

Języki programowania

Programowanie

to proces projektowania, tworzenia, testowania i utrzymywania kodu źródłowego programów komputerowych lub urządzeń mikroprocesorowych (mikrokontrolery).

Do przedstawiania algorytmów w postaci programów służą **języki programowania**.

Język programowania = zasady składni + instrukcje

Niezależnie od stosowanego języka programowania należy pamiętać, że:

- Język programowania jest językiem formalnym, co oznacza, że podlega jednoznacznym regułom. Cechy takiej nie mają języki naturalne, gdzie czasem sens zdania zależy od kontekstu, w którym zostało ono użyte, lub od naszej interpretacji.
- Postać instrukcji, w tym słów kluczowych, musi być bardzo precyzyjna - zgodna z zasadami składni. W programie nie może zabraknąć ani jednego koniecznego znaku. Tłumaczenie programu na kod maszynowy połączone jest ze sprawdzaniem poprawności składniowej zapisanych instrukcji. Instrukcje błędnie zapisane nie będą mogły zostać przetłumaczone, a tym samym program nie zostanie wykonany.
- Kolejność zapisywania instrukcji powinna odpowiadać kolejności operacji realizujących dany algorytm (podobnie jak umieszczanie bloków w schemacie blokowym). Program jest bowiem wykonywany zgodnie z zapisem.

RODZAJE JEZYKÓW

Podział ze względu na proces zamiany na kod maszynowy:

1. Program napisany w **języku kompilowanym**, zanim zostanie uruchomiony musi zostać skompilowany do kodu maszynowego.
2. Programy napisane w **językach interpretowanych**, są wykonywane przez interpreter linijka po linijce i nie muszą być wcześniej skompilowane.

Podział ze względu na poziom uproszczenia w kodowaniu:

1. **Niskiego poziomu** (np. ASSEMBLER język wymagający od programisty doskonałej znajomości sprzętu, na który pisany jest program.)
2. **Wysokiego poziomu** (narzędzia programistyczne w znacznym stopniu ułatwiają pisanie kodu i wyszukiwanie błędów programowanie w najwyższym poziomie może się ograniczać do programowania graficznego przy użyciu myszy)

Ze względu na kryterium przydatności:

a) Imperatywne - to metoda programowania, w której program składa się ze zmiennych oraz modyfikujących je operacji, z jawnie określonym przepływem sterowania. Typowe języki tego typu to C, Pascal.

Największą zaletą jest bliskość takiej reprezentacji do tego co rzeczywiście wykonywane jest przez komputer.

b) Funkcyjne - filozofia programowania będąca odmianą programowania deklaratywnego, w której funkcje należą do wartości podstawowych, a nacisk kładzie się na wartościowanie (często rekurencyjnych) funkcji, a nie na wykonywanie poleceń.

c) Logiczne - to metoda programowania, w której program podawany jest jako pewien zestaw zależności, a

obliczenia są dowodem pewnego twierdzenia w oparciu o te zależności.

d) Obiektowe - paradygmat¹ programowania, w którym programy definiuje się za pomocą obiektów — elementów łączących *stan* (czyli dane, nazywane najczęściej polami) i *zachowanie* (czyli procedury, tu: metody). Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.

Pod względem stopnia zaawansowania na języki generacji:

a) Pierwszej - są to języki maszynowe, czyli języki procesorów. Instrukcje są w nich zapisywane w postaci liczb binarnych.

Przykładowy kod programu:

```
11101010000000000000001111111111111111111000100111  
01100000000010100000001
```

wbrew pozorom ma to jakiś sens.

b) Drugiej - języki symboliczne, asemblery. Języki niskiego poziomu, pod względem składni tożsame z maszynowymi, z tą różnicą że zamiast liczb używa się tu łatwiejszych do zapamiętania mnemoników². Należy dodać, iż pod względem znaczenia kod ten jest jednoznaczny z kodem binarnym przedstawionym w poprzednim punkcie (tylko dla procesorów z grupy 80x86).

¹ paradygmat - wzorzec programowania przedkładany w danym okresie rozwoju informatyki ponad inne lub szczególnie ceniony w pewnych okolicznościach lub zastosowaniach.

² mnemoniki – to składający się z kilku liter kod-słowo, które oznacza konkretną czynność procesora. Przykładem mogą być: „add” (ang.dodaj) czy „sub” (ang. subtract - odejmij).

c) Trzeciej - języki wysokiego poziomu, proceduralne (imperatywne). W językach tych jedna instrukcja jest tłumaczona na kilka instrukcji procesora, najczęściej od 5 do 10. Pierwszym językiem tego typu był ALGOL. Do tej grupy należą między innymi: FORTH, BASIC - języki niestrukturalne, Pascal, C, FORTRAN - języki strukturalne, C++, Java - języki zorientowane obiektowo.

d) Czwartej - języki bardzo wysokiego poziomu, nieproceduralne (deklaratywne). Korzystając z tych języków programista skupia się na problemie, a nie na sposobie jego rozwiązania. Syntaktyka wielu języków czwartej generacji przypomina składnię języka naturalnego. Są one często używane do dostępu do baz danych. Przykładem języka z tej grupy jest: SQL.

e) Piątej - języki sztucznej inteligencji, języki systemów ekspertowych. Języki najbardziej zbliżone do języka naturalnego. Przykładem języka piątej generacji jest PROLOG.