

DATOVÉ TYPY

Proměnné

Konstanty

Proměnné

- Paměťové místo, které slouží k uložení nějakých údajů (vstupních, výstupních dat, mezivýpočtů, ...)
- **Typy proměnných** - deklarace = určení druhu hodnot, možnosti jejich použití a možné operace s nimi
 - + číselné typy sčítá, řetězce spojuje
- **Přiřazení hodnoty** proměnným v programu
 - A: = 10; C: = 'Chyba! '
 - D: = true B: = 'N'
- **Hodnoty zadá uživatel a jsou do proměnné načteny**
 - příkaz READ, READLN...

Konstanty

- Datový objekt, jehož hodnota se nemění
- Užití :např. v programu se vyskytne třeba 10 cyklů, z nichž každý se bude provádět 20x
 - Definice konstanty: (neuvádí se typ, je určen zadanou hodnotou)

const hmez = 20;

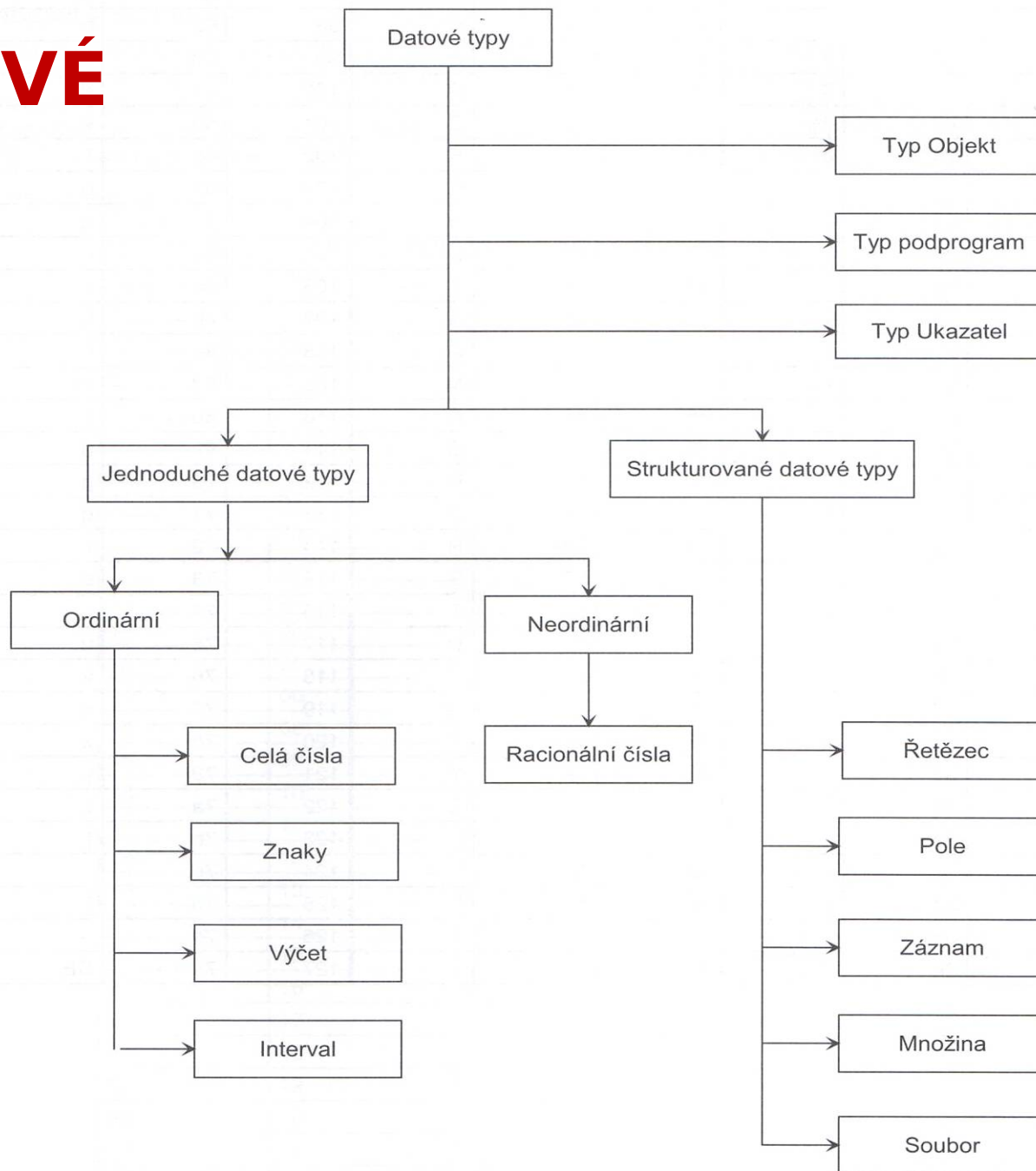
další příklady: min = 0;

max = 100;

zprava = 'Špatně zadaná hodnota'

