

Pascal – typy danych

Zmienna i typ

- Zmienna to obiekt, który może przybierać różne wartości.
- Typ zmiennej to zakres wartości, które może przybierać zmienna.
- Deklarujemy je w nagłówku poprzedzając słowem kluczowym **var** (od *variable* – zmienna).
- Przykładowa deklaracja.

`var x : integer;`

deklaracja nazwa zmiennej : typ zmiennej;
2

Podział typów danych:

- Według miejsca:
 - **wewnętrzne** - przechowywane w pamięci operacyjnej w czasie działania programu,
 - **zewnętrzne** - przechowywane w pamięci masowej w czasie działania programu.
- Według rezerwacji pamięci:
 - **statyczne** - pamięć przydzielana jest przed rozpoczęciem działania instrukcji bloku
 - **dynamiczne** - pamięć przydzielana jest w czasie działania określonych instrukcji.

Typy pascalowe

- **prosty**
 - Porządkowy
 - Wyliczeniowy
 - Logiczny
 - Znakowy
 - Całkowity
 - Rzeczywisty
- **napisowy**
- **Strukturalny**
 - Tablicowy
 - Rekordowy
 - Zbiorowy
 - plikowy (zewnętrzny)
- **wskaźnikowy (dynamiczny)**
- **Obiektowy**

Reprezentacja typów danych

- Zmienne w komputerze są reprezentowane za pomocą bitów.
- W zależności od ilości bitów słowo może mieć różną długość.
- Oznacza to, że zmienna może mieć różny zakres, jak zajmować niejednakową ilość miejsca w pamięci.
 - Zależność jest prosta: im więcej ma bitów, tym większy zakres, ale zmienna zajmuje więcej miejsca w pamięci komputera.
 - Dlatego wyróżniamy kilka typów każdego rodzaju liczb i innych zmiennych.

5

Typy całkowite

- Typ całkowity to liczba całkowita.
 - Typ całkowity jest predefiniowany słowem `Integer`.
- **Liczby bez znaku**
- Ten typ danych odpowiada kodowaniu wartości całkowitych w naturalnym kodzie dwójkowym.
 - **byte**
 - 1-bajtowa liczba całkowita bez znaku (8 bitów).
 - Zakres od 0 do $2^8 - 1$ (0 - 255).
 - **word**
 - 2-bajtowa liczba całkowita bez znaku (16 bitów).
 - Zakres od 0 do $2^{16} - 1$ (0 - 65535).

6

Typy całkowite

- **Liczby ze znakiem**
- Ten typ danych odpowiada kodowaniu wartości całkowitych w kodzie uzupełnieniowym do 2 - U2.
 - **shortint**
 - 1-bajtowa liczba całkowita ze znakiem w kodzie U2 (8 bitów).
 - Zakres od -2^7 (-128) do $2^7 - 1$ (127).
 - **integer**
 - 2-bajtowa liczba całkowita ze znakiem w kodzie U2 (32 bity).
 - Zakres od -2^{15} (-32768) do $2^{15} - 1$ (32767).
 - **longint**
 - 4-bajtowa liczba całkowita ze znakiem w kodzie U2 (32 bity).
 - Zakres od -2^{31} (-2147483648) do $2^{31} - 1$ (2147483647).

7

Typy całkowite – zestawienie (dla wszystkich języków Pascal)

bez znaku - NBC	Ilość bitów	Zakres	ze znakiem - U2		Zakres
byte	8b - 1B	$0 \dots 2^8 - 1$	shortint	8b - 1B	$-2^7 \dots 2^7 - 1$
word	16b - 2B	$0 \dots 2^{16} - 1$	smallint	16b - 2B	$-2^{15} \dots 2^{15} - 1$
longword	32b - 4B	$0 \dots 2^{32} - 1$	longint	32b - 4B	$-2^{31} \dots 2^{31} - 1$
			int64	64b - 8B	$-2^{63} \dots -1$
cardinal	32b - 4B	$0 \dots 2^{32} - 1$	integer	32b - 4B	$-2^{31} \dots 2^{31} - 1$

