

## Języki programowania

**Język programowania** pozwala programiście na precyzyjne przekazanie maszynie, jakie dane mają ulec obróbce i jakie czynności należy podjąć w określonych warunkach.

**Język maszynowy** – język programowania, w którym zapis programu wymaga instrukcji bezpośrednio jako liczb, które są rozkazami i danymi bezpośrednio pobieranymi przez procesor wykonujący ten program.

**Kod maszynowy** to postać programu komputerowego (zwana postacią wykonywalną lub binarną) przeznaczona do bezpośredniego lub prawie bezpośredniego wykonania przez procesor, przeznaczona na konkretny typ procesora (i zgodne z nim) wyrażona w postaci kodów rozkazów procesora i ich argumentów, praktycznie nieczytelna dla człowieka.

W systemach z rodziny DOS/Windows w kodzie maszynowym zapisane są programy w plikach typu "com" oraz "exe", w systemach unixowych są to pliki typu „elf”

Kod maszynowy jest (współcześnie) generowany za pomocą translatora (kompilatora lub interpretera), który przekształca czytelny dla programisty kod w języku programowania w kody instrukcji maszynowych.

Kod maszynowy nie może być wykonywany przez procesory innego typu. Niesie to za sobą wiele problemów związanych z przenośnością programów.

## Języki niskiego poziomu

Język niskiego poziomu – typ języka programowania, który w małym stopniu abstrahuje od konstrukcji jednostki centralnej komputera. Innymi słowy, język ten wykazuje duże podobieństwo do kodu maszynowego, zaś kompilacja jest w miarę nieskomplikowana.

Najbardziej typowym przykładem języka niskiego poziomu są asemblery.

W assemblerach zasadniczo jedno polecenie odpowiada jednemu rozkazowi procesora. Są to języki powstałe na bazie języka maszynowego poprzez zastąpienie liczb odpowiadających fragmentom rozkazów kodu maszynowego ich symbolicznymi odpowiednikami. Dzięki zamianie liczb na tzw. mnemoniki można pisać programy w miarę zrozumiałe dla człowieka, a jednocześnie bezpośrednio tłumaczone na kod maszynowy procesora, co pozwala zapewnić duży stopień kontroli programisty nad zachowaniem procesora.

## Przykład:

### pseudokod:

```
pętla od i = 1 do n  
    (instrukcje w pętli)  
kolejne instrukcje
```

### assembler:

```
        LD  0,i    (załaduj stałą 0 pod adres i)  
PEŃTLA  CMP  n,i    (porównaj wartości pod adresami n oraz i)  
        IF  RESZTA(jeśli równe, to skocz do instrukcji o etykiecie RESZTA)  
        ADD 1,i    (dodaj stałą 1 do wartości pod adresem i)  
        (...)    (instrukcje w pętli)  
        JMP PEŃTLA (skocz do instrukcji o etykiecie pętla)  
RESZTA (kolejne instrukcje programu)
```

Współcześnie praktycznie nie używa się asemblera do pisania całych programów dla komputerów osobistych. Jest on za to wciąż używany do pisania fragmentów wymagających bardzo wysokiej wydajności lub mających inne specjalne wymagania, np. dla oprogramowania mikrokontrolerów o niewielkich rozmiarach pamięci programu.

## **Języki wysokiego poziomu**

Język wysokiego poziomu – typ języka programowania, którego składnia i słowa kluczowe mają maksymalnie ułatwić rozumienie kodu programu dla człowieka, tym samym zwiększając poziom abstrakcji i dystansując się od sprzętowych niuansów.

Kod napisany w języku wysokiego poziomu nie jest bezpośrednio „zrozumiały” dla komputera – większość kodu stanowią tak naprawdę normalne słowa, np. w języku angielskim. Aby umożliwić wykonanie programu napisanego w tym języku należy dokonać kompilacji (lub interpretacji).

Język proceduralny (ang. procedural language) – język programowania umożliwiający tworzenie oprogramowania w postaci programu głównego oraz wielu procedur (lub podprogramów), z których każda realizuje określoną funkcję i może być wywoływana wielokrotnie przez program główny.

