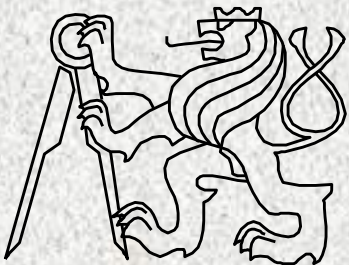


Jak v Javě – primitivní datové typy a jejich reprezentace



BD6B36PJV 002
Fakulta elektrotechnická
České vysoké učení technické

Obsah

- Celočíselný datový typ
- Reálný datový typ
- Logický datový typ, typ Boolean
- Znaky
- Kompatibilita datových typů
- Priorita operátorů

Datové typy

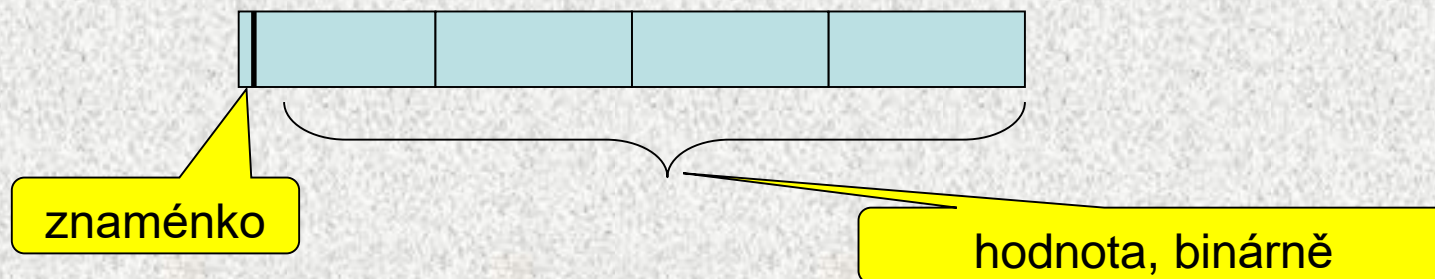
- Při návrhu algoritmů a psaní programů ve vyšších programovacích jazycích abstrahujeme od binární podoby paměti počítače
- S daty pracujeme jako s hodnotami různých datových typů, které jsou uloženy v datových objektech
- Datový typ (zkráceně jen typ) specifikuje:
 - množinu hodnot (+ způsob reprezentace)
 - množinu operací, které lze s hodnotami daného typu provádět
- Příklad typu: celočíselný typ `int` v jazyku Java:
 - množinou hodnot jsou celá čísla z intervalu `-2147483648 .. 2147483647`
 - množinu operací tvoří
 - aritmetické operace `+`, `-`, `*`, `/`, jejichž výsledkem je hodnota typu `int`
 - relační operace `==`, `!=`, `>`, `>=`, `<`, `<=`, jejichž výsledkem je hodnota typu `boolean`
 - a další
- Typ `int` je jednoduchý typ, jehož hodnoty jsou atomické (z hlediska operací dále nedělitelné)



graficky

Celá čísla v Javě – typ int

int je reprezentováno 32 bity



největší číslo $0111\dots111 = 2^{31} - 1 = 2147483647$

Pro zobrazování záporných čísel – doplňkový kód

nejmenší číslo v doplňkovém kódu $1000\dots000 = -2^{31} = -2147483648$

Reprezentace záporných celých čísel

- Doplnkový kód – $D(x)$
 - předpokládejme reprezentaci čísel pomocí 8 bitů, lze reprezentovat $2^8 = 256$ čísel = r (rozsah)

$$D(x) = \begin{cases} x, & \text{pro } 0 \leq x < r/2 \\ r+x, & \text{pro } 0 > x \geq -r/2 \end{cases}$$

