

Jak uczyć trudnych tematów - algorytmizacja i programowanie

Wydaje się, że najtrudniejszym zagadnieniem zarówno dla ucznia, jaki i nauczyciela w edukacji informatycznej są tematy dotyczące algorytmizacji i programowania. Z zagadnieniami tymi uczeń spotyka się już w szkole podstawowej, zaś dalej tematyka ta kontynuowana jest na lekcjach informatyki w szkole ponadpodstawowej.

Ogólne zasady dotyczące metodyki nauczania algorytmizacji

- Istotna jest kolejność wprowadzania podstawowych konstrukcji algorytmicznych i instrukcji w językach programowania: sekwencja zwana też algorytmem liniowym, selekcja (wybór), iteracja (pętla), rekurencja jako realizacja iteracji.
- Nauczanie przez rozwiązywanie zadań, które wymagają zastosowania pewnej liczby nowych elementów (konstrukcji algorytmicznych, instrukcji, typów danych, klasycznych algorytmów),
- Nauczanie przez analizę gotowych schematów blokowych, programów.

W szkole podstawowej materiał przedstawiony powinien być w następującej kolejności:

- Pojęcie algorytmu, przykład algorytmów z życia codziennego.
- Opisanie algorytmów w postaci ciągu kroków.
- Przedstawienie algorytmu w postaci schematów blokowych, wykorzystanie programu edukacyjnego DIA.
- Zapis algorytmu w postaci schematów blokowych w prostym języku programowania stworzonym na potrzeby edukacyjne jakim jest LOGO Komeniusz lub w innym języku programowania.

2. Algorytmy w postaci listy kroków

Algorytm – skończony ciąg jasno zdefiniowanych czynności (kroków), koniecznych do wykonania pewnego rodzaju zadań. Sposób postępowania prowadzący do rozwiązania problemu. Zadaniem algorytmu jest przeprowadzenie systemu z pewnego stanu początkowego do pożądanego stanu końcowego. Badaniem algorytmów zajmuje się algorytmika. Algorytm może zostać zaimplementowany w postaci programu komputerowego.

Jako przykład stosowanego w życiu codziennym algorytmu podaje się często przepis kulinarny.

Żeby przetestować listę kroków warto zacząć od algorytmów niewymagających użycia zmiennych i wprowadzać kolejno konstrukcje sekwencji, selekcji i iteracji. Należy zwrócić uwagę na formę gramatyczną wypowiedzianych rozkazów ma być ona w trybie rozkazującym, a także na użycie takich słów jak: **dopóki**, **powtarzaj** czy **jeśli**. Dalej przedstawiono przykładowe treści zadania wraz z ich rozwiązaniami.

Zadanie (sekwencja). Przedstaw w postaci ciągu kroków algorytm konstruowania symetralnej odcinka. Odcinek o końcach A i B zostanie narysowany na kartce leżącej przed nami, do dyspozycji mamy cyrkiel, ołówek i linijkę.

Rozwiązanie:

1. Weź cyrkiel do ręki;
2. Wbij nóżkę cyrkla w punkt A;
3. Rozłóż cyrkiel na odległość odcinka;
4. Narysuj okrąg;
5. Wyjmij nóżkę cyrkla z punktu A;
6. Włóż nóżkę cyrkla w punkt B;
7. Narysuj okrąg;
8. Odłóż cyrkiel
9. Weź ołówek;
10. Połącz punkty przecięcia się okręgów;
11. Odłóż wszystkie przyrządy.

Zadanie (selekcja). Przedstaw w postaci ciągu kroków algorytm wyjścia z domu Pani Kowalskiej. Pani Kowalska zna pogodę z TV, dzień może być deszczowy i chłodny. Pani Kowalska stoi w przedpokoju ubrana, ale bez kurtki, kurtka i parasol wiszą na wieszaku, klucze ma w prawej kieszeni kurtki, mieszka w bloku.

Rozwiązanie:

1. Nałóż kurtkę;
2. Weź parasol do lewej ręki;
3. Wyjdź z mieszkania;
4. Jeśli ktoś jest w domu, przejdź do punktu 8;
5. Wyjmij klucz;
6. Zamknij drzwi;
7. Schowaj klucz;
8. Wyjdź na zewnątrz;
9. Jeśli pada deszcz rozłóż parasol;

Zadanie (iteracja). Przedstaw w postaci ciągu kroków algorytm dokonywania zakupów w sklepie samoobsługowym od momentu wejścia do sklepu do wyjścia. Zakładamy, że nie ma kolejek do koszyków ani do kasy. Klient ma ze sobą spis zakupów do wykonania.

Rozwiązanie:

1. Weź koszyk;
2. Dopóki są jeszcze zakupy na kartce wykonuj:
3. Odszukaj regał, na którym powinien być dany towar;
4. Podejdź do regału;
5. Jeśli towar jest na regale, to przejdź do punktu 6, w przeciwnym razie do punktu 7;
6. Przełóż towar do koszyka;
7. Przejdź do kasy;
8. Jeśli masz zakupy w koszyku, zapłać za towar, w przeciwnym razie przejdź do punktu 10;
9. Przełóż zakupy do siatki;
10. Odłóż koszyk.

Następnie należałoby wprowadzić intuicyjne pojęcie zmiennej jako „pojemnika” czy też „pudełka” z zawartością oraz nazwą i znów przedstawić przykład w kolejności: sekwencja, selekcja czy iteracja. Warto równolegle z wprowadzeniem zmiennych wprowadzić instrukcję podstawienia :=.

