

Rozdział VI. Algorytmika

W rozdziale VI podręcznika *Informatyka. Podstawowe tematy. Wydanie nowe* wyjaśniamy, na czym polega stosowanie podejścia algorytmicznego w rozwiązywaniu problemów z wykorzystaniem komputera. Omawiamy wybrane sposoby prezentacji algorytmów, techniki algorytmiczne i algorytmy. Wyjaśniamy, na czym polega programowanie i modelowanie.

Temat 29. Opis problemu i przedstawianie sposobu jego rozwiązania w postaci graficznej

Celem tematu 29 jest wyjaśnienie, czym jest algorytm, omówienie etapów rozwiązywania problemu, sposobu zapisu algorytmu w postaci listy kroków oraz budowania schematu blokowego realizującego prosty algorytm.

- Algorytmika jest nowym działem dla uczniów gimnazjum. Zagadnienia związane algorytmiką nie są łatwe, dlatego należy je wprowadzać w sposób systematyczny i uporządkowany, zaczynając od prostych przykładów, przechodząc stopniowo do trudniejszych.
- Pojęcie algorytmu przedstawiamy jako uporządkowany i uściślony sposób rozwiązania danego problemu, zawierający szczegółowy opis wykonywanych czynności. Należy odwoływać się do znanych uczniom przykładów algorytmów z matematyki, fizyki i życia codziennego, zwracając uwagę na określanie danych i warunków, jakie mają być spełnione oraz wyników.
- Nauczyciel powinien wyjaśnić uczniom schemat postępowania przy rozwiązywaniu problemów za pomocą komputera (rys. 1, str. 228), korzystając z metody opisanej nad rysunkiem 1. Wybrany uczeń może przeczytać na głos treść metody. Można uzupełnić ten opis wyjaśnieniem, że komputer wykonuje programy, a programy realizują algorytmy.
- Etapy rozwiązywania problemów (punkt 2 tematu) najlepiej zaprezentować na konkretnych przykładach, podczas wykonywania i analizowania ćwiczeń 1-3, str. 230-232).
- Algorytmy można prezentować w kilku postaciach. W punkcie 3 tematu wyjaśniamy sposób prezentacji algorytmu w postaci listy kroków. Zaczynamy od wyjaśnienia, na czym polega zapis algorytmu w postaci listy kroków. Na przykładzie przepisu na gotowanie kisielu (str. 229) wskazujemy różnicę między algorytmami informatycznymi a nieinformatycznymi. Należy zwrócić uwagę uczniom na cechy algorytmów informatycznych: jasne określanie danych wejściowych, dokładne opisywanie wszystkich operacji oraz otrzymywanie tych samych wyników dla tych samych danych.
- Po przeanalizowaniu algorytmu obliczania pola trójkąta (Przykład specyfikacji zadania i listy kroków algorytmu, str. 230), uczniowie powinni wykonać ćw. 1 (str. 230). Należy sprawdzić poprawność wykonania tego ćwiczenia, zwłaszcza zapis specyfikacji zadania (często przez uczniów pomijany). Specyfikację zadania i listę kroków uczniowie mogą zapisać, korzystając z edytora tekstu. Wstawiając wzór na pole trapezu w jednym z kroków listy, przypomną sobie korzystanie z Edytora równań.
- W punkcie 4 tematu omawiamy prezentację algorytmu w postaci schematu blokowego. Wykorzystujemy do tego celu program ELI (wersja demonstracyjna tego programu jest zapisana na CD). Posługiwanie się programem ELI w podstawowym zakresie nie jest trudne. Uczniowie dość szybko zaczynają budować proste schematy blokowe na jednej planszy. Na początek wystarczy omówić (na konkretnym przykładzie) kilka podstawowych zasad budowania schematu w programie ELI, korzystając z projektora multimedialnego.
- Algorytm, który uczniowie będą prezentować w postaci schematu blokowego, jest prostym algorytmem liniowym (obliczenie sumy dwóch liczb i wyprowadzanie wartości). Uczniowie zapoznają się ze sposobem budowania schematu blokowego, korzystając z programu ELI.

