie list zadań przerabianych na zajęciach oraz innych materiałów z zajęć umożliwia uczniom nieobecny na zajęciach, zapoznanie się z ich tematyką, pobranie materiałów i wykonanie ćwiczeń. Platformy te umożliwiają również organizowanie ścieżek indywidualnych dla ucznia oraz kontakt za pomocą czatu lub forum.

10. Bibliografia

1. Limont W. (2010) Uczeń zdolny. Jak go rozpoznawać i jak z nim pracować. Gdańsk: GWP.
2. Rozporządzenie Ministra Edukacji Narodowej z 23 grudnia 2008 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół (Dz. U. z 2009 r. nr 4, poz. 17).
3. Mikiewicz P., Schemat pracy z uzdolnioną młodzieżą w Centrum Intelpekt. Ramowy model systemu.
4. Mikiewicz P., Krzychała S., Bariery społeczne w edukacji dziewcząt z małych miejscowości na Dolnym Śląsku, które przejawiają uzdolnienia w kierunkach matematyczno–przyrodniczych i zamierzają podjąć studia na tych kierunkach.
5. Gołębiak D. (red.) (2002) Uczenie metodą projektów. Warszawa: WSiP.
6. Janowicz J.: Standardy kształcenia uczniów zdolnych. Matematyka nr 1/2005, s. 19-25.
7. Brzezińska A. (red.) (2005). Psychologiczne portrety człowieka. Praktyczna psychologia rozwojowa. Gdańsk: Gdańskie Wydawnictwo Psychologiczne.