Język nieproceduralny (ang. non-procedural language) - język programowania pozwalający na pisanie programów jedynie w postaci ciągłej sekwencji instrukcji (niepodzielonej na procedury).

Programowanie obiektowe (ang. object-oriented programming) to metodologia tworzenia programów komputerowych, która definiuje programy za pomocą "obiektów" – elementów łączących stan (czyli dane) i zachowanie (czyli procedury, tu: metody). Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań. Podejście to różni się od tradycyjnego programowania proceduralnego, gdzie dane i procedury nie są ze sobą bezpośrednio związane. Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.

*[historia języków programowania]*

## FORTRAN (Formula Translator) (1957)

Pierwszy język programowania wysokiego poziomu, stworzony w latach 50. i nadal będący w użyciu. Jego głównym zastosowaniem są obliczenia naukowe i analiza numeryczna oraz arytmetyczna.

### FORTRAN I-IV, 66, 77

```
c  deklaracje zmiennych, stałych, tablic
integer a, i, n
real c, d(100), e
double precision f
parameter (f = 1000.0d0)
```

```
d(1) = 5.5d2
```

```
c  instrukcje wejścia/wyjścia
read(*,*) a
a = a + 5
write(*,*) a
```



```
c  instrukcja warunkowa
  if (a .gt. 0) then
    a=a+1
    c=a**3
  else
    a=a-1
    c=a/3
  endif
```

```
c  pętle
  do 55 i=1, 100
    write(*,*) d(i)
55 continue
```

```
  i=1
15 if (i .ls. 100) then
  write(*,*) d(i)
  i=i+1
  goto 15
endif
```

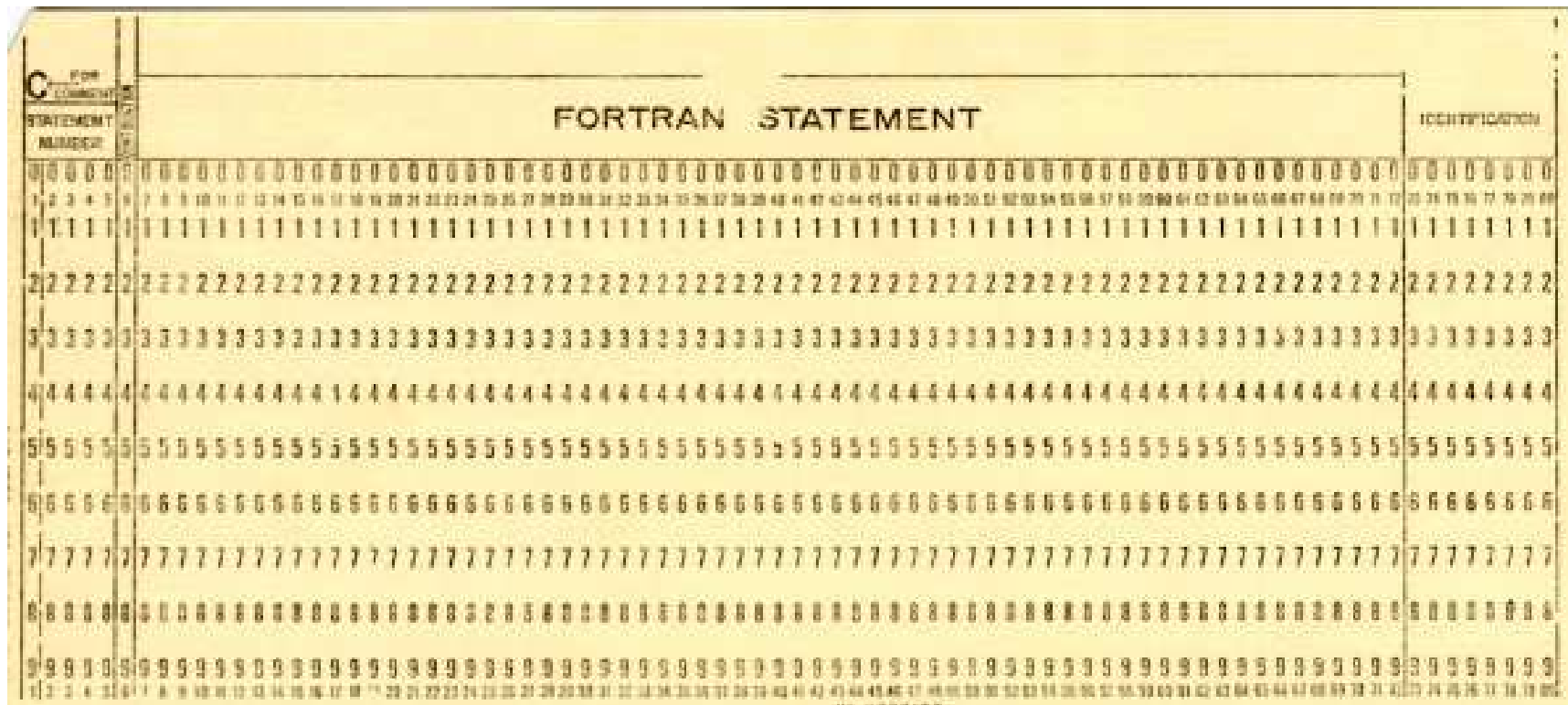
```
c  WERSJA DLA FORTRAN I
  if (a .gt. 0) goto 10
  a=a-1
  c=a/3
  goto 20
10 a=a+1
  c=a**3
20 continue
```

```
c niektóre kompilatory akceptują:
  do i=1,100
    (...)
  enddo
```

```
c  procedura
   subroutine licz(n, x)
   (...)
   end subroutine
```

```
c  funkcja
   real function silnia(n)
   (...)
   silnia = (...)
   end function
```

