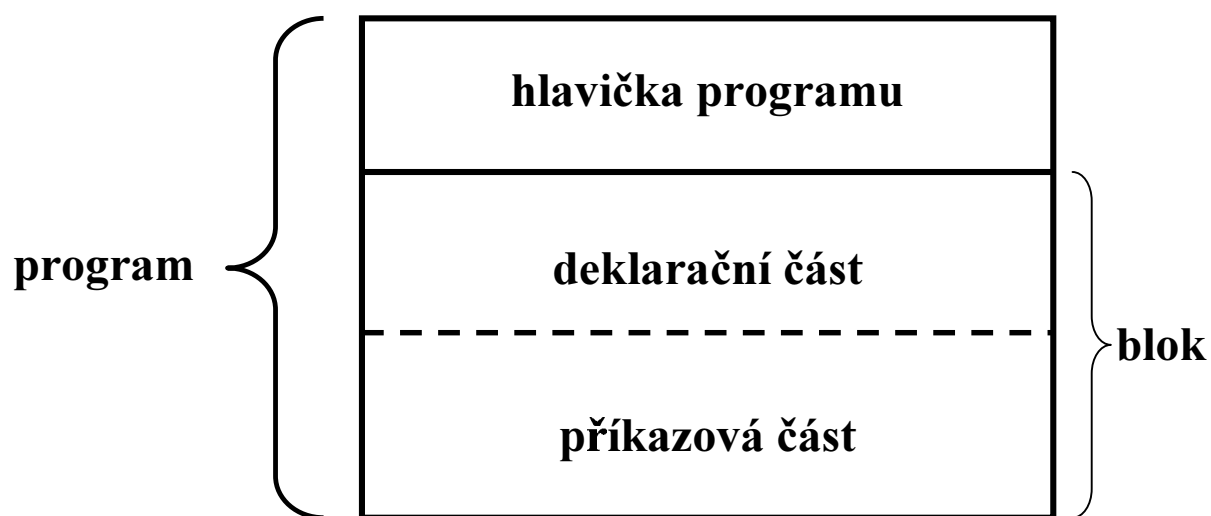
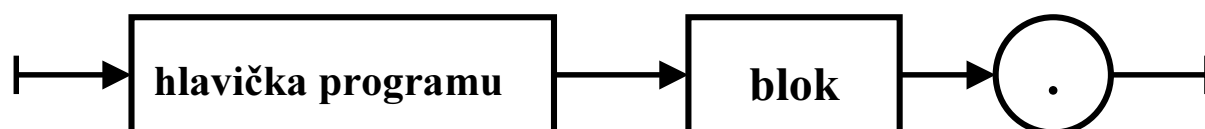


