

Informatyka 1

Wykład IV

Wyrażenia i instrukcje cd., ręczna symulacja, operacje wejścia/wyjścia

Robert Muszyński
ZPCiR IIAiR PWr

Zagadnienia: instrukcja warunkowa CASE-OF-END, instrukcja pętli REPEAT-UNTIL, ręczna symulacja, operacje wejścia/wyjścia, wbudowane procedury i funkcje w Pascalu.

Copyright © 2001–2006 Robert Muszyński

Niniejszy dokument zawiera materiały do wykładu na temat podstaw programowania w językach wysokiego poziomu. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych, prywatnych potrzeb i może być kopiowany wyłącznie w całości, razem ze stroną tytułową.

Instrukcja warunkowa CASE-OF-END

```
instrukcja-CASE = "CASE" wyrażenie "OF"  
                element-listy-wyboru { ";"  
                element-listy-wyboru } [ ";" ] "END".  
element-listy-wyboru = lista-stalych ":" instrukcja.
```

Przykład: dodawanie lub mnożenie

```
CASE Oper OF  
  '+' : Z := X + Y;  
  '*' : Z := X * Y;  
  '-', '/' : WRITELN('Operacja niedozwolona: "',  
                    Oper, '".')  
END
```

Instrukcja pętli REPEAT-UNTIL

instrukcja-REPEAT = "REPEAT" ciąg-instrukcji
"UNTIL" wyrażenie-logiczne.
ciąg-instrukcji = instrukcja { ";" ciąg-instrukcji }.
wyrażenie-logiczne = wyrażenie.

Przykład: algorytm Euklidesa — wej.: n, m ; wyj.: n

```
REPEAT
  k := n MOD m;
  n := m;
  m := k;
UNTIL m = 0;
```

Ręczna symulacja

Przykład: zamiana wartości zmiennych A i B

1	A := 5;
2	B := 10;
3	A := B;
4	B := A;

nr linii	A	B	...	Uwagi
	?	?		
1	5			
2		10		
3	10			
4		10		

```

1  N := 27;
2  Pierwsza := TRUE;
3  Dzielnik := 2;
4  WHILE Pierwsza AND (Dzielnik <= N DIV 2) DO
5  IF (N MOD Dzielnik) = 0
6  THEN Pierwsza := FALSE
7  ELSE Dzielnik := Dzielnik + 1;
8  ...

```

nr linii	N	Pierwsza	Dzielnik	Uwagi
	?	?	?	
1	27			
2		TRUE		
3			2	
4				warunek WHILE: TRUE
5				warunek IF: FALSE
7			3	
4				warunek WHILE: TRUE
5				warunek IF: TRUE
6		FALSE		
4				warunek WHILE: FALSE
8				

Operacje wejścia/wyjścia

- instrukcje READ/READLN oraz WRITE/WRITELN operują na strumieniach danych;
- dwa standardowe strumienie danych są zawsze dostępne, jednak program musi zadeklarować ich użycie:

```
PROGRAM Testy(INPUT,OUTPUT);
```

- standardowe strumienie danych zwykle utożsamiane są z klawiaturą i ekranem terminala, jednakże system Unix pozwala przypisać dowolny z nich do plików dyskowych

```
%testy <testy.dane >testy.wyniki
```

- mechanizm skierowania strumieni można wykorzystać przy testowaniu programów.

Instrukcje READ/READLN

```
READ(v1, ..., vn)
```

```
READLN(v1, ..., vn)
```

v_i — zmienna

Przykłady:

```
READ(oper);
```

```
READLN(arg1, arg2);
```

```
READLN;
```

Instrukcje WRITE/WRITELN

```
WRITE(e1, ..., en)
```

```
WRITELN(e1, ..., en)
```

e_i — wyrażenie lub zmienna

Przykłady:

```
WRITELN('Oto jestem, Swiecie');  
{ mieszane dane napisowe i liczby INTEGER }  
WRITELN(' a= ', a:8, ' b= ', b:8);  
{ wynik typu BOOLEAN }  
WRITE('Wyznosc nieujemny? ', b*b-4*a*c>=0.0);  
{ liczby REAL }  
WRITE('Wyniki: x1=', x2:3, ', oraz x2=', x2:0:0);
```


Operacje wejścia/wyjścia na plikach

Struktura programu:

```
PROGRAM prog1(INPUT,OUTPUT,plik1,plik2);  
  
VAR plik1,plik2: TEXT;  
  
{... przykład wywołania:}  
READLN(plik1,Oper,Arg1,Arg2);  
{...}  
WRITE(plik2,'Wartosc koncowa: ',Arg1);
```

Operacje wejścia/wyjścia na plikach

Inicjalizacja plików:

```
RESET(plik1, 'arytmetyka.d');  
REWRITE(plik2, 'wyniki.txt');  
APPEND(plik2, 'wyniki.txt');  
CLOSE(plik1);
```

Przydatne funkcje:

```
EOLN(tekst1);  
EOF(tekst1);
```

Wbudowane procedury i funkcje w Pascalu

Różnice między procedurami a funkcjami:

- funkcja wylicza i zwraca wartość, a procedura nie;
- procedura stanowi instrukcję i może wystąpić w programie jedynie w sekwencji instrukcji;
- funkcja stanowi wyrażenie i może wystąpić w programie jedynie w roli wyrażenia.

Wbudowane procedury i funkcje w Pascalu

Przykłady procedur:

- READ/READLN,
- WRITE/WRITELN,
- RESET/REWRITE, CLOSE,
- HALT;

Przykłady funkcji:

- SQRT, SQR,
- ABS,
- SIN, COS,
- EXP, LN,
- ROUND, TRUNC,
- ODD,
- EOLN, EOF

oraz

- $SUCC(w)$ — wylicza następną po w wartość w typie porządkowym,
- $PRED(w)$ — wylicza poprzednią przed w wartość w typie porządkowym,
- $ORD(w)$ — wylicza numer wartości w w typie porządkowym,
- $CHR(n)$ — wylicza znak o numerze porządkowym n .

$$\begin{array}{l|l} SUCC(PRED(w)) = w & CHR(ORD(w)) = w \\ PRED(SUCC(w)) = w & ORD(CHR(n)) = n \end{array}$$