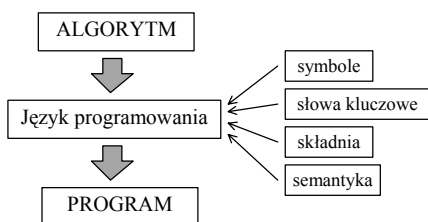


Języki programowania



Składnia typowego języka zawiera:

- warianty kilku struktur sterujących
- sposoby definiowania rozmaitych struktur danych
- wzorce podstawowych instrukcji

Algorytm sumowania liczb od 1 do N w hipotetycznym języku JP:

```

definiuj N, X, Y liczby całkowite
wczytaj N;
X := 0;
dla Y od 1 do N wykonaj
X := X + Y
koniec; wypisz X.
    
```

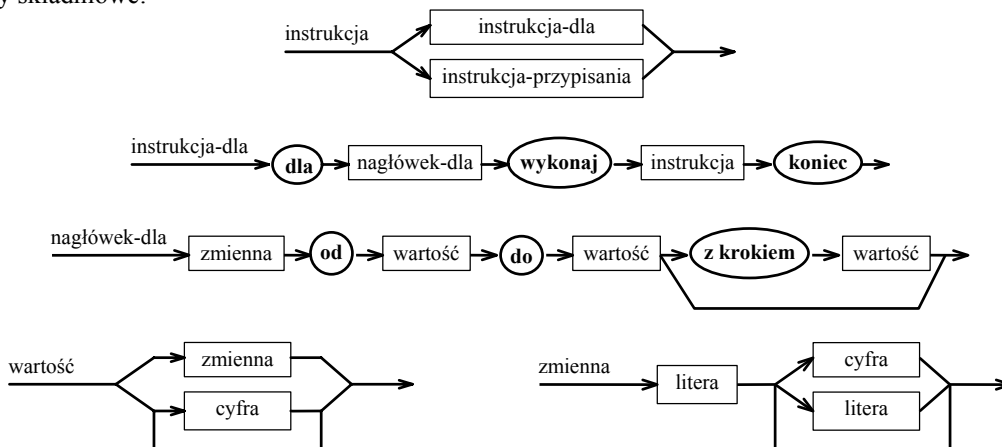
Słowa kluczowe: **definiuj, wczytaj, dla** itd.
 Instrukcje: przypisania - X := 0
 iteracji - **dla ... wykonaj ... koniec**

Definiowanie składni języka w notacji **BNF** (Backus-Naur Form):
 („|” oznacza „lub”)

```

<instrukcja> : <instrukcja-dla> | <instrukcja-przypisania> | ...
<instrukcja-dla> : dla <nagłówek-dla> wykonaj <instrukcja> koniec
<nagłówek-dla> : <zmienna> od <wartość> do <wartość>
<wartość> : <zmienna> | <liczba> | ...
    
```

Diagramy składniowe:



Definiowanie struktur danych: **definiuj TA tablica [1..50,8..107] w niej** liczby całkowite
 i odwołanie do elementu tablicy: TA[wartość,wartość]

Składnia języka programowania określa:

- jak opisywać struktury sterujące
- jak opisywać struktury danych
- jak tworzyć poprawne ciągi symboli dla nazywania zmiennych i struktur danych
- jak stosować interpunkcję (np. spacje, średniki, kropki, nawiasy)

Semantyka określa znaczenie poprawnych składniowo wyrażń

Przykład problemów semantycznych - zmienne proceduralne:

jeśli składnia dopuszcza zmienne, których wartościami są nazwy procedur, to procedura $Proc(V)$ może być wywoływana z różnymi wartościami parametru np. $Proc(Rand)$ lub $Proc(Quick)$.

