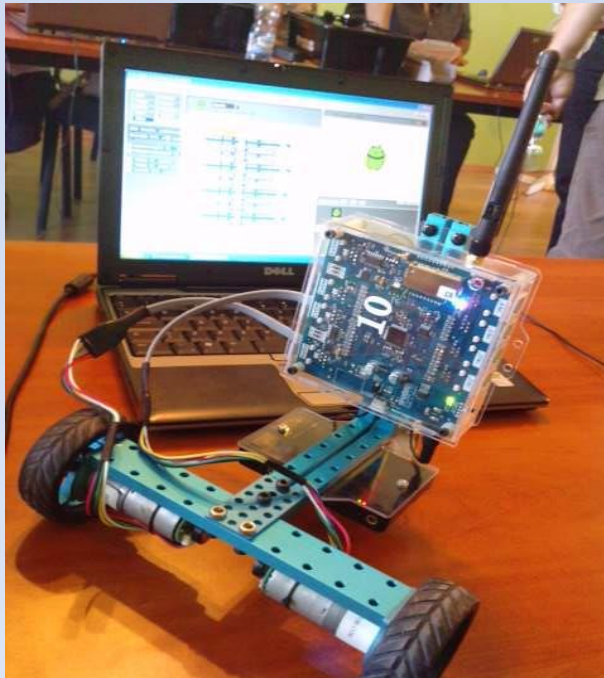


Metody prowadzenia zajęć w perspektywie mechatronicznej



Seminarium dla
woj. łódz. 20-21.03.2014
Klonowiec Stary



mgr inż. Anna Kozłowska



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

ORE
OŚRODEK
ROZWOJU
EDUKACJI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Plan prezentacji

- Różne podejścia do programowania
 - praca samodzielna i grupowa
- Indywidualizacja pracy w zależności od możliwości uczniów
 - praca z uczniem mniej zdolnym
 - praca z uczniem zdolnym
- Metody prowadzenia zajęć
- Sprawdź czy potrafisz

Różne podejścia do programowania

- wprowadzeniem do zagadnienia algorytmiki powinny być sytuacje z życia codziennego ucznia, które można przedstawić w formie schematów blokowych;
- możliwość kilku podejść projektowych, diametralnie różnych, lecz w konsekwencji doprowadzających do oczekiwanego efektu;
- nie jest ważny sposób dochodzenia do założonego celu, ale efekt końcowy;
- analizie może podlegać efektywność przyjętego sposobu realizacji zadania i jego optymalizacja;
- w każdym przypadku ważna jest interpretacja ucznia, uzasadnienie wybranych rozwiązań, zastosowanej metody.

- w trakcie wykonywania przez uczniów ćwiczeń, nauczyciel obserwuje sposób ich realizacji. W przypadku konieczności udzielenia pomocy stara się zmienić tok i kierunek myślenia uczniów naprowadzając ich na alternatywne sposoby, zmierzające do osiągnięcia określonego efektu działania programu;
- misje robotów zawierają elementy rywalizacji (wyścigi, zawody robotów). Takie ujęcie może powodować trudne sytuacje wychowawcze wśród uczniów wynikające np. z kilkukrotnej porażki autora programu.

- Praca samodzielna:

uczniowie (nauczyciel jest tu tylko obserwatorem oraz moderatorem procesu uczenia się uczniów) uczą się odpowiedzialności, podejmowania decyzji, dokonywania samooceny;

- Praca grupowa:

uczniowie rozwijają umiejętności podejmowania decyzji w grupie, rozwiązywania konfliktów, wyrażania własnych opinii, słuchania innych osób, poszukiwania kompromisów, dyskusowania, dokonywania oceny pracy swojej i innych. W działaniach uczniowie również nabywają umiejętności układania harmonogramów, planowania swojej pracy.

Indywidualizacja pracy w zależności od możliwości uczniów

- **Praca z uczniem mniej zdolnym**
 - nauczyciel powinien poznać predyspozycje uczniów w grupie i tak pokierować procesem uczenia się, aby uczeń mający problemy np. z programowaniem mógł zaprezentować inne, posiadane zdolności (np. konstrukcyjne). Ważne jest podejście nauczyciela, które jest ukierunkowane na wzmacnianie silnych stron ucznia przejawiającego trudności w uczeniu się.
 - uczniowie pracują na lekcjach w większości w parach lub zespołach kilku osobowych). Każdy z uczniów będzie mógł zaprezentować swoje silne strony, a efekt pracy całego zespołu będzie „produktem” synergicznym.