Uczniowie, wykonując ćw. 2 (str. 231) powinni korzystać z podręcznika i pomocy do programu ELI.

- Program ELI umożliwia analizę poprawności algorytmu i testowanie rozwiązania dla różnych danych, czyli zrealizowanie dwóch ostatnich etapów rozwiązywania problemów przedstawionych na str. 229. Działanie algorytmu uczniowie sprawdzają, wykonując ćw. 3 (str. 232). Aby były widoczne aktualne wartości zmiennych, należy w wykonywanym projekcie otworzyć okno **Zmienne**, co ułatwi sprawdzenie działania algorytmu dla wielu danych.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-8 ze str. 233.

Temat 30. Techniki algorytmiczne

Celem tematu 30 jest omówienie przykładowych algorytmów z warunkami i algorytmów iteracyjnych oraz ich prezentacja w postaci schematów blokowych. Zgodnie z *Podstawą programową* omawiamy również wykorzystanie arkusza kalkulacyjnego do realizacji algorytmów.

- W temacie 29 uczniowie prezentowali algorytm liniowy w postaci listy kroków i schematu blokowego. Algorytm liniowy realizuje jeden ciąg obliczeń.
- W punkcie 1 tematu 30 omawiamy algorytm z warunkami. Należy odwołać się do sytuacji warunkowych znanych uczniom z innych przedmiotów lub z życia codziennego (rys. 1, str. 234).
- Nauczyciel omawia, korzystając z projektora multimedialnego, w jaki sposób przedstawia się sytuacje warunkowe w postaci schematu blokowego w programie ELI (str. 234).
- Uczniowie przeprowadzają analizę schematu blokowego z rozgałęzieniami, korzystając z gotowego projektu zapisanego w programie ELI (ćw. 2, str. 235). Proponuję, aby najpierw wykonali kilkakrotnie ten algorytm (w sposób krokowy) na swoich komputerach. Następnie wybrany uczeń za pomocą projektora multimedialnego, może zaprezentować wykonanie algorytmu. Uczniowie udzielają odpowiedzi na pytania zawarte w treści ćwiczenia. Nauczyciel powinien korygować wypowiedzi uczniów i podsumować wykonanie ćwiczenia.
- Po analizie gotowego przykładu w ELI, uczniowie wykonują własny schemat blokowy z warunkiem prostym (ćw. 3, str. 235) i warunkiem złożonym (ćw. 4, str. 235). Nauczyciel powinien wcześniej pokazać tworzenie schematu blokowego z warunkiem złożonym.
- Technikę iteracji przedstawiamy jako wielokrotne powtarzanie tych samych operacji (punkt 2 tematu). Wybrany uczeń powinien przeczytać na głos cały akapit nad ramką i treść pojęcia z ramki (str. 235). Iterację najlepiej wyjaśnić na konkretnym przykładzie, np. sumowania liczb (ćw. 5, str. 236).
- Wykonując ćw. 5 (str. 236), uczniowie mogą zrozumieć, na czym polega technika iteracji. Podobnie jak poprzednio, uczniowie najpierw powinni samodzielnie przeprowadzić symulację działania algorytmu, a potem, korzystając z projektora multimedialnego, wspólnie z nauczycielem przeanalizować rozwiązanie i odpowiedzieć na wszystkie pytania zawarte w treści ćwiczenia.
- Nauczyciel powinien dokładnie wyjaśnić, co oznacza operacja: $suma := suma + liczba$. Trzeba powiedzieć, że nie jest to równość w rozumieniu matematyki. Jest to operacja przypisania wartości zmiennej *suma* poprzedniej wartości zmiennej *suma* zwiększonej o kolejną dodawaną liczbę. Poprzednia wartość zmiennej *suma* zostanie zastąpiona nową. Można odwołać się do przykładu dodawania liczb w pamięci lub za pomocą kalkulatora: dodajemy dwie liczby do siebie, zapamiętujemy wynik, następnie do zapamiętanej wartości dodajemy kolejną liczbę, zapamiętujemy wynik (poprzedni kasując) itd. Powtarzamy te same operacje, czyli postępujemy iteracyjnie.

