

Języki programowania

Piotr Dobosz

Czym jest język

- Określony sposób kodowania wiadomości
- Postać werbalna i/lub graficzna
- Język naturalny a język formalny
- Język formalny jako podstawowy język informatyczny

Cechy języka formalnego

- Skończony zbiór symboli
- Ze zbioru tworzy się ~~skończoną~~ ilość słów zawierających się we wcześniej wspomnianym zbiorze symboli
- Język posiada swoją składnię i gramatykę
- Może być podzbiorem języka naturalnego (ale NIE MUSI!)

Język programowania

- Język formalny
- Zawiera zestaw instrukcji
- Dzięki niemu możemy otrzymać na wyjściu określone wyniki
- Obecnie istnieją tysiące języków programowania

Cechy języka programowania

- Język programowania to notacja językowa pozwalająca na zapisanie programu, którego celem będzie wykonanie określonego algorytmu przetwarzania
- Język programowania musi mieć funkcję i cel
- Język programowania musi posiadać poziom abstrakcji
- Język programowania musi obsługiwać wyrażenia

Elementy języka programowania

- Składnia
- Semantyka (statyczna i dynamiczna)
- System typów
 - a) Typowanie i języki beztypowe
 - b) Statyczne i dynamiczne typowanie
 - c) Typowanie słabe i mocne
- Biblioteki (standardowe i uruchomieniowe)

Klasyfikacja języków programowania

- Języki programowania można klasyfikować po ich własnościach
- Języki programowania można klasyfikować po ich poziomie
- Języki programowania można klasyfikować po ich przeznaczeniu

Kod maszynowy (poziom pierwszy)

- Zawiera jedynie zera i jedynki
- Każdy mikroprocesor posiada RÓŻNE słowa do wykonywania (nie ma jednolitego "języka" maszynowego)
- Różnice w architekturze (Little Endian, Big Endian, Endianess)
- Praktycznie kod maszynowy to sekwencja zer i jedynek
- Chcąc zapisać zera i jedynki programiści wykorzystują liczby szesnastkowe

```
89 F8 A9 01 00 00 00 75 06 6B C0 03 FF C0 C3 C1 E0 02 83 E8 03 C3
```

Assembler (poziom 1)

- "składa" program z prostych instrukcji
- Kody szesnastkowe zostały zamienione na tzw. Mnemoniki (termin z psychologii i pedagogiki) - nazwy rozkazów mikroprocesora
- Nadal ściśle powiązanie z architekturą

```
.globl f
.text
f:
    mov    %edi, %eax    # Put first parameter into eax register
    test  $1, %eax      # Examine least significant bit
    jnz   odd           # If it's not a zero, jump to odd
    imul  $3, %eax      # It's even, so multiply it by 3
    inc   %eax          # and add 1
    ret                                # and return it
odd:
    shl   $2, %eax      # It's odd, so multiply by 4
    sub   $3, %eax      # and subtract 3
    ret                                # and return it
```

Języki proceduralne (poziom 2)

- Posiadają "uniwersalną" składnię, niezależną od platformy
- Składnia przypomina ludzkie słowa i zdania
- Posiadają złożoną składnię
- Pozwalają na pisanie procedur (funkcji), a nawet struktur czy klas
- Pomimo wszystko, pisane w nich aplikacje mogą nadal mieć silny związek z układem elektronicznym (wstawki assembler, sterowanie układami itp.)
- Języki z grupy: C, C++

```
int f(const int n) {  
    return (n % 2 == 0) ? 3 * n + 1 : 4 * n - 3;  
}
```

Języki obiektowe (poziom 3)

- Charakteryzują się przewagą typów złożonych (obektowych)
- Ideowo powinny posiadać jedynie obiekty
- Często oderwane od konkretnej platformy sprzętowej; operują jedynie na bibliotekach pośredniczących (bądź działają w środowiskach wirtualnych)
- Zawierają cechy języków poziomu drugiego

```
class ThingThatHoldsTheFunctionUsedInTheExampleOnThisPage {  
    public static int f(int n) {  
        return (n % 2 == 0) ? 3 * n + 1 : 4 * n - 3;  
    }  
}
```

Języki skryptowe (poziom 4)

- Ściśle związane ze środowiskiem, w którym będą się wykonywać
- Kwalifikowane jako wysokiego poziomu
- Najczęściej międzyplatformowe (poza skryptami w systemach operacyjnych)
- Najczęściej obiektowe (poza skryptami w systemach operacyjnych)
- Dynamiczne alokacje zmiennych, dynamiczne typy, automatyczna konwersja typów (najczęściej niejawną)

Języki wyspecjalizowane (poziom 5)

- Przeważnie posiadają specyficzną dla zadania składnię
- Działają na abstrakcyjnych zasobach, nie związanych ze sprzętem czy system operacyjnym
- Głównie do sprawdzania lub zarządzania
- Przykładem może być język SQL lub Lisp

Języki wizualne (poziom 6)

- Języki tego typu nie posiadają składni tekstowej (lub posiadają jej niezbędne minimum)
- Najczęściej w postaci wysokopoziomowej
- Nie zawsze języki tego typu muszą odnosić się stricte do programowania
- Przykładem tego typu języków może być Scratch, Blockly, Logo

Języki ezoteryczne (poziom 7)

- Języki te nie mają najczęściej znaczącej roli w programowaniu
- Często składają się z minimalnej ilości zbioru znaków
- Niektóre z tych języków pozwalają na budowę kompletnych aplikacji, chociaż większość z nich może stanowić jedynie ciekawostkę

```
+++++ [ >+++++>+++++>+++<<<- ]>+ .>+ .+++++  
 . .+++ .>+ .<<+++++ .> .+++ .----- .----- .>+ .
```

Inne rodzaje klasyfikacji języków

- Języki maszynowe
- Języki assemblerowe
- Języki wysokiego poziomu
- Języki systemowe
- Języki skryptowe
- Języki wyspecjalizowane (określonego przeznaczenia)
- Języki wizualne
- Języki ezoteryczne

Inna klasyfikacja języków programowania

- Języki niskiego poziomu:
 - 1) Kod maszynowy
 - 2) Języki assembler
- Języki wysokiego poziomu:
 - 1) C/C++
 - 2) Java
 - 3) Python
 - 4) JavaScript

Materiały i odnośniki

- https://en.wikipedia.org/wiki/Formal_language
- https://en.wikipedia.org/wiki/Programming_language
- <https://cs.lmu.edu/~ray/notes/pltypes/>