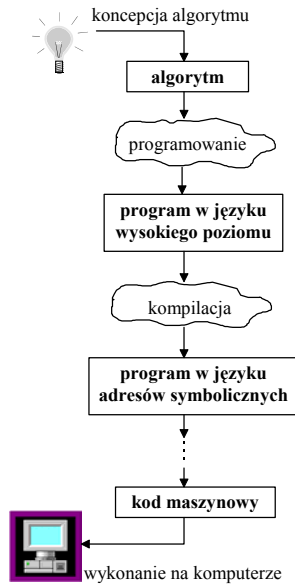
Jaki wynik uzyskamy gdy wywołamy $Proc(Proc)$ dla:



procedura $Proc(V)$:

1. wywołaj $V(V)$, umieszczając wynik działania w zmiennej X ;
2. jeśli $X = 1$, wróć i podaj wynik 0; w przeciwnym razie wróć i podaj wynik 1

Kompilatory i interpretatory

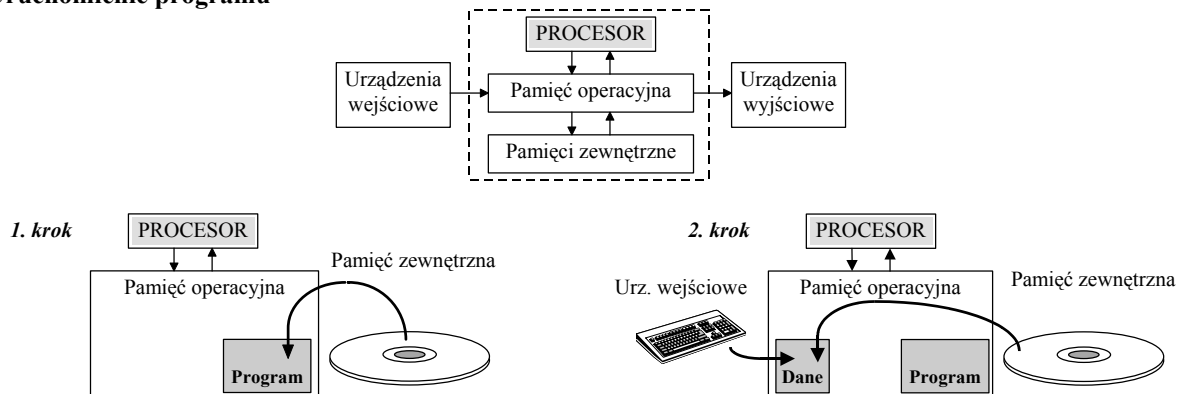


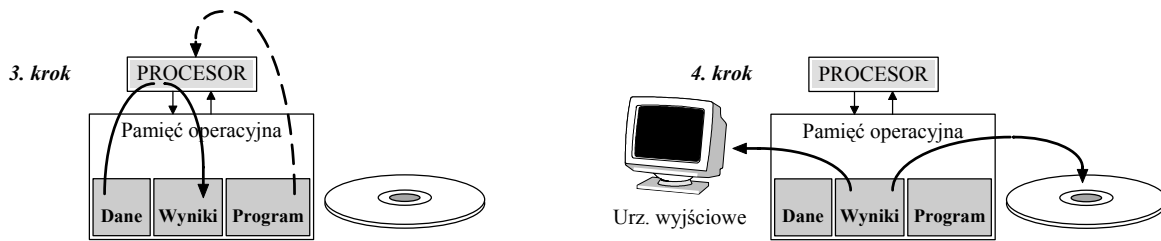
Język wysokiego poziomu	Asembler		
dla Y od 1 do N wykonaj (treść iteracji)	PĘTLA	LDS 0,Y	(załaduj 0 pod adres Y)
koniec		POR N,Y	(porównaj wart. pod adr.)
		SKR DALEJ	(jeśli równe, to skocz)
		DDS 1,Y	(dodaj 1 do wart. pod adr. Y)
		(tłumaczenie treści iteracji)	
		SKO PĘTLA	(skocz z powrotem)
	DALEJ	...	