- Warto jeszcze zwrócić uwagę na przypisanie: $nr := nr + 1$. Jest to tzw. licznik, który zwiększa się o 1 przy każdym przejściu pętli.
- W punkcie 3 tematu pokazujemy wykorzystanie arkusza kalkulacyjnego do realizacji algorytmów liniowych i algorytmów z warunkami. Nauczyciel powinien zwrócić uwagę, że większość obliczeń w arkuszu kalkulacyjnym przebiega według algorytmu liniowego. Warto przypomnieć uczniom przykładowe algorytmy liniowe, które realizowali w arkuszu (np. ćw. 5-6, str. 162-163).
- Stosując funkcję logiczną JEŻELI, można zrealizować w arkuszu kalkulacyjnym algorytm z warunkami (w tym z warunkami złożonymi). Uczniowie mogą odszukać ćwiczenia z rozdziału IV, w których już realizowali takie algorytmy (np. ćw. 7, str. 165).
- Uczniowie, wykonując ćw. 7 i 8 (str. 237), realizują przykładowe algorytmy liniowe i algorytmy z warunkami w arkuszu kalkulacyjnym.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-4 ze str. 238.

Temat 31. Na czym polega programowanie i modelowanie?

Celem tematu 31 jest wyjaśnienie podstawowych pojęć związanych z programowaniem oraz omówienie prezentacji algorytmów w języku programowania. Tłumaczymy również, na czym polega tworzenie prostego modelu i prezentujemy przykłady modeli zjawisk.

- Nauczyciel wyjaśnia uczniom, na czym polega prezentowanie algorytmu w języku programowania, podając przykłady konkretnych nazw pakietów programistycznych. Uczniowie mogą również podać znane im nazwy. Nauczyciel powinien zwrócić uwagę na korzystanie z licencjonowanego oprogramowania komputerowego.
- Proces kompilacji, uruchomienia i wykonania programu komputerowego napisanego w języku wysokiego poziomu nauczyciel powinien omówić na konkretnym przykładzie, korzystając z projektora multimedialnego. Proponuję przygotować przykładowy prosty program, np. program w języku Turbo Pascal pokazany na rys. 1 (str. 241).
- Korzystając ze schematu na rys. 1 (str. 241), nauczyciel powinien wyjaśnić, co to jest postać źródłowa i wynikowa programu i co oznacza deklaracja zmiennych. Z pojęciem zmiennej w algorytmie uczniowie spotkali się, prezentując algorytmy w postaci schematu blokowego w programie ELI. W oknie **Zmienne** mogli obserwować jak zmieniają się wartości zmiennych. Proponuję otworzyć i uruchomić przykładowy projekt w ELI.
- Różnicę między kompilacją a interpretacją programu najłatwiej pokazać na przykładzie programowania w języku Logo. Na prostym przykładzie można pokazać, że w przypadku języka Logo tłumaczenie programu następuje instrukcja po instrukcji. Jeśli popełnimy błąd przy wpisywaniu pojedynczej instrukcji, nie zostanie ona wykonana, tylko od razu pojawi się komunikat o błędzie – inaczej niż w przypadku kompilacji (gdzie program jest tłumaczony w całości i po przeprowadzeniu kompilacji wypisane są wszystkie błędy występujące w programie). Do interpretacji programu wrócimy jeszcze w temacie 32, gdy uczniowie będą pisać samodzielnie polecenia w języku Logo.
- W podsumowaniu zagadnienia prezentacji algorytmu w języku programowania, wybrany uczeń może przeczytać na głos definicje pojęć (*język programowania, translacja, kompilacja, interpretacja, komórka pamięci*) ze słowniczka (str. 240-241).
- W tym temacie uczeń dowiaduje się, na czym polega modelowanie rzeczywistości na przykładzie rzutu kostką sześcienną do gry. Proponuję, aby uczniowie posłużyli się najpierw prawdziwą kostką do gry i wykonali np. sześć rzutów (rzucanie kostką sześćdziesiąt razy byłoby za bardzo czasochłonne, ale można zlecić wcześniejsze wykonanie w domu takich rzutów i spisanie, ile razy wypadła dana liczba oczek). Nauczyciel omawia szczegółowo model rzutu kostką sześcienną do gry, korzystając z pliku *Rzutkostka* (CD) oraz z opisu

w podręczniku (str. 242). Zleca uczniom otwarcie pliku *Rzutkostka* i obserwację zachowania modelu (ćw. 1, str. 242).