8

Działania na typach całkowitych

:= + - * / DIV MOD
= <> < <= > >=
NOT AND OR XOR (t) SHL (t) SHR
(t)
Abs Chr Odd Ord Pred Sqr Succ
Dec (t) Inc (t) Str (t) Val (t)

Stałe całkowite: 0 +1 -25 59 -116

9

Typ okrojony typów całkowitych

type
cyfra = 0 .. 9;
indeks = 1 .. 100;
bajt = 0 .. 255;
bit = 0 .. 1;
zakres = -7 .. 13;
jeden = 1 .. 1;
end;

10

Typy rzeczywiste

- Typy rzeczywiste to liczby z przecinkiem. Zawierają część całkowitą i ułamkową.
- Typ rzeczywisty jest predefiniowany słowem Real.
- W odróżnieniu od rzeczywistości typ REAL to skończony podzbiór liczb wymiernych.

11

Nazwa	Bajty	Dokładność	Maximum
Real	6	12 cyfr	1.7E38
Single	4	8 cyfr	3.4E38
Double	8	16 cyfr	1.7E38
Extended	10	19 cyfr	1.1E4932
Comp	8	19 cyfr	9.2E18

12

Działania na liczbach rzeczywistych:

:= + - * / = <> < <= > >=

Abs Arctan Cos Exp Ln Round Sin

Sqr Sqrt Trunc Frac (t) Str (t) Val (t)

- NIE MA : Pred Succ
- NIE MA typu okrojonego
- Stałe rzeczywiste: 0.0 -3.1416 +0.01 1E6
123.456

13

Typ logiczny

- **Boolean**

- Przyjmuje dwie wartości logiczne: **TRUE** (prawda, tak) i **FALSE** (fałsz, Nie).

- Zmienna 8-bitowa (1 bajt).

- Jeśli wszystkie bity są ustawione na 0, to przedstawia wartość logiczną FALSE.

- Jeśli chociaż jeden z bitów ma wartość 1, to przedstawia wartość logiczną TRUE.

- Standardowo wyrażenia logiczne zmieniają zawartość najmłodszego bitu - pozostałe przyjmują zawsze wartość 0:

00000000 (BOOLEAN) = FALSE (fałsz)

00000001 (BOOLEAN) = TRUE (prawda)

14

Typy znakowe

- Znaki przechowywane są w pamięci komputera w postaci kodów ASCII (8 bitów) lub Unicode (16 bitów).
- Kod znaku jest liczbą całkowitą w naturalnym kodzie binarnym.
- **char**
 - Dana 1-znakowa. Zmienna tego typu przechowuje jeden znak ASCII (8-bitów, 1 bajt). Kod znaku zawiera się w granicach od 0 do 255.
 - Znaki o kodach mniejszych od 32 są zwykle tzw. znakami sterującymi, których celem nie jest prezentacja znaków, lecz wykonanie określonych działań.
 - Na przykład znak NL o kodzie 10 powoduje przejście z wydrukiem do nowego wiersza.
 - Znak CR o kodzie 13 ustawia kursor na początku wiersza.

15

Typy znakowe

- **string**

- Dana typu string jest 256 elementową tablicą znakową i może przechowywać ciąg znaków.]

- Dostęp do poszczególnych literek przechowywanego tekstu uzyskuje się za pomocą indeksu.

- Element o indeksie 0 zawiera informację o ilości przechowywanych znaków.

- Kolejne elementy zawierają poszczególne znaki tekstu.

- Ilość znaków jest zatem ograniczona do 255.

16

Typy znakowe

- W zmiennej *s* umieszczamy napis *Janusz*.
- Zawartość zmiennej typu **string**

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]
\0	J	a	n	u	s	z	?