# Karta do wprowadzania kodu



## FORTRAN 90, 95, 2003, 2008

wiele udoskonaleń, unowocześniona składnia, wiele nowych standardowych funkcji (np. operacje na macierzach), dynamiczna alokacja tablic

# COBOL (Common Business Oriented Language) (1959)

przeciwieństwo Fortranu – język przeznaczony do prac programistycznych w dziedzinach ekonomii i biznesu.

```
IDENTIFICATION DIVISION.
```

```
    Program-Id. BazaDanych.
```

```
*
```

```
ENVIRONMENT DIVISION.
```

```
*
```

```
DATA DIVISION.
```

```
*
```

```
01 PLIK-UNIWERSYTECKI
```

```
02 STUDENT occurs 100 times
```

```
    03 NAZWISKO pic A(15)
```

```
    03 PRZEDMIOT occurs 30 times
```

```
        04 NAZWA-PRZEDMIOTU pic AAAA999
```

```
        04 OCENA pic 9
```

```
    03 NUMER-ALBUMU pic 99999
```

```
02 WYDZIAL occurs 20 times
```

```
    03 NAZWA-WYDZIALU pic A(50)
```

```
    03 PRZEDMIOT occurs 80 times
```

```
        04 NAZWA-PRZEDMIOTU pic AAAA999
```

```
        04 PROWADZĄCY pic A(20)
```

PROCEDURE DIVISION.

Sortuj.

DISPLAY "Sorting...".

SORT PLIK-UNIWERSYTECKI ON KEY STUDENT.

\*

Stop Run.

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

MULTIPLY B BY B GIVING B-SQUARED.

MULTIPLY 4 BY A GIVING FOUR-A.

MULTIPLY FOUR-A BY C GIVING FOUR-A-C.

SUBTRACT FOUR-A-C FROM B-SQUARED GIVING RESULT-1.

COMPUTE RESULT-2 = RESULT-1 \*\* .5.

SUBTRACT B FROM RESULT-2 GIVING NUMERATOR.

MULTIPLY 2 BY A GIVING DENOMINATOR.

DIVIDE NUMERATOR BY DENOMINATOR GIVING X.