- Uczniowie, na bazie pliku *Rzutkostka*, mogą wykonać w arkuszu kalkulacyjnym inny model – rzut monetą.
- Na płycie CD jest zapisana aplikacja o nazwie *Modele PWN*. Korzystając z tego programu, uczniowie powinni obejrzeć i przeprowadzić symulację kilku modeli różnych zjawisk, np. przyrodniczych. Po zapoznaniu się z modelami, uczniowie powinni podzielić się w dyskusji swoimi spostrzeżeniami.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-7 ze str. 244-245.

Temat 32. Zapisywanie algorytmów w języku Logo

Celem tematu 32 jest pokazanie, jak pisać proste polecenia w języku Logo i zapisywać algorytm w postaci procedur, w tym algorytm z warunkami i algorytm iteracyjny.

- Realizację tego tematu proponuję rozpocząć od pisania prostych poleceń w języku Logo. Należy podkreślić, że Logo jest językiem edukacyjnym i, korzystając z niego, można poznać podstawowe pojęcia i metody dotyczące programowania oraz zdobyć umiejętności tworzenia procedur, w tym procedur z parametrami, i ich wywoływania. Zdobyta w gimnazjum wiedza i umiejętności dotyczące programowania stanowią dla ucznia dobrą podstawę do rozszerzania jej na lekcjach informatyki w szkole ponadgimnazjalnej.
- Często wśród uczniów i ich rodziców (czasem wśród nauczycieli) pojawiają się wątpliwości, po co uczyć programowania, przecież nie każdy uczeń zostanie w przyszłości informatykiem. Jest to mylne rozumowanie, ponieważ w ten sposób możemy oceniać przydatność dowolnego fragmentu wiedzy z każdego przedmiotu. Uczniowie gimnazjum powinni poznać daną dziedzinę wiedzy na tyle, aby mogli w przyszłości świadomie wybrać dalszy kierunek kształcenia. Dlatego na informatyce nie wolno pominąć: algorytmiki, programowania i baz danych. Ponadto nauka algorytmiki i programowania rozwija logiczne myślenie, a możliwości zdobycia tej umiejętności nie trzeba chyba komentować...
- Na początku zajęć należy krótko omówić, posługując się projektorem multimedialnym, budowę okna programu Logomocja oraz wyjaśnić, na czym polega programowanie w Logo. Nawiązujemy również do omówionej w temacie 31 interpretacji programu. Pisanie programu w Logo polega na wydawaniu poleceń żółtowi, który je wykonuje po naciśnięciu przez użytkownika klawisza **Enter**. Nauczyciel, powinien wpisać kilka przykładowych poleceń w Logo i pokazać efekt ich wykonania na ekranie, korzystając z projektora. Przy okazji może omówić kilka podstawowych zasad pisania programów w Logo. Uczniowie, którzy już w szkole podstawowej pisali polecenia w Logo, mogą podzielić się swoją wiedzą.
- Uczniowie piszą pierwsze programy (ćw. 1 i 2, str. 248), korzystając z opisu poleceń w tabeli (rys. 1, str. 247). Nie powinni tej tabeli przepisywać do zeszytu. W zeszycie powinni zapisywać ewentualnie nowe polecenia, dodatkowo podane przez nauczyciela.
- Ciekawą możliwością programu Logomocja jest możliwość zmiany postaci żółwia na wybraną postać animowaną utworzoną w Edytorze postaci i zapisaną w pliku. Jeśli w naszej pracowni nie jest zainstalowana pełna wersja programu, można skorzystać z gotowych obrazów zapisanych razem z wersją demonstracyjną na CD (w folderze *Obrazy*).
- W punkcie 3 tematu pokazujemy sposób zapisania powtarzających się operacji w Logo. Jest to wprowadzenie do zapisu algorytmów iteracyjnych.
- W punkcie 4.1 tematu uczniowie poznają, w jaki sposób definiować procedury bez parametrów w języku Logo. Proponuję wcześniej przygotować przykładową procedurę w Logo i omówić sposób definiowania i wywoływania procedury, korzystając z projektora. Należy skorzystać również z opisu podanego w punkcie 1 tematu.

