

## Wykład PASCAL

### Zmienne wskaźnikowe i dynamiczne



Zmienne dzielą się na *statyczne* i *dynamiczne*.

- ✓ *Zmienna statyczna* tworzona jest w trakcie kompilacji programu i istnieje do momentu jego zakończenia lub też do momentu zakończenia funkcji lub procedury, w której została zadeklarowana.
- ✓ *Zmienną dynamiczną* nazywamy zmienną utworzoną w trakcie wykonywania programu. Zmienne te tworzone są na tzw. *stercie*, której rozmiar jest na ogół różny.

W odróżnieniu od statycznej, zmienną dynamiczną można w dowolnym momencie usunąć, zwalniając w ten sposób niepotrzebnie zajmowaną pamięć.



Do pracy ze zmiennymi dynamicznymi używa się **typów wskaźnikowych**.

Elementami tych typów są **zmienne wskaźnikowe**, nazywane **wskaźnikami**.

Wskaźnik jest adresem pewnej zmiennej określonego typu. Mówiąc o wskaźniku zawsze mamy na myśli zmienną, której adres zawiera wskaźnik.

Ze względu na różny typ zmiennych mówimy o wskaźniku na liczbę typu *Byte*, wskaźniku na liczbę rzeczywistą *Real*, wskaźniku na łańcuch typu *String* itd - zawsze ze wskaźnikiem związany jest typ zmiennej na którą wskazuje dany wskaźnik.



## Zmienne wskaźnikowe



Każdy typ danych w Pascalu ma odpowiadający mu typ wskaźnikowy. Jest to typ, który nazywa się tak samo jak typ, któremu odpowiada, tylko poprzedzony jest znakiem **^**.

Na przykład:  
dla typu **Integer** jest to **^Integer**.

Zmienna takiego typu może przechowywać adres pewnych danych w pamięci – „wskazuje na coś”

Definiowanie i deklarowanie typów wskaźnikowych:

```

1  TYPE
2      Pr = ^real;
3      Pint = ^integer;
4      Prec =
5  VAR
6      x : Pr;
7      r : Prec;
8      i, j, k : Pint;

```

To samo w wersji skróconej:

```

10  VAR
11      x: ^real;
12      r: ^RECORD
13          ...
14          END;
15      i, h, k: ^integer;

```



## Operacje na zmiennych wskaźnikowych:

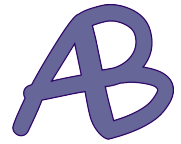
- przypisanie wartości jednego wskaźnika innemu wskaźnikowi,

$$\mathbf{x := y}$$

- relacja, tj. badanie czy wskaźniki mają identyczne wartości, co oznacza, że wskazują na ten sam obiekt w pamięci,

$$\mathbf{x = y}$$
$$\mathbf{x \langle \rangle y}$$

Uwaga: w obu powyższych przykładach  $x$  i  $y$  muszą być zmiennymi wskaźnikowymi tego samego typu.



## Operacje na zmiennych wskaźnikowych:

- przypisanie wartości jednego wskaźnika innemu wskaźnikowi,

$$\mathbf{x := y}$$

- relacja, tj. badanie czy wskaźniki mają identyczne wartości, co oznacza, że wskazują na ten sam obiekt w pamięci,

$$\mathbf{x = y}$$
$$\mathbf{x \langle \rangle y}$$

Uwaga: w obu powyższych przykładach  $x$  i  $y$  muszą być zmiennymi wskaźnikowymi tego samego typu.

## Adresy zmiennych

Adresy zmiennych pobrać możemy za pomocą operatora @

Aby poznać adres zmiennej „zmienna”, wystarczy, że napiszemy „@zmienna”.

```
1  var
2      zmienna: Integer;
3      adres_zmiennej: ^Integer;
4  begin
5      zmienna := 15;
6      adres_zmiennej := @zmienna; {pobiera adres zmiennej}
7      WriteLn(adres_zmiennej^); {wypisuje zawartość
8                               zmiennej ze wskazanego
9                               adresu}
10 end.
```

**Odwołanie do wartości obiektu wskazywanego odbywa się poprzez operator <sup>^</sup>**

Umieszczamy <sup>^</sup> za nazwą zmiennej wskaźnikowej.



# Adresy zmiennych



W Pascalu znak **^** ma dwa znaczenia:

- I. Może oznaczać typ wskaźnikowy, wtedy znajduje się przed nazwą typu. To zastosowanie zazwyczaj ma miejsce tylko w sekcji var.
- I. Może oznaczać operację pobrania wartości zmiennej z pamięci, wtedy znajduje się za nazwą zmiennej wskaźnikowej.