Zadanie (sekwencja). Przedstaw w postaci ciągu kroków algorytm obliczania pola i obwodu kwadratu o długości krawędzi zadanej jako zmienna a , wynik umieść odpowiednio w zmiennych P i O .

Rozwiązanie:

1. $P := a*a$;
2. $O := 4*a$.

Zadanie (selekcja). Przedstaw w postaci ciągu kroków algorytm obliczaniu mniejszej z liczb zadanych w zmiennych a i b , wynik podstaw w zmiennej min .

Rozwiązanie:

1. Jeśli $a > b$ przejdź do punktu 2, w przeciwnym razie przejdź do punktu 3;
2. $min := b$;
3. $min := a$.

Zadanie (iteracja). Przedstaw w postaci ciągu kroków algorytm obliczania największego wspólnego dzielnika NWD dla liczb naturalnych danych zmiennych liczba1 i liczba2, wynik podstaw do zmiennej nwd.

Rozwiązanie:

1. Dopóki liczba1 nie jest równa liczbie2 powtarzaj krok 2, w przeciwnym razie przejdź do kroku 3;
2. Od większej odejmij mniejszą i tę większą zastąp otrzymaną różnicą.
3. nwd:= liczba1.




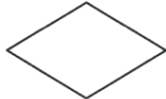
2. Schematy blokowe

Kolejny etap w nauczaniu informatyki to przedstawienie algorytmów w postaci *schematów blokowych*. Schematy blokowe służą do graficznej prezentacji algorytmów. W celu poprawienia czytelności opisywanych algorytmów można stosować schematy NS (zwane też schematami zwartymi). W opinii autorów skryptu schematy NS są dużo bardziej praktyczne w nauce programowania ze względu na swoją czytelność, konsekwencję zapisu i uporządkowanie, w przeciwieństwie do schematów blokowych, w których łatwo (szczególnie w przypadku początkującego adepta programowania) o chaos i nieczytelność. Jednakże w szkołach uczniowie muszą zostać nauczeni schematów blokowych, których znajomość (i umiejętność tworzenia) będzie od nich w dalszych etapach nauki (a w szczególności na egzaminie maturalnym z informatyki) wymagana stąd w niniejszym materiale konieczność ich wprowadzenia.

Zadaniem schematów blokowych jest przedstawienie kolejnych etapów wykonania zadania w postaci skrzynek (mają one kształt figur geometrycznych) połączonych liniami zakończonymi strzałkami.

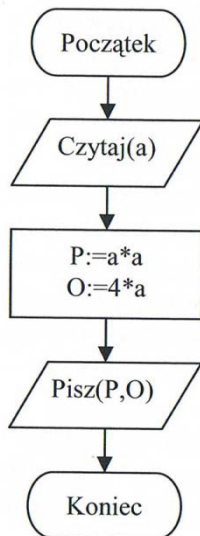
Przykładowe kształty skrzynek przedstawia poniższa tabela:

Tabela Podstawowe skrzynki do prezentacji algorytmu w postaci graficznej

	Skrzynka graniczna oznaczająca koniec lub początek algorytmu.
	Skrzynka operacyjna (obliczeniowa) służy do wykonywania różnych działań, np. sumowania $S:=a+b$.
	Skrzynka wejścia/wyjścia służy do wprowadzania (czytania) danych; lub wyprowadzania (pisanie, drukowanie) wyników.
	Skrzynka warunkowa (decyzyjna) służy do sprawdzania warunków, np. czy $N>0$.

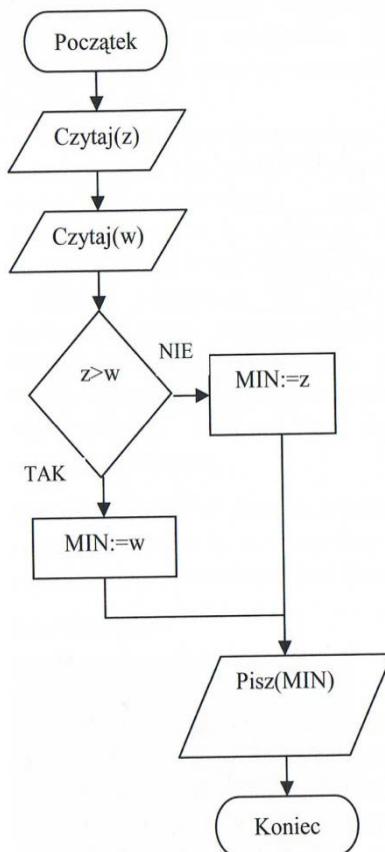
Zadanie (sekwencja). Przedstaw w postaci schematu blokowego algorytm, który pozyskuje od użytkownika wartość zmiennej a będącej długością boku kwadratu, następnie zmiennej P nadaje wartość równą polu kwadratu, a zmiennej O nadaje wartość równą obwodu kwadratu o boku długości a i wyświetla wartości zmiennych P i O .

Rozwiązanie:



Zadanie (seleksja). Przedstaw w postaci schematu blokowego algorytm, który pozyskuje od użytkownika dwie liczby rzeczywiste umieszczone w zmiennych z i w , następnie zmiennej MIN nadaje wartość mniejszej z liczb zadanych w zmiennych z oraz w . Następnie wartość zmiennej MIN jest wyświetlana na ekranie.

Rozwiązanie:



Zadanie (iteracja). Przedstaw w postaci schematu blokowego algorytm, który pobiera od użytkownika dwie liczby naturalne dodatnie dane w zmienną g i h. Następnie wyznacz ich największy wspólny dzielnik (NWD), wykorzystując algorytm Euklidesa.

Rozwiązanie: Wartość NWD po wykonaniu pętli znajduje się w zmiennej g lub h.