Indywidualizacja pracy w zależności od możliwości uczniów

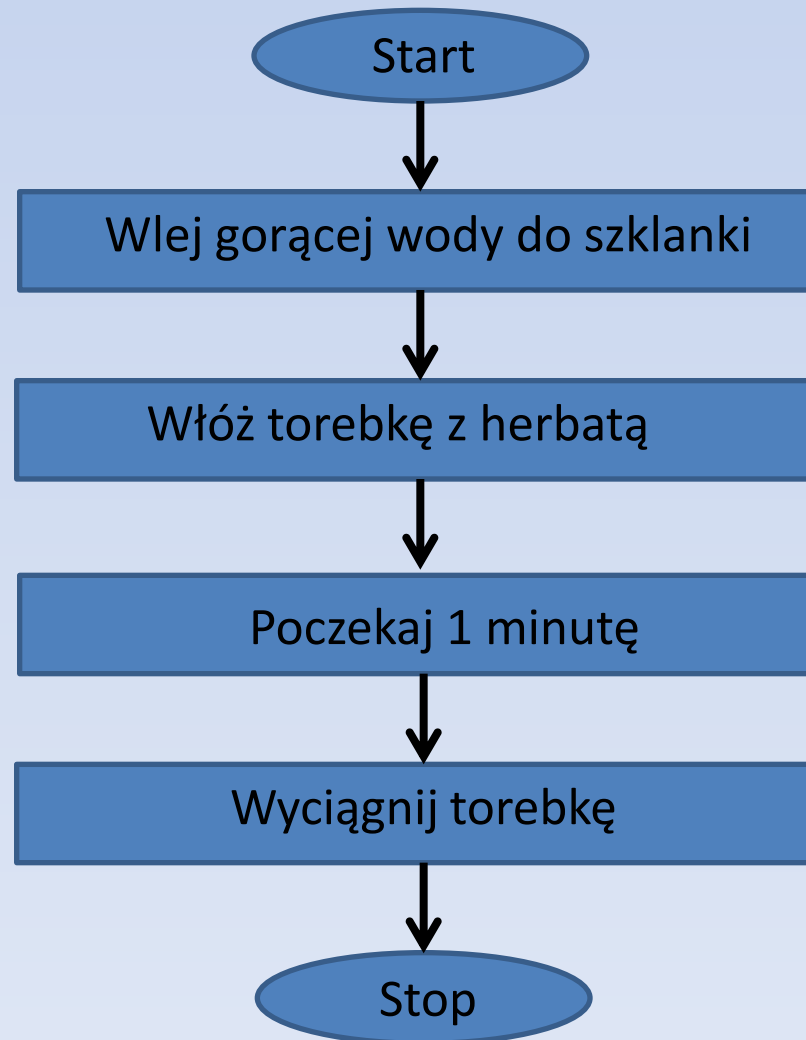
- **Praca z uczniem zdolnym**

- uczniowie przygotowują samodzielnie fragmenty zajęć na podstawie wyszukanych przez siebie w zasobach internetowych, podręcznikach lub pozycjach literatury popularnonaukowej informacji;
- propozycja zadania o wyższym poziomie trudności, w zakresie programowania (zmiana środowiska programistycznego, zastąpieniu graficznego języka programowania środowiska Prothio językiem tekstowym C/C++);
- pomoc innym uczniom mającym trudności w wykonywaniu zadań realizowanych na lekcji może stać się dobrą okazją do wypełnienia czasu który pozostał w wyniku szybszego zakończenia zadania.

Metody prowadzenia zajęć

- tworzenie schematów blokowych;
- tworzenie pseudokodów;
- połączenie instrukcji w całość;
- rozbudowa programu bazowego;
- modyfikacja gotowego programu;
- ilustracja końcowego efektu działania programu;
- stworzenie programu z podanych (wybranych) instrukcji;
- programowanie zespołowe.

1. Tworzenie schematów blokowych



Uczniowie budują schematy blokowe w celu ułatwienia projektowania programów.