- Następnie uczniowie samodzielnie definiują procedurę OŚMIOKĄT (str. 250-251), korzystając z podręcznika.
- W punkcie 4.2 tematu uczniowie poznają, w jaki sposób definiować procedury z parametrami. Nauczyciel, podobnie jak poprzednio, powinien dokładnie omówić definiowanie i wywoływanie procedury na konkretnym przykładzie. Należy przede wszystkim wskazać różnicę między parametrami formalnymi i aktualnymi (patrz opis na str. 246).
- Uczniowie, wykonując ćwiczenia 7-9 (str. 251) definiują procedury z parametrami. Powinni zauważyć, że w definicji nowej procedury mogą korzystać ze zdefiniowanej wcześniej.
- Podczas wykonywania wszystkich ćwiczeń nauczyciel powinien obserwować i sprawdzać pracę oraz pomagać nieradzącym sobie uczniom. Można podpowiedzieć uczniom, że jeśli do wiersza poleceń chcą wprowadzić polecenie, które wcześniej napisali i jest umieszczone w historii poleceń, mogą je wybrać, naciskając klawisze strzałek kursora (góra-dół).
- Jeśli szkoła dysponuje pełną wersją programu, uczniowie mogą zapisywać swoje projekty w pliku (ćw. 10, str. 252).
- W ostatnim punkcie tematu pokazujemy realizację prostej sytuacji warunkowej w języku Logo, a uczniowie, wykonując ćwiczenie 11 (str. 252) zapisują algorytm z warunkami w Logo.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-6 ze str. 252-253.

Temat 33. Wybrane algorytmy

Celem tematu 33 jest zapoznanie uczniów z algorytmami znajdowania wybranego elementu w zbiorze uporządkowanym i nieuporządkowanym oraz z przykładowymi algorytmami porządkowania zbioru elementów.

- Zgodnie z *Podstawą programową* uczeń: „opisuje sposób znajdowania wybranego elementu w zbiorze nieuporządkowanym i uporządkowanym, opisuje algorytm porządkowania zbioru elementów”.
- W celu ułatwienia zrozumienia wybranych algorytmów, przedstawione są one na schematycznych rysunkach, a niektóre zapisane dodatkowo w postaci listy kroków. Można również niektóre algorytmy wykonać praktycznie, posługując się przygotowanymi wcześniej rekwizytami.
- W punkcie 1 tematu 33 omawiamy sposób znajdowania wybranego elementu w zbiorze nieuporządkowanym. Zaczynamy od przedstawiania w postaci listy kroków algorytmu wyboru większej z dwóch liczb. Uczniowie powinni zapoznać się z listą kroków (str. 254), a następnie ze schematem blokowym (ćw. 1, str. 254). Algorytm jest bardzo prosty. Nauczyciela może wyręczyć wybrany uczeń i, po wykonaniu ćwiczenia, omówić działanie algorytmu.
- Wybór największej liczby spośród n liczb jest trochę trudniejszy. Ten algorytm powinien wyjaśnić nauczyciel, a uczniowie w tym czasie mogą śledzić listę kroków w podręczniku (str. 254).
- Algorytm znajdowania największego elementu w zbiorze nieuporządkowanym jest przykładem metody nazywanej przeszukiwaniem liniowym. Aby uczniowie lepiej zrozumieli ten algorytm, mogą wykonać go praktycznie, wybierając najwyższego ucznia spośród pięciu uczniów. Każdy będzie trzymał w ręku kartkę z odpowiednim napisem: uczeń 1 (160 cm), uczeń 2 (150 cm) itd. Proponuję najpierw wykonywać ten algorytm dokładnie według opisu na str. 255, a wyniki zapisywać na tablicy. Następnie uczniowie mogą wpisać na kartki liczby oznaczające ich wzrost i wykonać jeszcze raz algorytm. W podsumowaniu uczniowie sprawdzają działanie tego algorytmu dla innych danych, wykonując ćw. 2 (str. 255).