Kompilacja - przekładanie całego programu napisanego w języku wysokiego poziomu na program w języku niższego poziomu

Interpretacja - przekładanie kolejno instrukcji języka wysokiego poziomu na instrukcje poziomu maszynowego

Uruchomienie programu





WIEŻA BABEL JĘZYKÓW PROGRAMOWANIA

BASIC

(Beginner's All-Purpose Symbolic Instruction Code)

- najprostszy język wyższego poziomu - łatwy do opanowania
- język ogólnego przeznaczenia
- opracowany w połowie lat 60 przez Johna Kemeny'ego i Thomasa Kurtza z Dartmouth College, N.H.
- stał się popularny na początku lat 80 wraz z rozwojem komputerów osobistych
- początkowo interpretowany, a obecnie raczej kompilowany

FORTRAN

(Formula Translation)

- język przeznaczony do obliczeń inżynierskich i naukowych
- opracowany w 1957 r. przez Johna Backusa z IBM
- wielokrotnie modyfikowany
- wciąż popularny w zastosowaniach numerycznych
- wersja opracowana w 1990 r. - FORTRAN 90 została wyposażona w wiele dodatkowych elementów rozszerzających zakres zastosowań
- sterowanie: skok **goto**, instrukcja warunkowa i ograniczona iteracja (brak rekurencji)

Przykład programu:

```

integer I, MX, MN, A(100)
real RS
read (A(I), I = 1, 100)
MX = A(1)
MN = A(1)
do 10 I = 2, 100
if (A(I).gt.MX) MX = A(I)
if (A(I).lt.MN) MN = A(I)
10 continue
RS = (MN + MX)/2
write RS
end

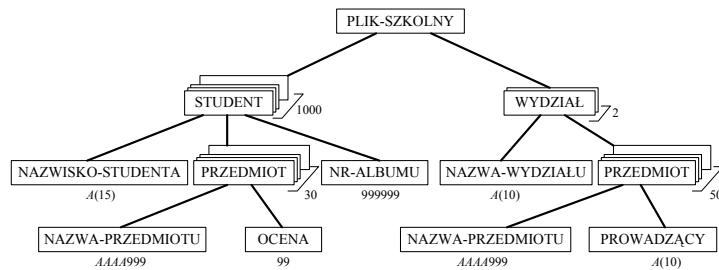
```

COBOL

(Common Business-Oriented Language)

- najpopularniejszy język w środowisku biznesu (banki, firmy ubezpieczeniowe, duże przedsiębiorstwa)
- firmy współpracujące z Departamentem Obrony USA powołały w 1959 r. CODASYL (Conference on Data Systems Languages), która opracowała język mający zapewnić łatwość przenoszenia programów na różne platformy sprzętowe i posiadający łatwe do zrozumienia instrukcje - podobne do zdań w języku angielskim
- od momentu powstania był wielokrotnie modyfikowany
- zawiera mechanizmy definiowania struktury pliku z danymi (sekcja **data division**)
- sterowanie: skok **goto**, instrukcja warunkowa i **perform** - ograniczona iteracja i wywołanie podprogramu

Przykładowa struktura pliku opisującego szkołę wyższą



Przykładowa definicja pliku opisującego szkołę wyższą

data division

```

01      PLIK-SZKOLNY
      02      STUDENT occurs 1000 times
      03          NAZWISKO-STUDENTA pic A(15)
      03          PRZEDMIOT occurs 30 times
      04              NAZWA-PRZEDMIOTU pic AAA4999
      04              OCENA pic 99
      03          NR-ALBUMU pic 999999
      02      WYDZIAŁ occurs 2 times
      03          NAZWA-WYDZIAŁU pic A(10)
      03          PRZEDMIOT occurs 50 times
      04              NAZWA-PRZEDMIOTU pic AAA4999
      04              PROWADZĄCY pic A(10)
    
```

PL/I

- złożony język zaproponowany przez SHARE - stowarzyszenie użytkowników komputerów IBM, w 1963 r.
- pierwszy podręcznik wydał IBM w 1965 r.
- ANSI (The American National Standards Institute) i inne organizacje kilkakrotnie modyfikowały język
- język przeznaczony zarówno do obliczeń naukowo-inżynierskich jak i do zastosowań w biznesie (łączy cechy FORTRANu, COBOLu i ALGOLu) - pozwala na przetwarzanie najrozmaitszych struktur danych i posiada wiele typów operacji arytmetycznych i innych

