

Cele edukacyjne

- Usystematyzowanie podstawowych pojęć: *algorytm z warunkami*, *iteracja*, *algorytm iteracyjny*, *zmienna sterująca*.
- Prezentacja algorytmów z warunkami i iteracyjnymi w trzech notacjach: w postaci listy kroków, schematu blokowego i programu komputerowego.

Proponowany czas realizacji – 6 godz.

Wskazówki metodyczne

- Temat 2. traktujemy częściowo jako powtórzenie i usystematyzowanie materiału z pierwszej klasy. Uczniowie powinni powtórzyć odpowiednie treści zawarte w tematach C4 i C5 z podręcznika *Informatyka podstawowa*.
- Podobnie jak w temacie 1. również w tym temacie łączymy algorytmikę z programowaniem – uczeń poznaje algorytmy z warunkami i iteracyjne oraz przedstawia je w trzech notacjach: w postaci listy kroków, schematu blokowego oraz programu komputerowego.
- Aby nauka programowania była skuteczna, podczas wprowadzania nowych instrukcji (tu: warunkowej i iteracyjnej) należy zawsze podawać konkretny przykład i na jego bazie omówić zastosowanie danej instrukcji. Nowe instrukcje i struktury danych oraz zasady programowania należy wprowadzać stopniowo, aby uczniowie zdążyli je utrwalić, pisząc po kilka programów, w których je zastosują.
- Szczegółowy opis podstawowych instrukcji programowania znajduje się w plikach *C_instrukcje.pdf* i *Pascal_instrukcje.pdf* (CD). Można te dokumenty wydrukować i rozdać uczniom.
- Tematy podręcznika (w tym temat 2.) są tak skonstruowane, aby uczniowie, analizując przykłady i korzystając z opisu instrukcji, mogli samodzielnie wykonywać większość zaproponowanych w podręczniku ćwiczeń i zadań.

Lekcja 6. Algorytmy z warunkami

- Na pierwszej lekcji tematu 2. wskazujemy różnicę między algorytmem liniowym a algorytmem z warunkami. Na przykładzie algorytmu sprawdzania, czy dana liczba jest parzysta, pokazujemy sposób prezentacji algorytmu z warunkami w postaci listy kroków i schematu blokowego. Polecamy uczniom odszukanie w pliku *bloki.pdf* (CD) bloku warunkowego.
- Na przykładzie tego samego algorytmu należy omówić zapisywanie algorytmu z warunkami w postaci programu komputerowego, w tym postać i działanie instrukcji warunkowej oraz zapisywanie warunków logicznych (prostych i złożonych) z zastosowaniem odpowiednich operatorów logicznych.

- Uczniowie po wcześniejszej dokładnej analizie i przetestowaniu przykładów z podręcznika powinni samodzielnie przedstawić podany w ćwiczeniu 3. (str. 24) algorytm liniowy w postaci listy kroków, schematu blokowego, a następnie w postaci programu. Uwaga: Schematy blokowe uczniowie mogą też rysować odręcznie w zeszytach.
- Jeśli uczniowie nie zdążą wykonać wszystkich zaplanowanych na tej lekcji ćwiczeń, powinni dokończyć je w domu.

Lekcja 7. Algorytm sprawdzania warunku istnienia trójkąta

- Jeśli uczniowie zrozumieją i nauczą się, w jaki sposób zapisywać algorytm z warunkami w różnych notacjach, bez problemu powinni poradzić sobie z narysowaniem schematu blokowego algorytmu sprawdzania warunku istnienia trójkąta i zapisaniem go w postaci programu komputerowego. Mogą jeszcze bazować na zapisanej w przykładzie 4. (str. 27) liście kroków tego algorytmu. Należy tylko zwrócić uwagę na sposób zapisywania warunków złożonych.
- Trzeba brać pod uwagę, że nie wszyscy uczniowie będą pamiętali warunek istnienia trójkąta, dlatego przypominamy go w podręczniku. W podanym rozwiązaniu sprawdzamy, czy długość dowolnego boku jest mniejsza od sumy długości dwóch pozostałych boków. Długości tych odcinków nie muszą oczywiście być liczbami naturalnymi, w zadaniu przyjmujemy jednak takie ograniczenie. Warto zauważyć, że nie musimy sprawdzać wszystkich par utworzonych z trzech podanych wartości. Wystarczy porównać długość najdłuższego odcinka z sumą długości dwóch pozostałych. Uczniowie zazwyczaj pamiętają tę zasadę z gimnazjum. Uczniom zainteresowanym można polecić napisanie programu, w którym zastosują tę drugą metodę.
- W przykładowych scenariuszach pokazuję szczegółowy przebieg lekcji 7. W przypadku tej lekcji wystarczy jedna wersja scenariusza, ponieważ skupiamy się przede wszystkim na rozwiązaniu problemu, niezależnie od zastosowanego języka programowania, którego wybór w tym przypadku jest drugorzędny. Dodatkowo w językach Pascal i C++ instrukcja warunkowa ma podobną postać i podobnie działa.