- Element w zbiorze uporządkowanym możemy znaleźć, stosując algorytm poszukiwania przez połowienie. Nauczyciel powinien wyjaśnić, że jest to przykład metody „dziel i zwyciężaj”. Wybrany uczeń czyta treść ramki na str. 255.
- Uczniowie wykonują praktycznie algorytm poszukiwania przez połowienie, grając w grę „zgadywanie liczby” (ćw. 3, str. 256). Grę rozgrywają w parach.
- Przykład stosowania algorytmu poszukiwania przez połowienie jest pokazany na rys. 1 (str. 256). Proponuję, aby uczniowie przygotowali wcześniej odpowiednie pomoce dydaktyczne: kartoniki z zapisanymi liczbami (poza kartonikiem z liczbą 40 pozostałe z jednej strony mają być pomalowane na zielono, a z drugiej na żółto). Uczniowie czytają samodzielnie opis algorytmu (str. 256) i analizują go na rysunku 1. Następnie wybrany uczeń może omówić przebieg algorytmu i wykonać go praktycznie, korzystając z przygotowanych pomocy. Nauczyciel powinien poprawić i uzupełnić wypowiedzi uczniów, jeśli mają problemy z wyjaśnieniem działania algorytmu.
- W punkcie 3 tematu 33 zapoznajemy uczniów z wybranymi algorytmami porządkowania zbioru elementów – sortowania przez wybór i sortowania bąbelkowego.
- Sortowanie przez wybór uczniowie poznają na przykładzie porządkowania dziesięciu osób z klasy według wzrostu. Uczniowie powinni wcześniej zapoznać się z listą kroków tego algorytmu i przeanalizować rysunek 2 (str. 257). Do przeprowadzenia pokazu będzie potrzebnych dwunastu uczniów. Dziesięciu z nich otrzymuje kartonik z kolejnym numerem i liczbą określającą wzrost (można wykorzystać pomoce z algorytmu przeszukiwania liniowego). Jeden z pozostałych uczniów czyta kolejne kroki listy kroków, drugi „przetwarza elementy ciągu”, natomiast pozostali uczniowie systematycznie wykonują kolejne kroki algorytmu. Liczbę określającą miejsce, w którym stoi uczeń najwyższy, można zapisywać na tablicy.
- Algorytm sortowania przez wybór można również pokazać, przekładając odpowiednio kartoniki z liczbami (można wykorzystać pomoce z algorytmu sortowania przez połowienie), korzystając z rysunku 2 (str. 257).
- Innym przykładowym algorytmem omówionym w tym temacie jest algorytm sortowania bąbelkowego. Nauczyciel powinien zlecić wcześniejsze przygotowanie odpowiednich pomocy dydaktycznych. Można również wykorzystać pomoce z poprzednich algorytmów – w tym przypadku są to również kartoniki z liczbami. Uczniowie na bazie rysunku 3 (str. 258) mogą samodzielnie wykonać ten algorytm, przedstawiając odpowiednio kartki.
- W podsumowaniu uczniowie mogą przeanalizować gotowy schemat blokowy algorytmu sortowania bąbelkowego, korzystając z programu ELI. Schemat blokowy nie jest prosty – jest zbudowany na dwóch planszach (na planszy głównej wywoływana jest procedura), a dane zapisane są na **Taśmie**. Nauczyciel powinien omówić dokładnie budowę tego schematu, a następnie sposobem krokowym, wspólnie z uczniami, sprawdzić jego działanie dla różnych danych (dane na **Taśmie** można zmieniać).
- Uwagi: Każdy z algorytmów należy wykonać kilkakrotnie dla różnych danych. Kartoniki z liczbami najlepiej rozkładać np. na dużym stole (jeśli taki stoi w pracowni) lub na podłodze, aby wszyscy uczniowie mogli obserwować wykonywanie danego algorytmu. Podczas wykonywania algorytmów należy zwracać uwagę na to, że powtarzamy kilkakrotnie te same operacje, czyli wykonujemy je iteracyjnie.
- Nie należy wymagać od wszystkich uczniów, aby potrafili zapisywać omówione w tym temacie algorytmy w postaci programu komputerowego lub schematu blokowego. Uczniowie powinni przede wszystkim je rozumieć i umieć opisać ich działanie. Jeśli jednak w klasie są uczniowie zainteresowani zapisaniem danego algorytmu w postaci programu, powinniśmy im to umożliwić i pomóc.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-7 ze str. 259.