Struktura programů v PASCALU

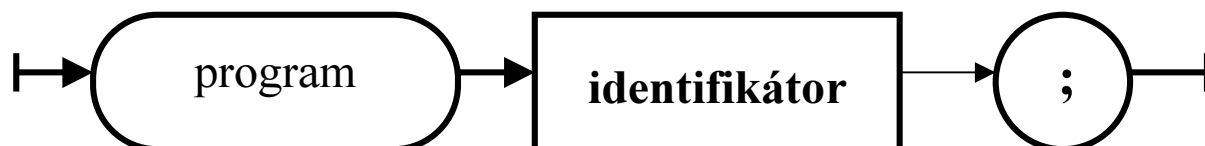


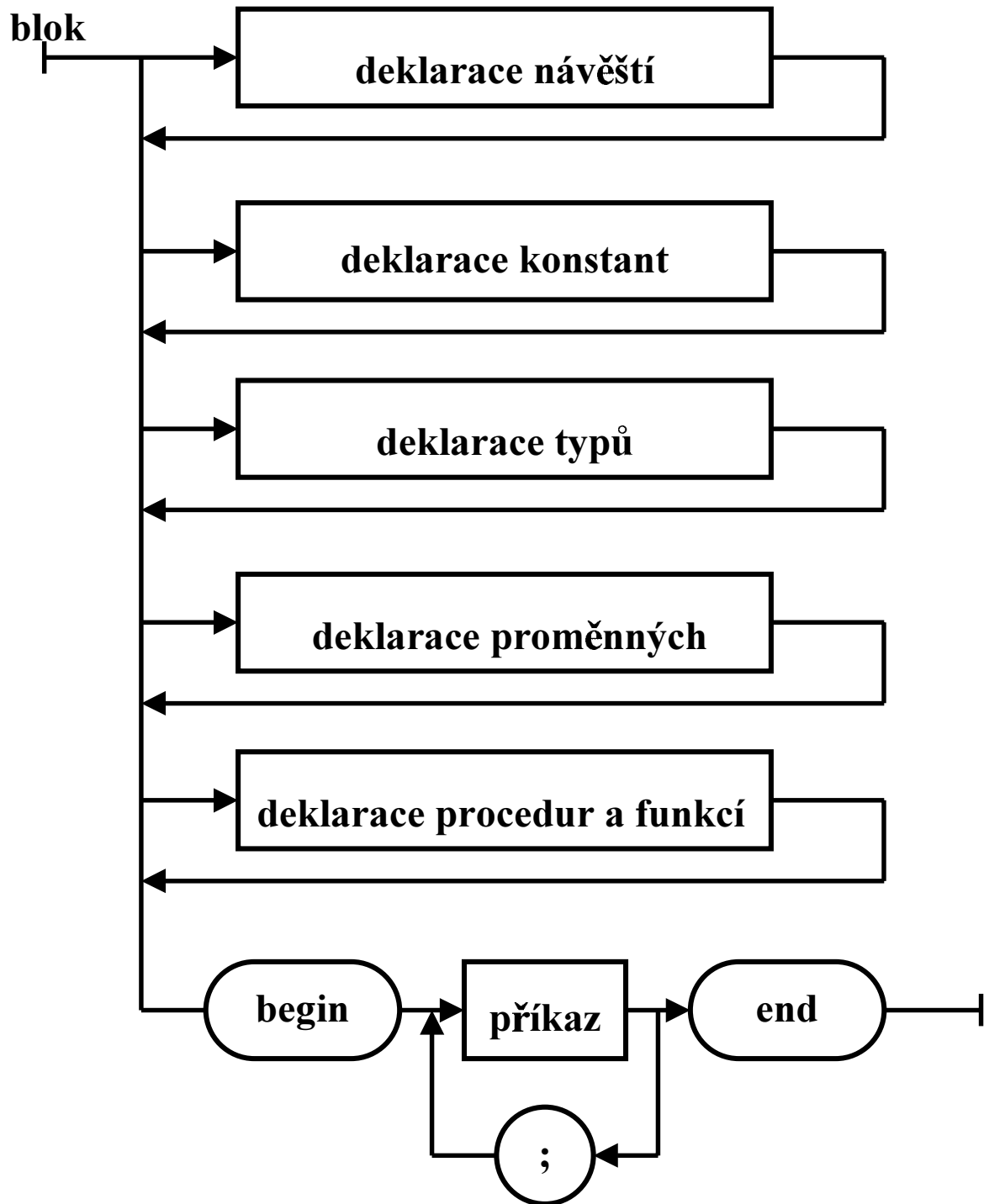
Syntaktické diagramy

program



hlavička programu





Deklační část - obsahuje definice významu identifikátorů použitých v programu

Příkazová část - má tvar složeného příkazu, je to posloupnost příkazů oddělovaných středníkem začínající klíčovým slovem *begin* a končící slovem *end*

Úprava programů: Pascal nepředepisuje úpravu zápisu programu, nutné je pouze vzájemně oddělovat identifikátory, klíčová slova a čísla znakem mezera , popř. novým řádkem.

Pro další zápis programů :

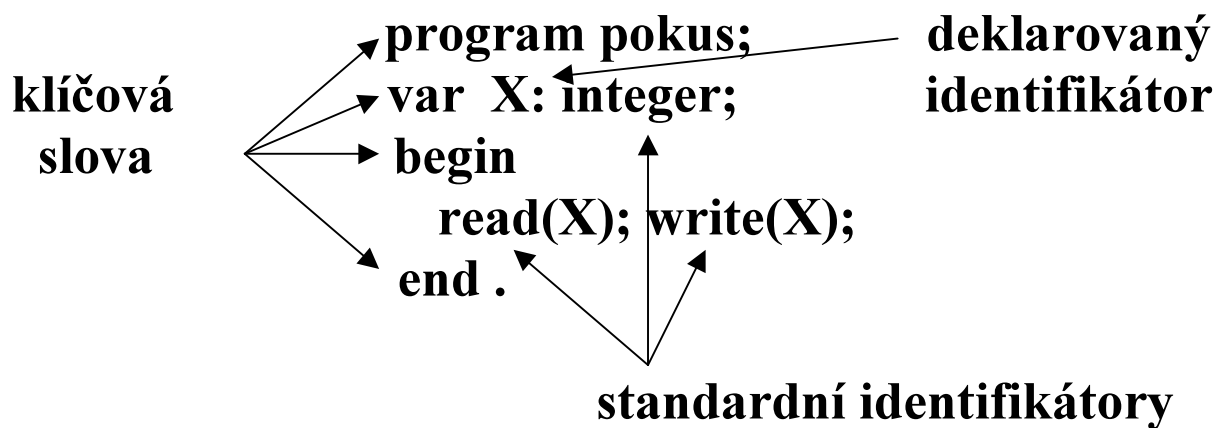
klíčová slova - malým písmem
identifikátory - velkým písmem