Názvy proměnných a konstant: znaky malých nebo velkých písmen bez diakritiky a čísla

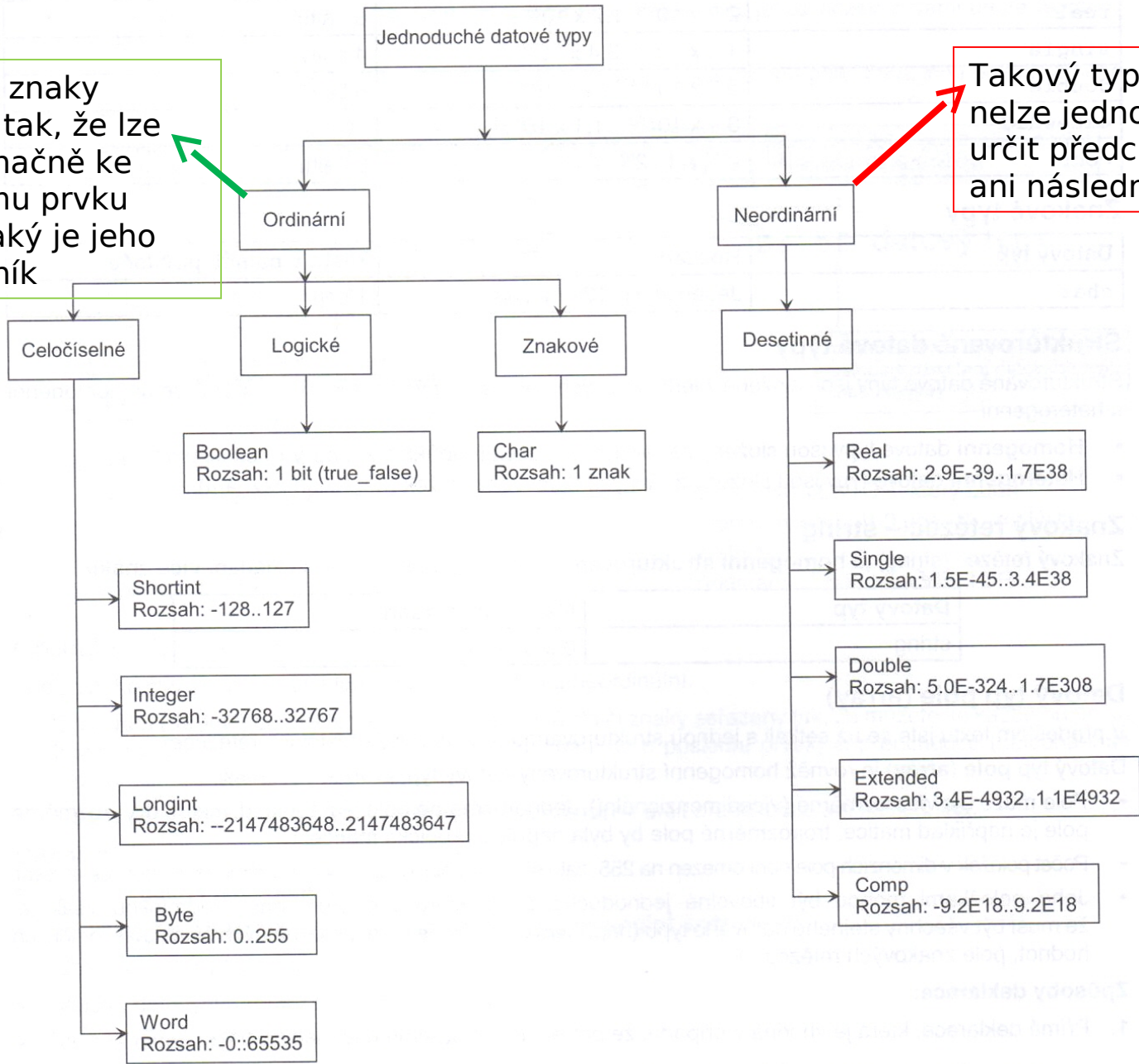
DATOVÉ TYPY



Jednoduché datové typy a jejich rozsahy

Čísla či znaky řazeny tak, že lze jednoznačně ke každému prvku určit, jaký je jeho následník