2. Tworzenie pseudokodu

Pseudokod - jest taki sposób zapisu algorytmu, który, zachowując strukturę charakterystyczną dla kodu zapisanego w języku programowania, rezygnuje ze ścisłych reguł składniowych na rzecz prostoty i czytelności.

Uczniowie konstruują pseudokod, przy pomocy którego piszą swój program w Prophio.

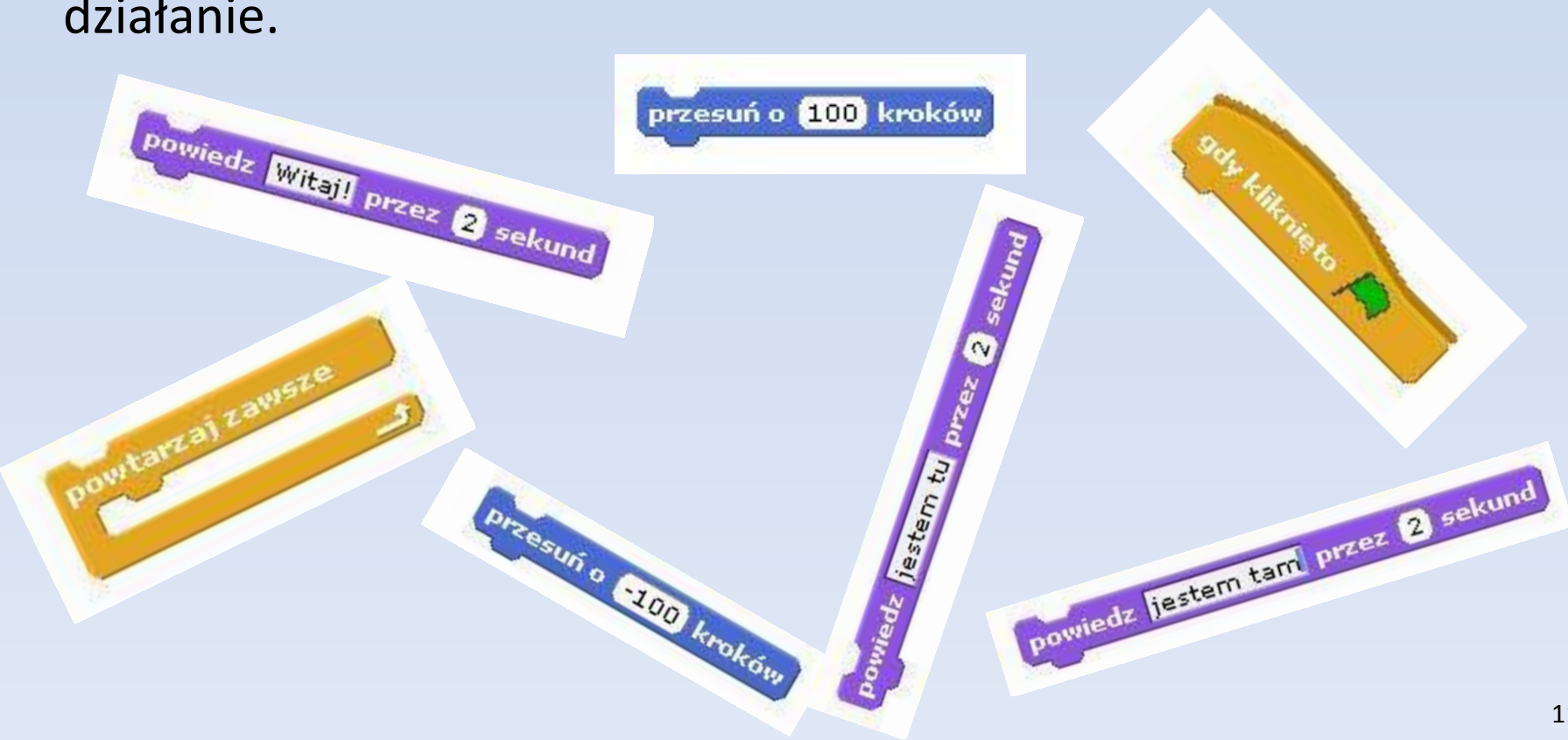
Przykład pseudokodu:

- wlej gorącej wody do szklanki,
- włóż torebkę z herbatą,
- poczekaj 1 minutę,

jeżeli herbata jest dość mocna **to** wyciągnij torebkę
w przeciwnym razie poczekaj 2 minuty
koniec warunku

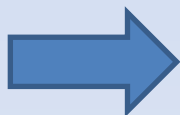
3. Połączenie instrukcji w całość

Uczniowie dostają instrukcje w formie papierowych kartoników. Po ułożeniu poszczególnych poleceń w formie programu przepisują skompletowany skrypt w Prophyo i sprawdzają jego działanie.