Identifikátory

- slouží k označování proměnných, konstant, typů, procedur a funkcí.
 - jsou to posloupnosti písmen a číslic začínající písmenem (velikost písmen nehraje roli), v identifikátoru je možné použít znak podtržení _
 - každý identifikátor použitý v programu musí mít definovaný význam
-
- standardní identifikátory - jejich význam je dán jazykem
 - ostatní identifikátory - význam je dán deklarací

Každý nestandardní identifikátor musí být deklarován !!!!

Př.



Konstanty

- datové objekty, které nemění hodnotu v průběhu výpočtu

1. literály = přímé zápisy hodnot.

Př.

Literály typu:

- integer 25 -191 34
- real 25.6 -35.0 0.25E+4
- boolean true false
- char 'A' '-' ' '
- řetězec 'ALFA'

2. deklarované konstanty

```
const identifikátor=hodnota;  
    identifikátor=hodnota;  
    ....  
    identifikátor=hodnota;
```

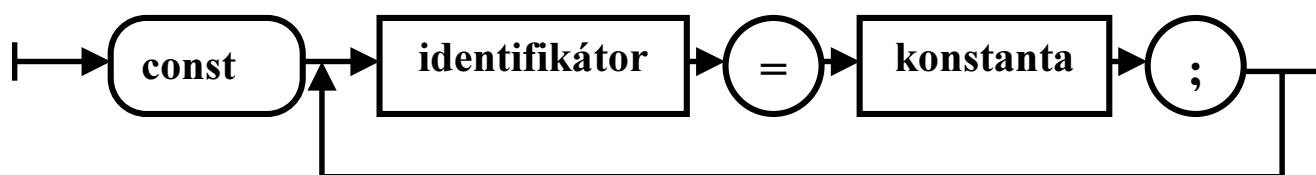
Příklad:

```
const PI = 3.14159;  
    DOLNIMEZ=1;  
    HVEZDICKA='*';  
    MAX = 100;  
    MIN = -MAX;
```

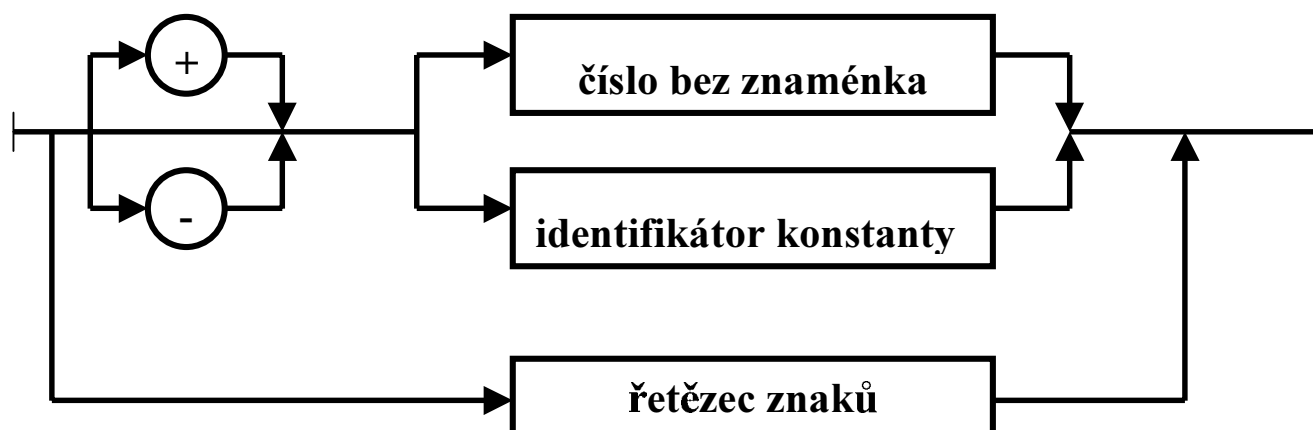
Typ deklarovaných konstant není dán explicitně, ale vyplývá z typu hodnoty, která se konstantě přiřazuje.