Należy uważać, żeby nie pomylić się, w którym miejscu umieścić znak **^**

```
1  var
2  adres_zmiennej: ^Integer;
3  begin
4
5      adres_zmiennej^ := 10;
6      WriteLn(adres_zmiennej^);
7      .....
8  end.
```



Przypisanie wartości zmiennej dynamicznej do zmiennej statycznej

$\mathbf{k := x^{\wedge};}$

Przypisanie wartości zmiennej dynamicznej do zmiennej dynamicznej

$\mathbf{y^{\wedge} := x^{\wedge};}$

Porównanie zmiennej dynamicznej i zmiennej statycznej

$\mathbf{x^{\wedge} < k ;}$

## Adresy zmiennych

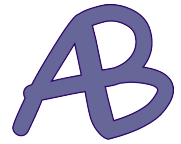
---



Niezależnie od typu wskaźnika zawsze można mu przypisać tzw. *adres pusty*, który w języku Pascal oznaczamy słowem *Nil* - wskaźnik o wartości *Nil* nie wskazuje niczego.



## Zmienne dynamiczne



## Tworzenie i usuwanie zmiennych dynamicznych

---

Tworzenie i usuwanie zmiennych odbywa się wewnątrz programu za pomocą procedur **New( )** oraz **Dispose( )**.

Procedury te wymagają jako argumentu zmiennej wskaźnikowej.

**NEW(p)** – tworzenie zmiennej dynamicznej typu zgodnego ze zmienną wskaźnikową **p**.

Odbywa się w dwóch etapach:

1. utworzona zostaje zmienna dynamiczna wartości nieokreślonej (rezerwacja pamięci);
2. zmiennej **p** zostaje przypisana nowa wartość będąca wskazaniem (adresem) nowo utworzonej zmiennej.

Do wartości tak utworzonej zmiennej dynamicznej można się odwoływać poprzez nazwę: **p^**.



## Tworzenie i usuwanie zmiennych dynamicznych

---

**DISPOSE(p)** – usuwanie zmiennej dynamicznej wskazywanej przez wskaźnik **p**

1. zmiennej **p** zostaje przypisana wartość nieokreślona, tj. zmienna **p** nie wskazuje już na żaden określony obiekt w pamięci;
2. zwolniona zostaje pamięć zajmowana przez zmienną dynamiczną i może być od tej pory wykorzystana do innych celów.

## Zmienne dynamiczne



```
1  type
2  tablica_real = array[1..1024 * 1024] of Real;
3  tablica_longint = array[1..1024 * 1024] of Longint;
4
5  var tr: ^tablica_real;
6      tl: ^tablica_longint;
7      typ, ile, i: Longint;
8      suml: Longint;
9      sumd: Real;
10
11 begin
12   Write('Jaki typ zmiennych wczytac? (1 - Real, 2 - Longint): ');
13   Readln(typ);
14   Write('Ile zmiennych wczytac? (1 - 1 000 000): ');
15   Readln(ile);
16   if typ = 1 then begin
17     New(tr);
18     for i := 1 to ile do
19       Read(tr^[i]);
20     for i := 1 to ile do
21       sumd := sumd + tr^[i];
22     Writeln('Suma: ', sumd);
23   end
24   else
25   begin
26     New(tl);
27     for i := 1 to ile do
28       Read(tl^[i]);
29     for i := 1 to ile do
30       suml := suml + tl^[i];
31     Writeln('Suma: ', suml);
32   end;
33 end.
```

Źródło:  
[http://main.edu.pl/pl/user.phtml?op=lesson  
&n=18&page=pascal](http://main.edu.pl/pl/user.phtml?op=lesson&n=18&page=pascal)

## Zmienne dynamiczne



```
1  type
2  tablica_real = array[1..1024 * 1024] of Real;
3  tablica_longint = array[1..1024 * 1024] of Longint;
4
5  var tr: ^tablica_real;
6  tl: ^tablica_longint;
7  ile, i: Longint;
8  suml: Longint;
9  sumd: Real;
10
11 begin
12   Write('Ile zmiennych wczytac? (1 - 1 000 000): ');
13   Readln(ile);
14   Writeln('Podaj zmienne typu Real:');
15   New(tr);
16   for i := 1 to ile do
17     Read(tr^[i]);
18   for i := 1 to ile do
19     sumd := sumd + tr^[i];
20   Writeln('Suma: ', sumd);
21   Dispose(tr);
22   Writeln('Podaj zmienne typu Longint:');
23   New(tl);
24   for i := 1 to ile do
25     Read(tl^[i]);
26   for i := 1 to ile do
27     suml := suml + tl^[i];
28   Writeln('Suma: ', suml);
29   Dispose(tl);
30 end.
```

Źródło:  
[http://main.edu.pl/pl/user.phtml?op=lesson  
&n=18&page=pascal](http://main.edu.pl/pl/user.phtml?op=lesson&n=18&page=pascal)



## Literatura:

---

### W prezentacji wykorzystano przykłady i fragmenty:

- R. Jarża, *Turbo Pascal. Szkoła programowania*, Wydawnictwo Robomatic 2000.  
(dostępne w bibliotece uczelni)
- <http://pascal.kurs-programowania.pl>
- <http://4programmers.net>
- T. M. Sadowski, *Turbo Pascal. Programowanie*, Helion 1996.