4. Rozbudowa programu bazowego

Uczniowie dostają podstawowy skrypt, który sami muszą rozbudować przy pomocy dostępnych instrukcji w Prophio i osiągnąć zadany końcowy efekt np. zmianę kostiumu.



4. Rozbudowa programu bazowego – efekt końcowy

Program bazowy



Program rozbudowany



5. Modyfikacja gotowego programu

- Uczniowie dostają gotowy program i testują go. Następnie muszą osiągnąć ten sam efekt działania skryptu przebudowując go w dowolny sposób (zmieniając użyte instrukcje).
- Sposób dochodzenia do założonego celu (konstrukcja programu) nie jest ważny, jedynie efekt końcowy (określony efekt). Ważna jest interpretacja ucznia, uzasadnienie wybranych rozwiązań czy zastosowanej metody.

6. Ilustracja końcowego efektu działania programu

- Uczniowie obserwują tylko końcowy efekt działania programu.

Następnie:

- piszą własny pseudokod/ schemat blokowy działania skryptu;
- tworzą swój program, żeby osiągnąć zamierzony efekt końcowy działania.

7. Stworzenie programu z gotowych instrukcji

- nauczyciel przedstawia uczniom wybrane instrukcje z dostępnych palet;
- uczniowie mają za zadanie napisanie własnego programu przy pomocy z góry określonych instrukcji;
- wybór i kolejność zastosowanych poleceń zależy od inwencji ucznia;
- wszystkie programy uczniów (sposoby użytych funkcji są prezentowane na forum klasy).

8. Programowanie zespołowe

- podział uczniów na kilkusobowe grupy i wybór lidera;
- propozycja tematyki zadania zaczerpniętego ze scenariuszy popularnych gier komputerowych;
- określenie celu, efektu końcowego działania programu przeprowadzenia eksperymentu, przetestowanie działania programu;
- prezentacja ostatecznych wyników działania programu w sposób ciekawy.

8. Programowanie zespołowe

- podział uczniów kilku osobowe grupy, wybór przedstawiciela;
- uczniowie w zespołach opracowują schemat blokowy działania własnego programu (efekt końcowy skryptu zaproponowany przez nauczyciela);
- każdy uczeń w grupie ma przydzielone swoje zadanie, pisze swój wybrany fragment programu;
- wszystkie napisane fragmenty przez uczniów zostaną skomponowane w całość przez lidera zespołu.

9. Rywalizacja między zespołami

- uczniowie tworzą kilku osobowe zespoły,
- każdy zespół realizuje to samo zadanie programistyczne, przedstawione przez nauczyciela;
- po określonym czasie liderzy grup prezentują programy stworzone przez swoje zespoły;
- po wszystkich prezentacjach omawiane są sposoby dojścia do ostatecznego efektu działania skryptów przez zespoły, zastosowane instrukcje.
- wybranie najciekawszego programu, sposobu osiągnięcia założonego efektu końcowego przez poszczególne grupy.

Sprawdź czy potrafisz

- Uczniowie w parach opracowują odpowiedzi na pytania z ***Sprawdź czy potrafisz***. Następnie odpowiedzi są porównywane z innymi parami i omawiane na forum klasy.
- Propozycja ułożenia własnych pytań w ***Sprawdź czy potrafisz*** przez uczniów:

Uczniowie w parach wymyślają swoje pytania dla kolejnej pary, tak żeby każda para dostała inne pytania. Po upływie czasu przewidzianego na opracowanie odpowiedzi osoba z każdej pary przedstawia otrzymane pytania i udzielone odpowiedzi. Wszystkie odpowiedzi są omawiane na forum klasy.

Dziękuję za uwagę