- Zawartość komórek zmiennej *s* poza obszarem tekstu (*s*[7]...*s*[255]) jest niezdefiniowana, tzn. mogą tam być dowolne znaki pochodzące z poprzednich działań nad tą zmienną lub obszarem pamięci przez nią zajmowanym.

17

Typy wyliczeniowe

- Typ wyliczeniowy to rodzaj typu danych zawierający listę wartości, jakie może przyjmować zmienna tego typu.
- Przykład

Type

```
pora_roku = (wiosna, lato, jesien, zima);  
miesiac = (styczen, luty, marzec, kwiecień);
```

18

Typy okrojone

- Typ okrojony określa podzbiór dowolnego typu porządkowego lub wyliczeniowego.
- Jest on ograniczony pewnym zakresem. Wykorzystywany jest do określania typu zmiennych, które mogą przyjąć wartości ze ściśle określonego przedziału, np.

```
TYPE TRok=1990..1999;
```

```
VAR a : TRok;
```

- Zmienna "a" może przyjąć wartości od 1990 do 1999.

Type

```
Dni_roboty=(poniedzialek,wtorek,środa,czwartek,piatek);
```

19

Typy całkowite

- Typ zbiorowy określa zbiór potęgowy, którego elementy są typu porządkowego.
- Przez zbiór potęgowy należy rozumieć zbiór wszystkich możliwych podzbiorów wartości typu porządkowego, włączając w to zbiór pusty.
- Definicja tego typu jest następująca:
 - Typ_zbiorowy = SET OF Typ;
- Przykład:

Type

```
Litery = SET OF 'a'..'z';
```

```
Cyfry = SET OF '0'..'9';
```

- Teraz zmienne typu *Cyfry* mogą przyjmować tylko wartości od '0' do '9'.

20

Typy tablicowe

- Typ tablicowy określa tablicę składającą się ze stałej liczby elementów.
 - Wszystkie elementy tablicy też muszą być określonego typu.
 - Do każdego elementu odwołuje się przez indeks podawany w nawiasach kwadratowych i określający pozycję elementu.
- Każda tablica może być wielowymiarowa.
- Przykładowa deklaracja tablicy 50 znaków może wyglądać następująco:

Type

```
wektor = array[0...50] of Integer;
```

21

Typy plikowe

- Typy plikowe są powiązane z plikami.
- Plik jest ciągiem elementów tego samego typu, tyle że liczba jego składowych jest zmienna.
 - Jest ona uzależniona od przebiegu wykonywania programu, a w szczególności od skojarzenia pliku z fizycznym zbiorem danych.
 - Od tablicy plik różni się ponadto metodą dostępu do poszczególnych elementów.
- Definicja pojedynczego typu plikowego ma postać:

TYPE

```
Identyfikator_typu=file of opis_typu_elementów_pliku;
```

- lub

```
TYPE Identyfikator_typu = file;
```

- Przykłady:

```
TYPE Dane = file of Integer;
```

```
Zbior = file;
```

```
Wyniki = Text;
```

22

Typy rekordowe

- Typ rekordowy opisuje złożoną strukturę danych.
 - Taka struktura składa się z pewnej liczby pól. Każde z nich jest ściśle określonego typu. Definicję rekordu zamieszcza się między słowami RECORD i END;
- Przykładowo, by zapamiętać dane o pracownikach firmy, używając definicji tablicy typu X, gdzie X jest typem rekordowym:

```
TYPE Osoba = RECORD
```

```
  Imie : String[10];
```

```
  Nazwisko : String [20];
```

```
  Wiek : Byte;
```

```
END;
```

```
VAR Ludzie : Array[1..1000] of Osoba;
```

- Do odczytania informacji zapisanych w polu rekordu używa się kropki np.:

```
Writeln(Ludzie[10].imie);
```

```
Ludzie[11].nazwisko:='Nowak';
```

23