Pascal

- opracowany przez Niklausa Wirtha z Federal Institute of Technology w Zurichu w końcu lat 60.
- w zamierzeniu miał służyć celom edukacyjnym w systematycznej nauce programowania oraz pozwalać na budowę szybkich i niezawodnych kompilatorów
- pierwszy kompilator opracowany przez Wirtha w 1974 r. był za darmo rozpowszechniany w środowiskach uniwersyteckich
- wywarł duży wpływ na inne później powstające języki programowania, np. na język Ada
- reguły syntaktyczne języka są zwarte i czytelne co pozwala łatwiej go opanować w porównaniu z innymi językami wyższego poziomu
- można w nim przejrzysto opisywać nawet złożone algorytmy i struktury danych
- programy są łatwe do zrozumienia i łatwo wykrywa się w nich błędy
- oferuje możliwości definiowania nowych struktur danych (wskaźniki)

Przykładowe definicje nowych typów zmiennych

type kolor = (niebieski, czerwony, purpurowy, brązowy, biały);

type paleta = **set of** kolor;

type dzień = 1..365

Przykładowy program wypełniający listę

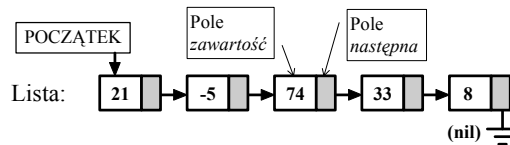
```

type komórka = record
    zawartość : integer;
    następna : dowiązanie
end;
type dowiązanie = ↑ komórka;
var POCZĄTEK, X : dowiązanie;
    
```

```

begin
new(POCZĄTEK);
X := POCZĄTEK;
while not eof do
  begin
    read(X↑.zawartość);
    if not eof then
      begin
        new(X↑.następna);
        X := X↑.następna
      end
    end;
    X↑.następna := nil
  end.

```



C

- został opracowany przez Dennisa Ritchie z AT&T Bell Laboratories w 1972 r.
- choć jest językiem wyższego poziomu to zawiera jednak wiele instrukcji charakterystycznych dla języków niższych poziomów (bezpośrednie operacje na adresach i bitach)
- pozwala tworzyć programy łatwe do przenoszenia na różne platformy sprzętowe
- system operacyjny UNIX został prawie całkowicie napisany w C
- staje się coraz bardziej popularny zarówno na mikrokomputerach jak i na większych maszynach

Snobol

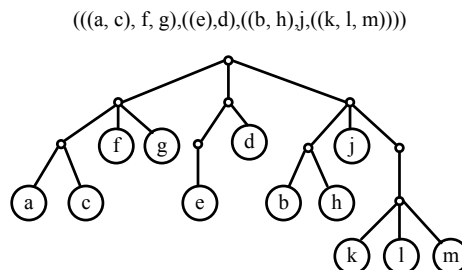
(String Oriented Symbolic Language)

- powstał we wczesnych latach 60 jako narzędzie do symbolicznego manipulowania tekstami
- eksploatuje pojęcie **wzorca** definiowanego przez programistę
- język dostarcza wiele operatorów działających na wzorcach i umożliwia budowanie wzorców złożonych
- sterowanie: głównie warunkowa instrukcja skoku
- typowa instrukcja składa się z tekstu (lub nazwy zmiennej tekstowej), poszukiwanego wzorca i etykiety wskazującej następne instrukcje do wykonania w razie sukcesu lub porażki dopasowywania wzorca

LISP (List Processor)

- został opracowany na przełomie lat 50 i 60 przez grupę z Massachusetts Institute of Technology kierowaną przez Johna McCarthy'ego (z wykorzystaniem formalizmu matematycznego zwanego rachunkiem **lambda**)
- opracowano kilka wersji języka (ta z 1984 r. jest de facto standardem)
- oparty jest na listach, jako podstawowych strukturach danych, i na rekurencji, jako podstawowej strukturze sterującej
- łatwości manipulowania listami obiektów o różnej naturze stanowi o jego unikalności
- wymaga dużych obszarów pamięci i jest zwykle interpretowany
- sprawdza się najlepiej w obliczeniach symbolicznych (rozgrywanie gier, przetwarzanie języka naturalnego, uczenie się, podejmowanie decyzji i wnioskowanie logiczne)

Przykład: lista lispowa i odpowiadające jej drzewo



Prolog

(Programming in Logic)