Lekcja 8. Algorytm z warunkami zagnieżdżonymi

- Algorytmy z warunkami zagnieżdżonymi zazwyczaj sprawiają uczniom większe problemy, dlatego należy dokładnie omówić prezentację takiego algorytmu w postaci listy kroków i schematu blokowego, zwracając uwagę na kroki listy i miejsce w schemacie, gdzie warunki się zagnieżdżają.
- Omawiając zapis takiego algorytmu w postaci programu, należy zwrócić uwagę na odpowiedni sposób zapisywania instrukcji warunkowych. Warto odwoływać się do schematu blokowego algorytmu, aby uczniowie zauważyli, że taki rysunek może czasem być pomocny.

Lekcja 9. Algorytm iteracyjny

- Mimo że uczniowie powinni zapoznać się z pojęciem iteracji w gimnazjum i na zajęciach z informatyki realizowanej w zakresie podstawowym, należy powtórzyć tę technikę, porządkując zwłaszcza słownictwo w tym zakresie. Uczniowie powinni zrozumieć i zapamiętać opis iteracji podany w ramce na stronie 30: *Iteracja polega na wielokrotnym powtarzaniu tej samej operacji (ciągu operacji). Iterację implementujemy, stosując tzw. pętlę. Z pętlą mamy do czynienia, gdy w pewnym kroku algorytmu wracamy do jednego z wcześniejszych kroków, co powoduje, że kroki te mogą zostać wykonane wiele razy.*
- Zaczynamy od prostego przykładu: algorytmu obliczania iloczynu n liczb. Pokazujemy również realizację tego algorytmu w trzech notacjach. Bardzo dokładnie omawiamy listę kroków podaną w przykładzie 8. (str. 30). Należy wyjaśnić, czym jest zmienna sterująca pętli. Wyjaśniamy też, że przypisanie: $i:=i+1$ (lub $i++$ występujące w zapisie programu w języku C++) oznacza przy-

pisanie wartości zmiennej i poprzedniej wartości tej zmiennej, powiększonej o jeden. Należy też wyjaśnić przypisanie: `iloczyn:=iloczyn*a` (lub `iloczyn*=a` w zapisie programu w języku C++). Wbrew pozorom operacja ta nie jest oczywista dla uczniów.

- Z algorytmami operującymi na n danych uczniowie będą często się stykać. Z tego powodu warto pierwszy schemat blokowy i program wspólnie z nimi omówić. Uczniowie powinni wskazać miejsce wprowadzania ilości liczb (n) i mnożenia tych liczb, określić operacje, które są powtarzane, sposób przechodzenia do kolejnego kroku iteracji, sposób jej zakończenia. Powinni wskazać zmienną sterującą pętlą.
- Nie wprowadzamy w tym temacie wszystkich instrukcji iteracyjnych. Celowo omawiamy tylko instrukcję `for`, a pozostałe – w temacie 5. W ten sposób uczeń, wykonując określoną liczbę zadań, może skuteczniej zrozumieć i utrwalić działanie jednej instrukcji.

Lekcja 10. Pętle zagnieżdżone

- Zagadnieniem trudniejszym dla uczniów są pętle zagnieżdżone. Należy zacząć od omówienia prezentacji algorytmu w postaci schematu blokowego zawierającego tego typu pętle (przykład 10, str. 34). Nauczyciel wskazuje na rysunku 5. (str. 35) sposób konstrukcji pętli zagnieżdżonych oraz omawia zapis tego algorytmu w języku Pascal i/lub w C++. Następnie uczniowie piszą program, stosując pętlę zagnieżdżoną (ćw. 19, str. 36).

Lekcja 11. Rozwiązywanie zadań

- Tę lekcję poświęcamy na samodzielne rozwiązywanie zadań przez uczniów. Należy wykonać wszystkie ćwiczenia i zadania, które nie zostały zrobione na poprzednich lekcjach. Jeśli zabraknie czasu, to pozostałe zadania uczniowie powinni wykonać w domu.

Błędy i problemy uczniów

- Część uczniów ma jeszcze problemy z poprawnym zapisaniem nawet prostego programu.
- Uczniowie mają problemy z zapisem instrukcji programu w odpowiedniej kolejności.
- Zapominają o odpowiednim umieszczaniu znaków interpunkcyjnych. Nie przywiązują wagi do precyzyjnego zapisu tekstu programu.
- W języku C++ małe i wielkie litery mają różne znaczenie, o czym uczniowie zapominają.
- Część uczniów nie pamięta, że po każdej zmianie program należy na nowo skompilować.
- Mają trudności ze zrozumieniem zapisu algorytmu zawierającego warunki lub pętle zagnieżdżone – zarówno w przypadku schematu blokowego, jak i programu w języku Pascal lub C++.