## Złośliwości:

“Compiles Only Because Of Luck”,

“Completely Over and Beyond reason Or Logic”,

“Completely Obsolete Business Oriented Language”

## Pascal (1970)

Dawniej jeden z najpopularniejszych języków programowania, uniwersalny, ogólnego zastosowania. Pierwotnie służył celom edukacyjnym do nauki programowania strukturalnego.

```
c := 5;
a := a+5;

for i:=1 to 10 do
begin
  (...)
end;

while(w = 0) do
begin
  (...)
end;

repeat
  (...)
until (w > 0);

if a > 0 then b := 2;
```

Zalety Pascala – czytelność kodu, rygorystyczna kontrola typów danych, wraz z pojawieniem się C stały się dla programistów wadami, m.in. przez co drastycznie spadła jego popularność. Popularność Pascala wzrosła ponownie z pojawieniem się Delphi – środowiska programistycznego, opartego na obiektowym Pascalu, pozwalającego na błyskawiczne tworzenie atrakcyjnych wizualnie aplikacji pod Windows.

## Prolog (Programming In Logic) (1971)

Język opiera się na klauzulach, stwierdzających że pewne fakty logiczne wynikają ze zbioru innych faktów.

Program składa się ze zbioru znanych (tzn. podanych przez programistę) faktów, na podstawie których program usiłuje odpowiadać na zapytania.

```
ZJADA(koty, myszy)
```

```
ZJADA(psy, koty)
```

```
ZJADA(myszy, żaby)
```

```
ZJADA(x, y) :- zjada(x,z), zjada(z,y)
```

```
?-ZJADA(kto, żaby)
```

## Lisp (List Processing) (1958)

Język oparty na przetwarzaniu list

```
setq lista1 (list 1 2 'abc)  
(setf (third lista1) 'def)
```

```
setq lista2 (list 1 2 (list 3 4))
```

```
(+ 1 2 3 4)  
(+ lista2)
```



## Python (1990)

Nowoczesny, interpretowany, strukturalny język programowania. Może zająć miejsce Pascala w nauce programowania.

Bogata biblioteka funkcji i procedur.

```
if a > 0:  
    a=a+1  
    c=a*3  
else:  
    a=a-1  
    c=a/3
```

```
for i in range(1:n+1):  
    print i  
    print i*i
```

## Ada

Język ogólnego zastosowania, podobny w składni do mieszanki Fortranu, Pascala i C

## C / C++ / C# (C Sharp)

Nowoczesny język wszechstronnego zastosowania, dość trudna i często niespójna składnia.

W C++ / C# wprowadzono mechanizm obiektowości.

```
#include <iostream>
using namespace std;

double factorial(int);

int main()
{
    int n;
    cin >> n;
    cout << factorial(n);
}

double factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
}
```

## Smalltalk

Czysto obiektowy – wszystko jest obiektem, nawet liczby.

## Java

Język obiektowy, będący pochodną C++ i Smalltalka. Składnia bardzo podobna do C++, wyeliminowano pewne naleciałości z C.

„Wirtualna maszyna” – środowisko uruchomieniowe zapewniające przenośność kodu.

## PHP

Język programowania służący przede wszystkim do tworzenia dynamicznych stron WWW i wykonywany w tym przypadku po stronie serwera, z możliwością zagnieżdżania w HTML.

Składnia podobna do C++

## Perl (Practical Extraction and Report Language)

Perl został zaprojektowany jako praktyczne narzędzie do analizy plików tekstowych i tworzenia raportów.

```
$_ = "Wybrana liczba to 150\n";  
print;  
s/ (\d+) / $1 * 2 /ex;  
print;
```

## **Wizualne języki programowania**

Języki (środowiska programistyczne), pozwalający w prosty sposób programować aplikacje z graficznym interfejsem użytkownika.

Delphi, Visual C++, Tcl, Visual Basic

# Pseudojęzyki

## HTML

Język programowania składający się ze znaczników (ang. tags) stosowany do pisania stron WWW.

## TeX/LaTeX

System składania publikacji, szczególnie przydatny do składania tekstów naukowych.

## PostScript

Język komend dla drukarek

## XML

Język przechowywania danych