Temat 34. Programujemy w Baltie

Celem tematu 34 jest pokazanie tworzenia programów w środowisku programowania Baltie. Omawiamy m.in. możliwości powtarzania poleceń i tworzenia animacji.

- Program Baltie jest programem edukacyjnym, który podobnie jak program Logomocja, ułatwia uczniowi rozumienie, na czym polega programowanie. Jest to program bardzo przyjazny dla użytkownika, dlatego uczniowie szybko i sprawnie się nim posługują. Niektórzy uczniowie mogli poznać ten program w szkole podstawowej.
- Program Baltie umożliwia pracę w trzech trybach (**Budowanie**, **Czarowanie** i **Programowanie**). Można krótko pokazać, na czym polega praca w dwóch pierwszych trybach, ale nie należy omawiać ich szczegółowo (przeznaczone są bowiem dla młodszych dzieci).
- Należy, na konkretnym przykładzie, wyjaśnić uczniom sposób tworzenia programu w środowisku Baltie (str. 260-261). Wskazujemy również różnicę między tworzeniem programu w Baltie a pisaniem programu w Logo. W Logo wydawaliśmy polecenia żółwiowi, który po zaakceptowaniu zaraz je wykonywał, natomiast w programie Baltie wprowadzamy wszystkie polecenia dla czarodzieja i wykonujemy je, uruchamiając program.
- Przed przystąpieniem do pisania pierwszego programu (ćw. 1-4, str. 261-262), uczniowie powinni zapoznać się z praktycznymi poradami na str. 261.
- W punkcie 2 tematu 34 uczniowie tworzą programy realizujące algorytmy iteracyjne. Należy wcześniej przypomnieć, na czym polega iteracja. Nauczyciel krótko omawia sposób realizacji powtórzeń w programie Baltie, a uczniowie samodzielnie wykonują ćw. 5 i 6 (str. 263), tworząc i modyfikując programy.
- Korzystając z programu Baltie, warto omówić pętle zagnieżdżone (punkt 3). W podręczniku systematycznie naprowadzamy uczniów na zastosowanie pętli w pętli – podczas budowania muru (ćw. 7 i 8, str. 263)
- Uczniowie, tworząc w prostym środowisku program z pętlą zagnieżdżoną, skuteczniej zrozumieją tę metodę i łatwiej będzie im zastosować ją w przyszłości, np. gdy będą programowali w języku Pascal lub C.
- Na zakończenie tematu 34 (punkt 4) omawiamy możliwość tworzenia animacji w programie Baltie. Można przypomnieć sposób tworzenia animacji w Edytorze postaci. Po krótkim wprowadzeniu nauczyciela, uczniowie powinni wykonać ćwiczenia 9 i 10 (str. 264) samodzielnie, korzystając z podręcznika.
- Nauczyciel powinien obserwować wykonywanie wszystkich ćwiczeń, pomagać nieradzącym sobie uczniom i nie przeszkadzać, jeśli dobrze sobie radzą. Należy sprawdzać wykonanie poszczególnych ćwiczeń i zlecać poprawienie, gdy są zrobione błędnie.
- Uwaga: Na płycie *Niezbędnik nauczyciela* jest zapisany podręcznik użytkownika programu Baltie.
- Do domu należy zadać przeczytanie treści całego tematu i przygotowanie ustnie odpowiedzi na pytania 1-5 ze str. 265.