Wskazówki do niektórych ćwiczeń, pytań i zadań

Ćwiczenie 5. (str. 26) – pliki `T2_c5_R.pas`, `T2_c5_R.cpp`.

Ćwiczenie 10. (str. 28) – pliki `T2_c10_R.pas`, `T2_c10_R.cpp`.

Ćwiczenie 13. (str. 30) – pliki `T2_c13_R.pas`, `T2_c13_R.cpp`.

Ćwiczenie 17. (str. 34) – pliki `T2_c17_R.pas`, `T2_c17_R.cpp`.

Ćwiczenie 19. (str. 36) – pliki `T2_c19_R.pas`, `T2_c19_R.cpp`.

Zadanie 1. (str. 37) – odpowiedzi (Pascal i C++) a. zawsze; b. dla $x \leq y$; w instrukcji w podpunkcie a brak słowa kluczowego `else`, co oznacza, że instrukcja występująca tu w trzeciej linii będzie wykonana jako kolejna po `if`.

Zadanie 2. (str. 37) – pliki `T2_z2_R.pas`, `T2_z2_R.cpp`.

Zadanie 7. (str. 37) – należy zastosować pętle zagnieżdżone.

Lekcja 7. (temat 2.)

Algorytm sprawdzania warunku istnienia trójkąta



Wiedza i umiejętności		Treści, pytania, ćwiczenia i zadania z podręcznika, materiały z CD, formy pracy na lekcji
podstawowe	rozszerzające	
Uczeń:	Uczeń:	
Zna i omawia warunek istnienia trójkąta. Potrafi zapisać warunek złożony. Buduje schemat blokowy algorytmu sprawdzania warunku trójkąta. Korzystając z przykładu, zapisuje algorytm sprawdzania warunku istnienia trójkąta; stosuje warunek złożony.	Korzystając z dodatkowych źródeł, znajduje inny, niż podany w podręczniku, sposób sprawdzenia, czy z danych trzech odcinków można zbudować trójkąt. Zapisuje ten algorytm w postaci programu komputerowego.	Temat 2. z podręcznika (str. 27-28); ćwiczenia 8-10 (str. 27-28); zadanie domowe zadanie 1. (str. 37); Formy pracy: wprowadzenie, praca z podręcznikiem; ćwiczenia.
Podstawa programowa:		
5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń:		
2) stosuje podejście algorytmiczne do rozwiązywania problemu;		
4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;		
5) posługuje się podstawowymi technikami algorytmicznymi;		
6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;		
7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;		
11) opisuje podstawowe algorytmy i stosuje: f) algorytmy badające własności geometryczne – sprawdzanie warunku trójkąta;		
17) ocenia zgodność algorytmu ze specyfikacją problemu;		
21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;		
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;		
26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania;		

Przebieg lekcji – (język Pascal i/lub C++)

1. Nauczyciel podaje temat i cel lekcji oraz sprawdza zadanie domowe. Uczniowie odpowiadają na zadane pytania. Odpowiedź na pytanie 2. (str. 36) wybrany uczeń przedstawia, korzystając z projektora.
2. Nauczyciel lub wybrany uczeń omawia warunek istnienia trójkąta (str. 27). Nauczyciel lub uczeń podaje inne możliwe rozwiązanie: porównanie długości najdłuższego odcinka z sumą długości dwóch pozostałych.

3. Uczniowie analizują listę kroków algorytmu sprawdzania warunku trójkąta (przykład 4, str. 27), testują ją dla wybranych danych (ćw. 8, str. 27) i tworzą schemat blokowy tego algorytmu (ćw. 9, str. 27).
4. Nauczyciel zwraca uwagę, w jaki sposób w języku Pascal (i/lub w C++) zapisuje się instrukcję zawierającą warunek złożony (przykład 5, str. 28).
5. Korzystając z przykładu 5. (str. 28), uczniowie wykonują ćwiczenie 10. (str. 28). Nauczyciel nadzoruje wykonywanie ćwiczenia, udzielając ewentualnie niezbędnych wskazówek.
6. Uczniom zainteresowanym można polecić (dodatkowo) napisanie programu, w którym zastosują drugą omówioną metodę.
7. W podsumowaniu zajęć wybrany uczeń, korzystając z projektora, omawia i prezentuje rozwiązanie ćwiczenia 10. (str. 28). Swoje rozwiązanie pokazuje też uczeń, który wykonał zadanie drugą metodą.
8. Nauczyciel ocenia najbardziej aktywnych uczniów, zwłaszcza tych, którzy pracowali samodzielnie, i poleca przeczytanie w domu z podręcznika części tematu omawianej na lekcji (str. 27-28). Uczniowie zainteresowani mogą zapoznać się samodzielnie z treścią punktu 4. tematu 2.