Identifikátor konstanty se nesmí vyskytnout na levé straně přiřazovacího příkazu !!!!!!!

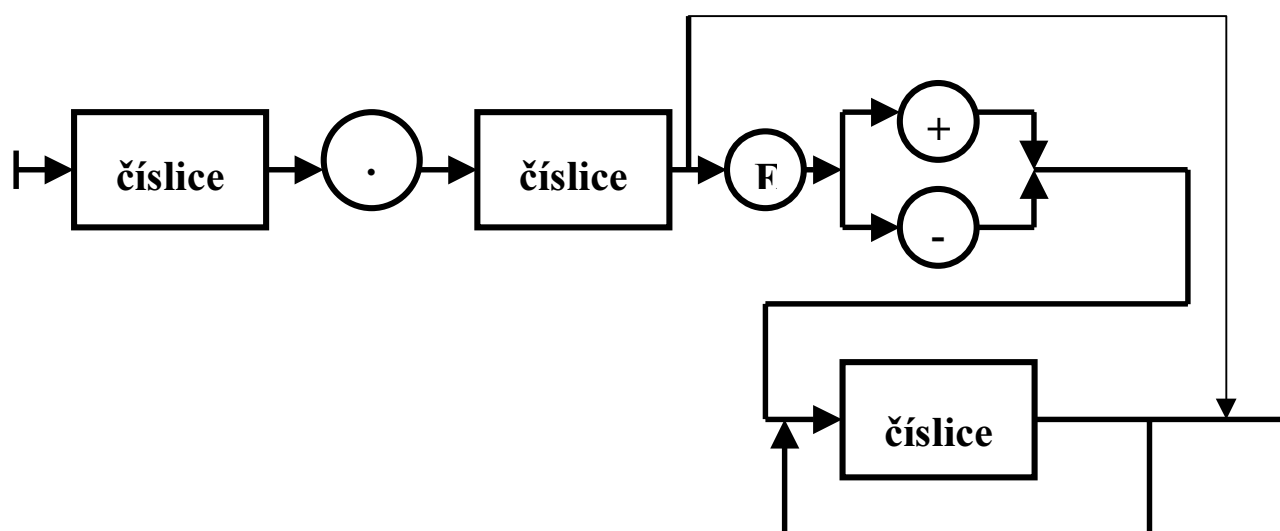
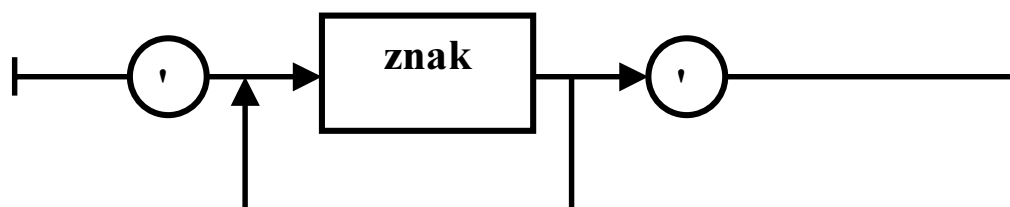
Deklarace konstant