- zaprojektowany w 1970
- opiera się na **klauzulach** stwierdzających, że pewne **fakty** logicznie wynikają ze zbioru innych faktów (fakt oznacza zwykle, że pewne elementy pozostają w jakimś wzajemnym związku)
- rekurencja jest zasadniczą częścią języka
- program składa się ze zbioru „znanych” faktów, zbioru klauzul i dyrektywy, która poleca sprawdzenie prawdziwości pewnego faktu
- centralną strukturą danych są drzewa
- wykonanie programu polega na próbie dopasowania zmiennych i wyrażeń we fragmentach klauzul do ich odpowiedników w faktach - proces **uzgadniania**
- nadaje się podobnie jak Lisp do obliczeń symbolicznych i uznaje się go za odpowiedni do zastosowań z dziedziny sztucznej inteligencji

Przykłady stosowania klauzul:

klauzula **ZJADA(koty,myszy)** określa związek pomiędzy elementami

klauzula **ZJADA(X,Y) :- ZJADA(X,Z), ZJADA(Z,Y)** podaje związek logiczny pomiędzy trzema klauzulami

dyrektywa ? - **ZJADA(X,żaby)** może służyć do uzyskania odpowiedzi czy słonie jedzą żaby

APL

(A Programming Language)

- język pierwotnie opisany w książce Kennetha Iversona z IBM „A Programming Language” wydanej w 1962 r.
- w 1968 r. IBM przedstawiło pierwszą wersję kompilatora APL/360
- początkowo używany wyłącznie do obliczeń naukowo-inżynierskich, ale od czasu opracowania wersji APLSV (APL Shared Variable) w 1973 r. znalazł także zastosowanie w programach dla biznesu
- instrukcje mają wyjątkowo prostą notację a zestaw operacji jest bardzo rozległy
- operatory można stosować zarówno do zmiennych liczbowych jak i do wielowymiarowych tablic, które są podstawowymi strukturami danych

Przykłady działania operatorów

$$\begin{array}{|c|c|c|} \hline 5 & 7 & 1 \\ \hline 3 & 6 & 9 \\ \hline 2 & 4 & 1 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 1 & 1 & 2 \\ \hline 3 & 1 & 0 \\ \hline 2 & 2 & 4 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline 6 & 8 & 3 \\ \hline 6 & 7 & 9 \\ \hline 4 & 6 & 5 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline 5 & 7 & 1 \\ \hline 3 & 6 & 9 \\ \hline 2 & 4 & 1 \\ \hline \end{array} < \begin{array}{|c|c|c|} \hline 1 & 1 & 2 \\ \hline 3 & 1 & 0 \\ \hline 2 & 8 & 4 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$$

$$+/ \begin{array}{|c|c|c|} \hline 5 & 7 & 1 \\ \hline 3 & 6 & 9 \\ \hline 2 & 4 & 1 \\ \hline \end{array} \rightarrow \boxed{38} \quad \begin{array}{|c|c|} \hline 3 & 2 \\ \hline \end{array} \rho \begin{array}{|c|} \hline 6 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 & 6 \\ \hline \end{array}$$

Ada

- opracowanie języka zainicjował Departament Obrony USA w 1975 r. w celu uzyskania języka ogólnego przeznaczenia umożliwiającego łatwe przenoszenie programów
- pierwsza wersja powstała w 1979 r. i została nazwana imieniem księżnej Lovelace - asystenki Charlesa Babbage'a.
- język podobny do Pascala, ale wyposażony w wiele dodatkowych cech pozwalających pisać strukturalne programy o olbrzymich rozmiarach
- ze względu na swoje bogactwo nie jest zbyt wygodny dla przeciętnego użytkownika i z tego powodu nie stał się popularny poza zastosowaniami militarnymi w USA

Badania nad językami programowania

- ◇ podstawy matematyczne interpretacji i kompilacji
- ◇ definiowanie semantyki metodą operacyjną lub denotacyjną
- ◇ tworzenie języków zapytań i manipulowania danymi
- ◇ opracowywanie środowisk programowania
- ◇ wizualne języki programowania - „rysowanie” programów
- ◇ równoważność języków
- ◇ rozbudowa struktur sterujących
- ◇ zwiększanie elastyczności - możliwości definiowania własnych struktur