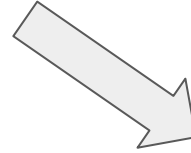
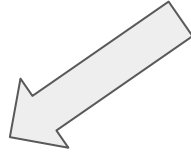


Podprogramy

Pascal, Ada

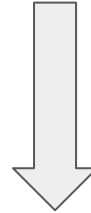


Funkcje



Wykonuje obliczenia,
zwraca wartość

Procedury



Nie zwraca wartości

Podprogram główny

zdefiniowany w kodzie źródłowym, wybrany podprogram, od którego rozpocznie się wykonanie gotowego programu.

```
main()  
{  
}
```

```
int main(int argc, char *argv[])  
{  
}
```

Procedury

PROCEDURE
(skrót - PROC)

Pascal, Ada, Algol,
Comala

SUBROUTINE (
skrót - SUB)

Basic, Visual
Basic

PERFORM

Cobol

Funkcje

PROCEDURE ... RETURNS

Function

Algol

Pascal, Ada,
Visual Basic

Przez nazwę - identyfikator przypisany do podprogramu w jego deklaracji jest to najczęściej spotykany przypadek w językach wysokiego poziomu np. Ada, Comala, Clipper, Pascal. Visual Basic

Przez etykietę - Basic

Przez liczbę - Basic, Visual Basic, Jean

Przez adres/referencję - Pascal

Rodziny podprogramów

możliwość definiowania wielu podprogramów o tej samej nazwie

78

```
DCL A GENERIC (PR1 WHEN(FLOAT),  
               PR2 WHEN(CHAR),  
               PR3 WHEN(FLOAT, CHAR),  
               PR4 WHEN(LABEL));
```

Metody wywołania podprogramu

Funkcyjne – w wyrażeniu, do którego podprogram zwraca obliczoną wartość,
np. poprzez nazwę z listą argumentów

Proceduralne - instrukcja wywołania np.
poprzez nazwę z listą argumentów,
po słowie kluczowym

Komunikacja podprogramu z otoczeniem

- zmienne globalne
- argumenty przypisywane i zdefiniowane w podprogramie parametry
- rezultaty funkcji
- pola obiektu
- wyjątki
- zmienne nakładane
- obszary wspólne

Podprogram w języku Java

```
class Nazwa {  
    ...  
    [public | protected | private] [native] [static] [synchronized] type name ( type1 arg1, type2  
    arg2 ){  
        ...  
    }  
    ...  
}
```

Podprogram w języku Pascal

```
procedure Procedura( {argumenty} );  
begin  
    { instrukcje do wykonania }  
end;
```

```
function Funkcja( {argumenty} ) : integer; { typ wartości funkcji }  
begin  
    { instrukcje do wykonania (ciało funkcji) }  
end;
```

Koniec