konstanta



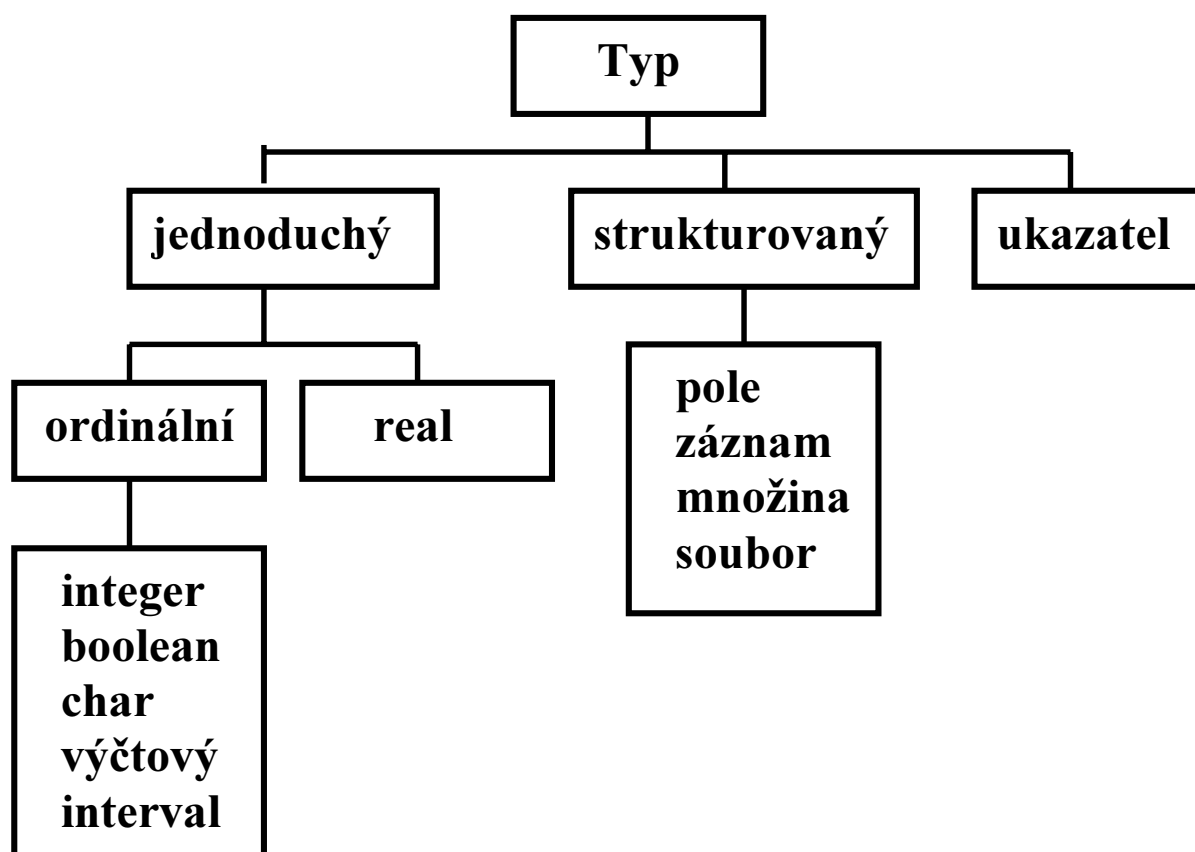
řetězec znaků



Proměnné

- datový objekt, který má své jméno a hodnotu. Obor hodnot a operace s proměnnou jsou určeny *typem* proměnné

Hodnotu proměnné definuje přiřazovací příkaz, popř. příkaz vstupu, nikoliv její deklarace (po deklaraci má proměnná nedefinovanou hodnotu) !!!!



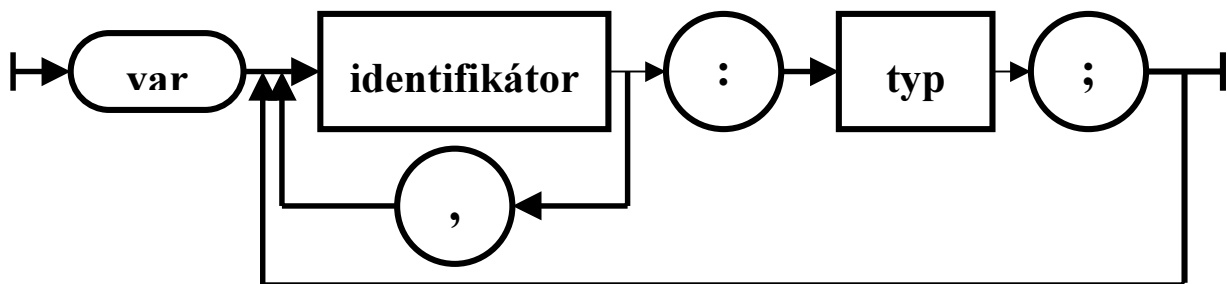
Deklarace proměnných

```
var seznam proměnných: typ;  
    seznam proměnných: typ;  
    .....  
    seznam proměnných: typů
```

Příklad:

```
var DOLNI: integer;  
    MAX,MIN: real;  
    KONC_ZNAK: char;  
    KONEC: boolean;
```

úsek deklarací proměnných



typ



Celočíselné datové typy (integer)

- základní typ *integer* dovoluje pracovat s proměnnými v rozsahu

$< -(Maxint+1), Maxint >$

Maxint je konstanta definovaná Pascalem (pro Turbo Pascal je *Maxint* = 32767) (2 byty)

Další celočíselné typy definované v Turbo Pascalu:

Typ	Rozsah	Počet slabik
byte	0 .. 255	1
shortint	-128 .. 127	1
word	0 .. 65535	2
integer	-32768 .. 32767	2
longint	-2147483648 .. 2147483648	4

Operace s celočíselnými typy

- aritmetické (všechny operandy musí být celočíselné, výsledek je celočíselný)

+ sčítání div celočíselné dělení

- odčítání mod zbytek po dělení

* násobení

- relační (operandy celočíselné, výsledek je typu boolean)

= rovnost

<> nerovnost

< menší než

> větší než

<= menší nebo rovno

>= větší nebo rovno

- standardní funkce (argument celočíselný, výsledek celočíselný)

abs(x) výsledkem je absolutní hodnota x

sqr(x) výsledkem je x^2

odd(x) výsledkem je *true*, je-li x liché, jinak *false*

- vstup (zadávat pouze celočíselný literál, jinak dochází k chybě při běhu programu)

read(x)

readln(x)

- výstup

write(x:c)

writeln(x)

c - celkový počet znaků, které se vytisknou

Typy reálných čísel

- základní typ *real* umožňuje pracovat s reálnými čísly v rozsahu

$$< -10^{38}, 10^{38} >$$

s přesností na 11 až 12 platných číslic (číslo je uloženo na 6 bytech).

Další typy pro práci s reálnými čísly v Turbo Pascalu

Typ	Rozsah	Platné číslice	Počet slabik
real	$2.9 \cdot 10^{-39}$ až $1.7 \cdot 10^{38}$	11-12	6
single	$1.5 \cdot 10^{-45}$ až $3.4 \cdot 10^{38}$	7-8	4
double	$5.0 \cdot 10^{-324}$ až $1.7 \cdot 10^{308}$	15-16	8
extended	$3.4 \cdot 10^{-4932}$ až $1.1 \cdot 10^{4932}$	19-20	10
comp	$2^{-63} + 1$ až 2^{63}	19-20	8

Kromě typu *real* musí být ostatní typy, pokud jsou používány překládány s využitím matematického koprocessoru (popř. jeho emulace), tj. překládat s direktivou `{N+}` (popř. `{E+}`).

Operace s typem *real*

- aritmetické operace (operandy mohou být typu *real*, popř. celočíselného typu, výsledek je typu *real*)

+ sčítání

* násobení

- odčítání

/ dělení

- standardní funkce (argumenty reálné popř. celočíselné, výsledek typu *real*)

abs(x)	absolutní hodnota x
sqr(x)	x^2
sqrt(x)	druhá odmocnina $x \geq 0$ (pro $x < 0$ chyba)
sin(x)	funkce sinus (x je v radiánech)
cos(x)	-''- cosinus (-''-)
exp(x)	x-tá mocnina e
ln(x)	přirozený logaritmus $x > 0$
arctan(x)	hlavní hodnota arkus tangenty x v radiánech

- relační operátory (stejně jako u celočíselných typů)
- vstup (-''-)

- výstup

write(x) - výstup v semilogaritmickém tvaru
write(x:c) - - '' - se zvoleným počtem míst
write(x:c:d) - výstup v desetinném tvaru

c - celkový počet znaků (doplnění zleva mezerami)
d - počet desetinných míst zprava

(stejný formát platí i pro writeln)

Vnitřní reprezentace čísel typu integer

- čísla *integer* jsou zobrazena v paměti pevným počtem n bitů (obvykle $n=16$ bitů, popř. viz tabulka)
Pomocí n bitů je možné zobrazit 2^n různých hodnot, tj. interval od 0 do 2^n-1 .

Čísla typu integer se zobrazují tak, že se vhodnou transformací převedou na interval $\langle 0, 2^n-1 \rangle$.

Nejčastější transformace:

- kód s posunutou nulou
- přímý kód
- doplňkový kód

Kód s posunutou nulou

$$x' = x + M/2 \quad \text{nebo} \quad x' = x + M/2 - 1, \quad \text{kde } M=2^n$$

x' - číslo v kódu s posunutou nulou

x - převáděné číslo

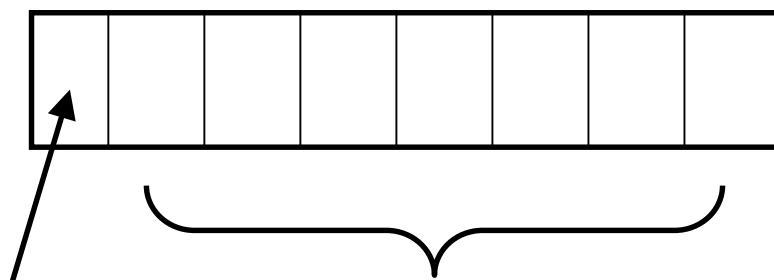
Rozsah zobrazení :

