

T: Programowanie i techniki algorytmiczne – wprowadzenie do programowania

Aby przedstawić algorytm w postaci programu komputerowego, trzeba go zapisać jako ciąg instrukcji języka programowania.

Każda instrukcja programu, podobnie jak figura (blok), odpowiada odpowiedniej operacji. Dlatego kolejność występowania instrukcji w programie decyduje o kolejności wykonywanej operacji

Ze względu na poziom wykonywania programu języki, programowanie możemy podzielić na:

- **język wysokiego poziomu**, np. Java, C, C++, Python, Visual Basic, Pascal, JavaScript
- **języki niskiego poziomu** (wewnętrzne)

Program napisany w języku wysokiego poziomu nie jest zrozumiały dla komputera. Komputer potrafi wykonywać tylko instrukcje (polecenia) zapisane w języku wewnętrznym, zrozumiałym dla procesora.

To tak jakby, ktoś mówił do Was w innym języku a Wy rozpoznajecie jedynie język polski. Potrzebujemy zatem nauczyć się nowego języka lub znaleźć sposób na jego przetłumaczenie.

Proces tłumaczenia programu napisanego w języku programowania wysokiego poziomu w język wewnętrzny komputera nazywamy translacją. Może on przebiegać w formie kompilacji lub interpretacji.

Kompilacja to przetłumaczenie całego programu na język zrozumiały dla procesora, aby można było go wykonywać na komputerze.

Interpretacja to tłumaczenie programu w jednym z języków programowania, instrukcja po instrukcji, komputer wykonuje każdą instrukcję pojedynczo.

Tłumaczeniem programu zajmuje się program, tzw. translator. Tłumaczenie programu połączone jest ze sprawdzaniem poprawności składniowej zapisanych instrukcji. Jeśli instrukcja jest napisana niepoprawnie, to nie zostanie przetłumaczona i pojawi się komunikat o błędzie. Trzeba wówczas ją odszukać, poprawić i powtórnie przeprowadzić translację.

Środowiska programowania

Aby stworzyć program, musimy skorzystać z wyspecjalizowanych pakietów programistycznych, zawierających zwykle edytor kodu źródłowego, a także kompilator i inne narzędzia wspomagające programowanie.

Przykładami takich środowisk mogą być: Microsoft Visual Studio, Delphi, JBuilder.

Niezależnie od zastosowanego środowiska:

- *kolejność występowania instrukcji w programie powinna odpowiadać kolejności operacji realizujących dany algorytm*
- *postać instrukcji musi być zgodna z zasadami składni danego języka – nie może zabraknąć nawet jednej spacji, jednego dwukropka, przecinka, czy innego znaku w miejscach, w których są wymagane*

Zmienne w programie

Zmiennym wykorzystywanym w programie przyporządkowane są określone komórki pamięci. W wielu językach programowania (np. Pascal i C++) zmienne przed użyciem należy zadeklarować. Deklaracja zmiennej pozwala kompilatorowi dokonać przydziału pamięci operacyjnej dla tej zmiennej.

W deklaracji zmiennej podaje się jej nazwę oraz typ (określający rodzaj danych przechowywanych w zmiennych – mogą to być np. liczby, znaki, wartości logiczne). Deklaruje się zmienne, w których będą przechowywane dane wejściowe dla programu, wyniki działania programu, a także dane pomocnicze niezbędne do wykonania obliczeń.

Sytuacje warunkowe

Z sytuacjami warunkowymi spotykamy się w każdej dziedzinie wiedzy i życia codziennego.

Na pytanie „Czy pada deszcz?” odpowiedź może brzmieć „tak” lub „nie”.

W zależności od tego, czy warunek jest spełniony czy nie, wybieramy odpowiedni sposób postępowania



Sytuacja warunkowa występuje wtedy, gdy wynik lub dalsze działanie zależy od spełnienia (lub niespełnienia) pewnego warunku. Algorytm zawierający sytuacje warunkowe nazywamy algorytmem z warunkami.

Interacja, czyli powtarzanie poleceń polega na wielokrotnym powtarzaniu tej samej operacji (ciągu operacji).

Iteracje. implementujemy (piszemy kod źródłowy), stosując tzw. pętlę.

Biegając wokół stadionu pokonujemy za każdym razem dystans 400 m (zawsze taki sam), kolejne przekroczenie linii mety, powoduje powrót do już wcześniej wykonywanych działań (kroków), które przy dalszym bieganiu wokół stadionu mogą zostać wykonane wiele razy.

W realizacji algorytmów iteracyjnych ważne jest prawidłowe określenie sposobu zakończenia działań. Niepoprawne określenie tych warunków może spowodować, że obliczenia nigdy się nie zakończą, czyli nastąpi zapętlenie algorytmu.

Realizując dowolne zadanie musimy wiedzieć jak go zakończymy, Jeśli tego nie określimy nasze działania będą chaotyczne i „zapętłone”.

- Spotkajmy się? - Gdzie? - Gdzie chcesz? - O której? - O której chcesz?

- Dobra, tylko nie spóźnij się!

Ważniejsze zagadnienia do zapamiętania:

- ☑ Programowanie polega na przedstawieniu algorytmu w postaci instrukcji języka programowania, w kolejności wyznaczonej przez ten algorytm.
- ☑ Komputer wykonuje tylko instrukcje (rozказы) zapisane w języku wewnętrznym.
- ☑ Aby program był zrozumiały dla komputera, musi zostać przetłumaczony z języka wysokiego poziomu na język wewnętrzny komputera. Proces tłumaczenia nazywamy translacją. Może ona przebiegać w formie kompilacji lub interpretacji.
- ☑ Do tworzenia programów możemy stosować dydaktyczne środowiska programowania lub wyspecjalizowane pakiety programistyczne.
- ☑ Zmiennym wykorzystywanym w programie przyporządkowane są określone komórki pamięci operacyjnej o unikatowym adresie. Poza miejscem w pamięci ze zmienną związana jest zawsze jakaś wartość.
- ☑ Problem możemy podzielić na problemy cząstkowe, a następnie każdy z nich przedstawić w postaci oddzielnego podprogramu.
- ☑ W algorytmie z warunkami występują sytuacje warunkowe – wynik lub dalsze działanie algorytmu zależą od spełnienia określonego warunku.

Na teraz to wszystko o programowaniu, na kilka zajęć przed zakończeniem edukacji w klasie siódmej, nie chcę byście wygrywali kolejne programy zajmujące trochę pamięci. Do programowania wróćcie jeszcze w klasie ósmej.

Od następnej lekcji przechodzimy do jednego z programów z pakietu Microsoft Office – do Excel'a.

Jeżeli ktoś nie posiada tego pakietu biurowego to proszę o skorzystanie z jego darmowych odmian, a ich opis i bezpieczną drogę ściągnięcia znajdzie pod tym adresem:

<https://www.komputerswiat.pl/poradniki/programy/najlepsze-darmowe-pakiety-biurowe-alternatywy-dla-microsoft-office/b49k6fn>