



# Čísla v algoritmech a programech

- $10^{26}$  Poloměr vesmíru
- 2828 studujících studentů MFF UK
- 3.142857... Ludolfovo číslo
- $10^{16}$  stáří vesmíru v sekundách
- 3 délka bakalářského studia na MFF UK v letech
- 36524 délka života stoletého člověka (ve dnech) (!)
- ...

# Čísla v algoritmech a programech

=> pracujeme se dvěma typy čísel:

## CELÁ

malý omezený rozsah  
přesné hodnoty

## REÁLNÁ

velký rozsah  
přibližné hodnoty  
neumíme representovat všechny hodnoty

# typ Integer

## HODNOTY

Celá čísla zobrazitelná v počítači

**ROZSAH** se může lišit v jednotlivých implementacích,  
proto konstanta **MAXINT**, rozsah je

**-MAXINT . . +MAXINT**

, typicky  $2^{15}$  nebo  $2^{31}$

# typ Integer - operace

## OPERÁTORY TYPU NÁSOBENÍ

- \* násobení
- div dělení
- mod zbytek po dělení
- / dělení s výsledkem typu real

## OPERÁTORY TYPU SČÍTÁNÍ

- + sčítání
- odčítání

## RELAČNÍ OPERÁTORY

=, <>, <, >, <=, >=

# Celočíselné dělení a zbytek

$i \text{ div } j$      $i \text{ mod } j$

je jasné, jaké hodnoty budou výsledkem, pro  $i \geq 0$ ,  $j > 0$

Mimo tento rozsah - raději vyzkoušet!

# Priorita operátorů

1. operace typu násobení

2. operace typu sčítání

3. relační operátory

V případě stejné priority - odleva.

# typ Integer - funkce

**ABS ( x )**     absolutní hodnota

**SQR ( x )**     druhá mocnina

**ODD ( x )**     x je liché



**POZOR! POZOR! POZOR! POZOR!**

**Správná hodnota aritmetického výrazu je zaručena pouze tehdy, když jsou všechny argumenty i mezivýsledky hodnotami typu (integer)!**

**Kdyby MAXINT byl 1000:**

**600+500-400**

**(běhové kontroly)**

# Turbo/Borland (/Free) Pascal

**integer**      2B

**shortint**    1B

**longint**     4B

**byte**         1B

**word**         2B

# typ **Real**

## HODNOTY

Racionální čísla,  
jen konečná množina,  
výsledek je jen **APROXIMACE** správného výsledku.

## VÝSLEDNÉ CHYBY záleží

na representaci čísel  
na řešení úloze  
na zvoleném algoritmu.

Odhady velikosti chyb se zabývá **NUMERICKÁ MATEMATIKA**.

# typ Real - zápis konstant

## HODNOTY

desetinná tečka a nepovinně číslice za ní  
nepovinně exponent ve tvaru E<integer>

### SEMILOGARITMICKÝ TVAR

0.3768

-326.21

1234.

0.03E-4

-10.583E60

**POZOR! POZOR! POZOR! POZOR!**

V paměti je jiná reprezentace  
než desítkový zápis

=>

při vstupu a výstupu dochází  
k zaokrouhlování!

**POZOR! POZOR! POZOR! POZOR!**

# Co udělá program?

# Poučení

Testovat reálná čísla  
(vypočtená nebo načtená)  
na rovnost nemusí mít dobrý smysl !

Místo toho:

```
if abs(koruny1-koruny2) < Epsilon  
    then
```

# typ Real - operace

- \* + - výsledek je typu Real,  
je-li alespoň jeden z operandů  
typu Real
- / výsledek je vždy typu Real

Takže  $4 / 2$  je typu Real.



# typ Real – funkce

ABS( x )

SQR( x )

SIN( x )

ARCTAN( x )

EXP( x )

TRUNC( x )

SQRT( x )

COS( x )

LN( x )

ROUND( x ) rozsah

# Složitost algoritmů

potřeba ČASU, PAMĚTI...

## Jak ji měřit?

- . NE absolutně
- . NE pro konečně mnoho úloh

## Raději:

„Co se stane, když se velikost úlohy zdvojnásobí?“

# Složitost algoritmů

Formálně:

$N$  ..... velikost úlohy

$f(N)$  .... spotřeba (času nebo paměti)

- . v nejhorším případě
- . v průměrném případě

# Složitost algoritmů

Definice:

Funkce  $g$  je  $O(f)$

když

$$\exists m_0, C$$
$$\forall n > m_0: g(n) \leq C \cdot f(n)$$

# Složitost algoritmů - Příklad

```
var x,i,N,Sum: integer;
```

```
begin
```

```
  read( N );
```

```
  Sum := 0;
```

```
  i := 1;
```

```
  while i <= N do
```

```
  begin
```

```
    read( x );
```

```
    Sum := Sum + x;   i := i+1;
```

```
  end;
```

```
  writeln( Sum / N )
```

```
end
```

časová složitost  $O(N)$

# Složitost algoritmů - Příklady

složitost	1 sekunda	1 hodina	1000 hodin
$n$	1	3.600	3.600.000
$n \log n$	1	568	278.000
$n^2$	1	60	1.897
$n^3$	1	15	153
$2^n$	1	12	22

**POZOR! POZOR! POZOR! POZOR!**

**Složitost neříká (skoro) nic  
o konkrétním případě!**

## Příklad:

algoritmus	složitost
A1	$1000 n$
A2	$100 n \log n$
A3	$10 n^2$
A4	$n^3$
A5	$2^n$

? Který algoritmus je nejrychlejší?



<b>N</b>	<b>nejrychlejší algoritmus</b>
<b>2..9</b>	<b>A5</b>
<b>10..35</b>	<b>A3</b>
<b>36..22026</b>	<b>A2</b>
<b>22027..</b>	<b>A1</b>