$$-M/2 \leq x \leq M/2 - 1 \quad \text{nebo} \quad -M/2 + 1 \leq x \leq M/2$$

$$n=8 \quad -128 \leq x \leq 127 \quad \text{nebo} \quad 127 \leq x \leq 128$$

Použití: zobrazení exponentu u čísel v pohyblivé řádové čárce

Přímý kód



znaménko absolutní hodnota čísla

0 +

1 -

$x' = x$ pro $x \geq 0$

$x' = M/2 - x$ pro $x \leq 0$

Rozsah zobrazení:

$$-M/2 + 1 \leq x \leq M/2 - 1$$

n=8 $-127 \leq x \leq 127$

**Nevýhody : hodnota 0 má dva obrazy (-0 , +0)
 komplikované základní operace**

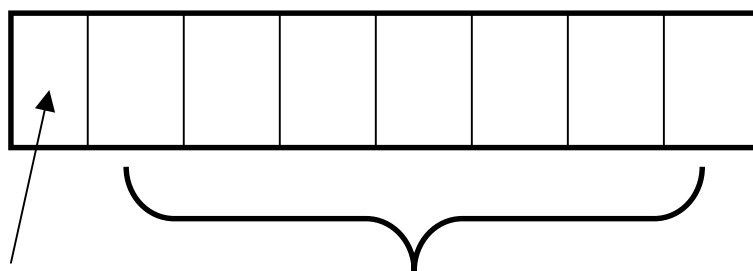
**Použití : zobrazení celých čísel u starších typů
 počítačů**

Příklad: (n = 8 bitů)

-10 = 10001010 +10 = 00001010

-127 = 11111111 +127 = 01111111

Doplňkový kód



znaménko

0	+	hodnota čísla ve dvojkové soustavě
1	-	dvojkový doplněk čísla

$$\begin{array}{ll} x' = x & \text{pro } M/2 - 1 \geq x \geq 0 \\ x' = x + M & \text{pro } -M/2 \leq x < 0 \end{array}$$

Rozsah zobrazení:

$$-M/2 \leq x \leq M/2 - 1$$

$$n=8 \quad -128 \leq x \leq 127$$

Výhody: jediná nula, operaci odčítání lze převést na sčítání

Použití : nejčastější způsob zobrazení celých čísel v počítači

Příklad: (n= 8 bitů)

+5 = 00000101	-5 = 11111011
+127 = 01111111	-128 = 10000000
+1 = 00000001	-1 = 11111111

Vnitřní reprezentace čísel typu *real*

- zobrazení reálných čísel se provádí v pohyblivé řádové čárce, tj. ve tvaru

$$x = \text{mantisa} * \text{základ}^{\text{Exponent}}$$

V počítači je pak číslo zobrazováno jako dvojice (Mantisa, Exponent), přesnost zobrazení závisí na počtu číslic (bitů) mantisy, rozsah zobrazení na počtu číslic (bitů) exponentu.

Typ single – 4 byty (32 bitů)

Formát IEEE (počítače PC)

Z	exponent	mantisa
31 30	23 22	0

Z - znaménko mantisy (0= + , 1 = -)

exponent - zobrazení v kódu s posunutou nulou (0= 127)

mantisa - zobrazení v přímém kódu v normalizovaném tvaru se skrytým bitem tj. $1 \leq |m| < 2$

Hodnota čísla h se určí na základě následujících pravidel:

if $0 < e \leq 255$ then $h = (-1)^Z * 2^{(e-127)} * (1.m)$

if $e=0$ and $m \neq 0$ then $h = (-1)^Z * 2^{(-126)} * (0.m)$

if $e=0$ and $m=0$ then $h = (-1)^Z * 0.$

if $e=255$ and $m=0$ then $h = (-1)^Z * \text{Inf.}$

if $e=255$ and $m \neq 0$ then $h = \text{NaN}$

Příklad: zobrazit číslo -10

$10 = (1010)_2 \longrightarrow \text{normovaný tvar} = 1.010 * 2^3$

↑
skrytý bit (nezobrazuje se)