Takový typ, kde nelze jednoznačně určit předchůdce ani následníka



Datové typy - jednoduché

boolean

Rozsah	Místo v paměti počítače
True (pravda), False (nepravda)	1 bit

shortint
integer
longint
byte
word

Rozsah	Místo v paměti počítače
-128 .. 127	8 bitů (1 bajt)
-32768..32767	16 bitů (1 slovo = 2 bajty)
-2147483648..2147483647	32 bitů (2 slova = 4 bajty)
0..255	8 bitů (1 bajt)
0..65535	16 bitů (1 slovo = 2 bajty)

real
single
double
extended
comp

Rozsah	Místo v paměti počítače
$2,9 \times 10^{-39} .. 1,7 \times 10^{38}$	6 bajtů
$1,5 \times 10^{-45} .. 3,4 \times 10^{38}$	4 bajty
$5,0 \times 10^{-324} ... 1,7 \times 10^{308}$	8 bajtů
$3,4 \times 10^{-4932} .. 1,1 \times 10^{4329}$	10 bajtů
$-2^{63} + 1 .. 2^{63} - 1$	8 bajtů

Číselné datové typy a operace

- INTEGER (+ - * DIV MOD /...vrací real)
- REAL (můžeme do něj přiřazovat celá čísla, nelze použít funkci DIV a MOD)

Datový typ ŘETĚZEC

- Znakový řetězec - *STRING*
- Homogenní strukturovaný datový typ
- Slouží k ukládání více znaků
- Obsahuje několik (až 255 položek)
 - PC si vyhradí v paměti místo pro 255znaků, šetříme-li paměť, definujeme velikost (var Text2:String[10])
- Všechny položky jsou stejného typu
- Var *retez: string;*
- Funkce *LENGHT(retez)* - délka řetězce (počet znaků)
delka:=lenght(retez)

BOOLEAN – logický datový typ

- Může nabývat jen dvou hodnot TRUE/
FALSE
- Operace: NOT, AND, OR, XOR
- Pravdivostní tabulky

Datový typ POLE (array)

- Jednorozměrné – vektor
- Dvojměrné – matice
- Vícezměrné
- Počet položek není omezen, max. počet se musí deklarovat (předem dané místo v paměti)
- Všechny položky pole musí být stejného typu (celočíselné pole, pole znakových řetězců, pole logických hodnot, ..)

Deklarace proměnná POLE

1. Přímá deklarace

Array [1 .. N, 1 .. M] of datový typ

Příklad:

```
Var A: array [1 .. 100] of integer;
```

```
Var B: array [1 .. 100] of real;
```

```
Var C: array [1 .. 100] of string;
```

2. Nepřímá deklarace - pomocí nového datového typu

- Pomocí klíčového slova **TYPE** deklarovat nový datový typ

Příklad:

- deklarace pomocí **VAR** všechna pole tohoto typu

```
TYPE POLICKO = array [1 .. 100] of integer;
```

```
Var A,B : POLICKO;
```

Datový typ ZÁZNAM (record)

- Umožňuje vkládat data různých datových typů
- Všechny položky musí být předem deklarovány

Type

```
TypZaznam=record  
  Jmeno:String[25];  
  Vek:integer;  
  Plat:real;  
  MaAuto.Boolea;  
End;
```

Var

```
Osoba:TypZaznam
```

Zadání/tisk dat do/z proměnné typu record

- Je třeba určit, do jaké části proměnné se má zadaná hodnota uložit

```
osoba.Jmeno:= 'Karel';
```

```
osoba.Vek:=31;
```

Konverze datových typů

- Používá se v situaci, kdy je potřeba s určitým výrazem, hodnotou, nebo proměnnou pracovat, jako by byla jiného typu.

1) Přetypování – nejprve se uvede nový typ a do závorky se dá výraz (proměnná), kterou chceme přetypovat

- pozor – velikost musí být stejná, nemíchat celočíselné a reálné typy !

Příklad: Integer ('A'); Boolean(0);
Char (48); I: = Integer('A') (v I bude 64)
Char (POMOCNA);

2) Konverzní funkce

- Funkce knihovny, které slouží ke konverzi datových typů

IntToStr	StrToInt	Konverze celého čísla na řetězec a naopak
FloatToStr	StrToFloat	Konverze reálného čísla na řetězec a naopak

Příklad:

```
Var  b:string;  
      a:integer;
```

....

```
B:= IntToStr(a);
```