

TYPY W PASCAL

Typ całkowity:

Nazwa typu	Zakres	Format
Shortint	-128..127	znak, 8-bitowy (1 bajt)
Longint	-2147483648..2147483647	znak, 32-bitowy (4 bajty)
<u>Integer</u>	-32768..32767	znak, 16-bitowy (2 bajty)
Longword	-2147483648..2147483647	bez znaku, 32-bitowy (4 bajty)
Byte	0..255	bez znaku, 8-bitowy (1 bajt)
Word	0..65535	bez znaku, 16-bitowy (2 bajty)

Przykład:

```
var liczba_dni, wiek : integer;
begin
    writeln('podaj swój wiek');
    readln(wiek);
    writeln('ile dni zostało się do twoich imienin')
    readln(liczba_dni);
    writeln('Masz ',wiek,' lat i do twoich imienin zostało ',liczba_dni,' dni');
end.
```

Typ rzeczywisty:

Nazwa typu	Zakres	Format
<u>Real</u>	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	znak, 48-bitowy (6 bajtów)
Single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	znak, 32-bitowy (4 bajty)
Double	$1.5 \times 10^{-309} \dots 1.7 \times 10^{309}$	znak, 64-bitowy (8 bajtów)
Extended	$3.6 \times 10^{-4951} \dots 1.1 \times 10^{4932}$	znak, 80-bitowy (10 bajtów)
Comp	$-2^{63} \dots 2^{63} - 1$	znak, 64-bitowy (8 bajtów)

Przykład:

```
var wzrost: real;
begin
    writeln('podaj swój wzrost w metrach');
    readln(wzrost);
    writeln('Masz ',wzrost,' metra wzrostu');
end.
```

Typ znakowy:

Typ znakowy służy do przechowywania pojedynczych znaków. Wartości zmiennym typu Char przyporządkujemy za pomocą instrukcji przypisania. Deklaracja zmiennej znak typu znakowego wraz z przypisaniem jej znaku B ma postać:

```
var znak :char;           //deklaracja zmiennej 'znak'
znak:='B';                //przypisanie zmiennej znak wartości B
```

Typ łańcuchowy:

W języku Pascal ciągi znaków zapisuje się przy użyciu typu łańcuchowego String. Potrafi on przechować ciągi znaków o dynamicznie zmieniającej się długości, którą określamy jako atrybut typu podawany w nawiasie kwadratowym.

Gdy nie podamy wielkości w nawiasie kwadratowym, zmienna typu string przyjmie maksymalną długość 255 znaków.

```
var imie :string[20]; //imie zmienna typu znakowego 'String' mogąca przechować 20 znaków
```

PRZYKŁAD ZASTOSOWANIA:

```
var imie :string[20];
begin
    writeln('jak masz na imie ? ');
    readln(imie);
    writeln('witaj' ,imie);
    readln;
end.
```

WAŻNE:

Poniższe zapisy są równoważne – maksymalna długość tekstu jaki można zapisać w zmiennych zadeklarowanych poniżej to 255 znaków:

Var text1:string;	\	Obie zmienne będą w stanie przechować
Var text2:string[255];	/	maksymalnie 255 znaków.

Typ logiczny:

Typ **Boolean** jest typem, którego zmienne przechowują tylko dwie wartości: TRUE – prawda albo FALSE – fałsz. Typ ten dopuszcza stosowanie operatorów logicznych.

Przykład zastosowania typu logicznego:

```
var rowne:Boolean;
begin
    rowne := 2=2;           //zmiennej rowne zostanie przypisana wartość TRUE lub
                           //FALSE – w zależności czy warunek (w tym wypadku
                           //sprawdzamy czy 2 rowne jest 2) jest spełniony;
    writeln(rowne);         //poniewaz porównaliśmy 2 z 2 więc zwróci TRUE
    rowne := 2=5;           //wynik porównania 2 i 5 jest równy FALSE poniewaz 2
                           //nie jest rowne 5
    writeln(rowne);         //Zwróci FALSE bo 2 nie jest rowne 5
    rowne := 2<5;           //Poniewaz 2 jest mniejsze od 5 do zmiennej przypisane
                           //zostanie TRUE.
    writeln(rowne);
end.
```

Typ okrojony:

Jest to typ porządkowy, definiowany przez użytkownika. Umożliwia on eliminowanie danych znajdujących się poza zakresem. Przykładem użycia typu okrojonego może być zmienna tydzień, której zadaniem jest przechowanie numerów dni tygodnia. Przykładowo, zamiast używać typu Byte, można zastosować typ okrojony:

```
var
    tydzień :1..7;
```

Taki zapis sprawi, że zmienna tydzień przechowa wartości od 1 do 7 i próba przypisania jej jakiegokolwiek innej wartości skończy się wyświetleniem komunikatu o błędzie. W zmiennych typu okrojonego wyróżniamy typ bazowy. Dla typu składającego się z wartości znakowych, np. typu Char, typ okrojony zadeklarujemy następująco:

```
var
```

alfabet : 'a'..'z';

Dla takiej deklaracji wartościami dopuszczalnymi zmiennej alfabet są małe litery alfabetu łacińskiego, natomiast typ Char jest typem bazowym.

Typ wyliczeniowy:

Podczas tworzenia programu możemy posługiwać się tzw. typem okrojonym. Jego implementacja i działanie są podobne do typu wyliczeniowego z tą różnicą, że elementami typu wyliczeniowego są nazwy zdefiniowane przez programistę.

Typ wyliczeniowy jest więc ograniczony do zbioru wcześniej wprowadzonych wartości. Każdy element zbioru typu wyliczeniowego ma swój własny numer porządkowy, który dla pierwszego elementu wynosi 0. Jeśli deklarowalibyśmy typ wyliczeniowy, który składałby się z pór roku, to posłużylibyśmy się zapisem:

```
type pory_roku=(wiosna, lato, jesien, zima) ;
```

wiosna otrzyma w powyższej definicji wartość porządkową 0, a zima 3.

Procedur writeln, write, read i readln nie można niestety stosować w odniesieniu do zmiennych tych typów, ponieważ nie potrafią wypisać ani pobrać zawartości zmiennej typu wyliczeniowego.

Typ wyliczeniowy definiujemy w części programu lub podprogramu rozpoczynającej się słowem kluczowym type:

```
type nazwa_typu_wyliczeniowego=(element_0,element_1,element_2,...)
```

Przykładem użycia typu wyliczeniowego jest program z listingu:

```
program typ_wyliczeniowy;
type
  pory_roku=(wiosna, lato, jesien, zima);
var
  pora: pory_roku; {definicja zmiennej pora określa, że jest ona
                    typu pory_roku}
begin
  writeln ('Namaluje tyle gwiazdek, ile elementow ma typ wyliczeniowy
          pory_roku') ;

  for pora :=wiosna to zima do
    writeln ('*') ;
  readln ;
end.
```

Jak wynika z powyższego przykładu, elementy typu pory roku nie zawierają polskich znaków (jesien).

Zadeklarowaliśmy też zmienną pora, która jest typu pory_roku, a zatem może przyjmować wartości: wiosna, lato, jesien, zima. Nasz program wykona pętlę, przyjmując dla zmiennej pora kolejno wartości wiosna, lato, jesien, zima, za każdym razem wyświetlając na konsoli gwiazdkę. Gdy zmienimy linie programu i zamiast for pora := wiosna to zima do wpiszemy for pora := lato to zima do, wówczas nasza pętla zacznie wykonywać wypisanie gwiazdki od lata (wartość 1) do zima (wartość 4) i zamiast czterech gwiazdek zostaną wyświetlone tylko trzy.