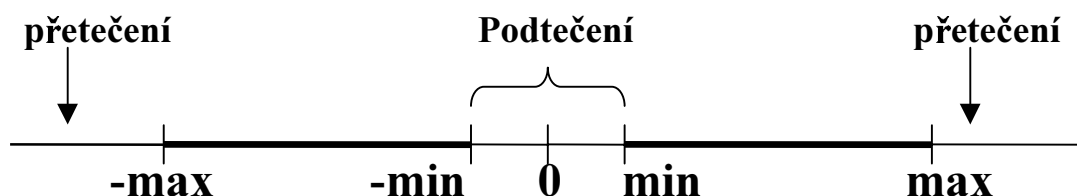
1 10000010 01000000

exponent mantisa

Příklady zobrazení:

Číslo	Z	Exponent	Mantisa
0	0	00000000	0000...0000
0	1	00000000	0000...0000
+1	0	01111111	0000...0000
-1	1	01111111	0000...0000
$+\infty$	0	11111111	0000...0000
$-\infty$	1	11111111	0000...0000

Přetečení a podtečení



Základní typ real (6 bytů)



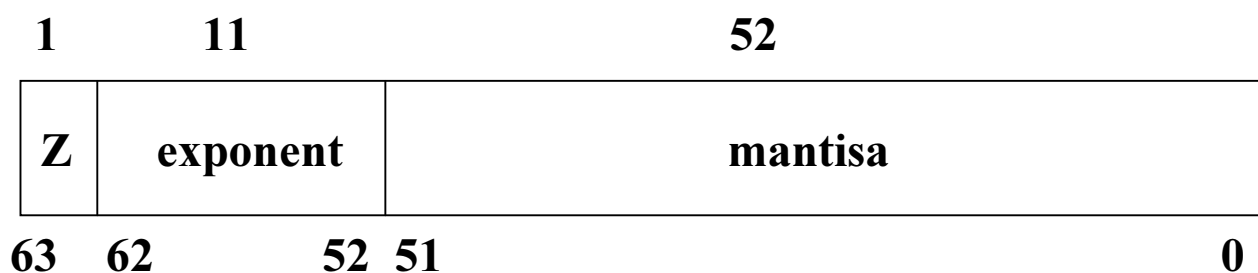
Hodnota čísla h se určí na základě následujících pravidel:

if $0 < e \leq 255$ then $h = (-1)^Z * 2^{(e-129)} * (1.m)$

if $e=0$ and then $h=0$

V základním typu real není možné uchovávat nenormalizovaná čísla, NaN, a nekonečno

Typ double



Hodnota čísla h se určí na základě následujících pravidel:

if $0 < e \leq 2047$ then $h = (-1)^Z * 2^{(e-1023)} * (1.m)$

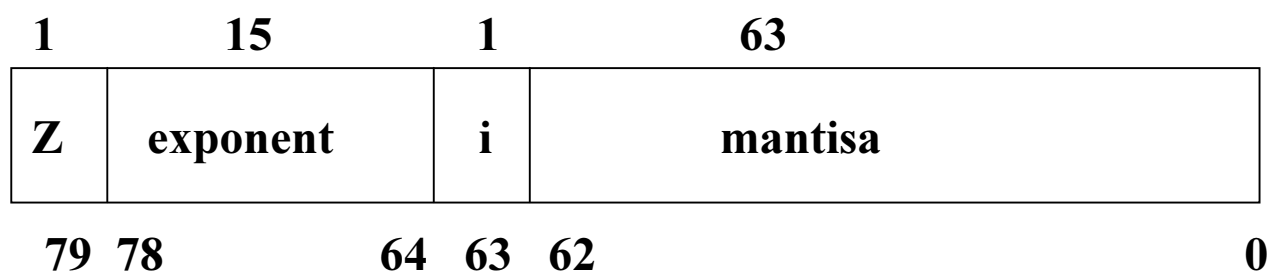
if $e=0$ and $m \neq 0$ then $h = (-1)^Z * 2^{(-1022)} * (0.m)$

if $e=0$ and $m=0$ then $h = (-1)^Z * 0.$

if $e=2047$ and $m=0$ then $h = (-1)^Z * \text{Inf.}$

if $e=2047$ and $m \neq 0$ then $h = \text{NaN}$

Typ *extended* (10 byte)



Hodnota čísla h se určí na základě následujících pravidel:

if $0 < e \leq 32767$ then $h = (-1)^Z * 2^{(e-16383)} * (i.m)$

if $e = 32767$ and $m = 0$ then $h = (-1)^Z * \text{Inf.}$

if $e = 32767$ and $m \neq 0$ then $h = \text{NaN}$