*Pozor na důsledky počítání
(v doplňkovém kódu) pro `int`:*
 $2\,000\,000\,000 + 1\,000\,000\,000 =$
 $-1\,294\,967\,296$

desítkově	doplňkový kód
0 – 127	00000000 – 01111111
128	nelze zobrazit na 8 bitů v d.k.
-128	$D(-128) = -128+256 = 128 = 10000000$
-1	$D(-1) = -1+256 = 255 = 11111111$
-4	$D(-4) = -4+256 = 252 = 11111100$

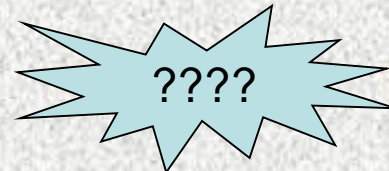
Nepřesnost v zobrazení čísel

Nepřesnosti způsobují

1. Čísla, jež mají v dané soustavě periodický rozvoj (1/3 ve dvojkové či dekadické soustavě, 1/10 ve dvojkové)
2. iracionální čísla (e , π , $\sqrt{2}$,.....)
3. Čísla, která mají příliš dlouhý zápis

Implementace `double` (na 64 bity)

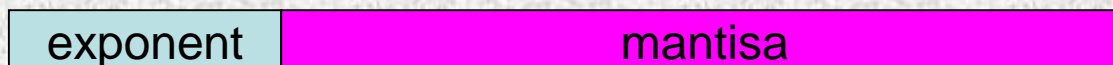
- 1bit znaménko – dvě možnosti, +,-
- 11 bitů exponent – 2048 možností
- 52 bitů – 4 503 599 627 370 496 možností, tedy asi 4,5 biliardy
 - není možné v typu `double` přesně uložit čísla se zápisem delším než 52 bitů!
 - čím větší exponent tím větší „mezery“ mezi sousedními aproximacemi!!!



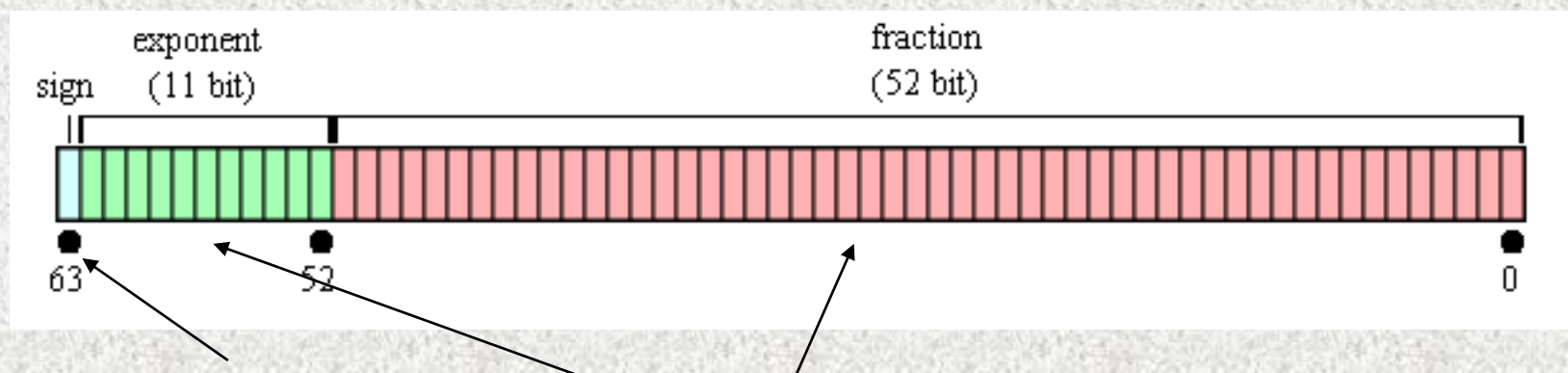
Necelá čísla v Javě – typ double

Reálná čísla X se zobrazují ve tvaru:

$$X = \text{mantisa} * \text{základ}^{\text{exponent}}$$



double je reprezentován 64 bity, norma IEEE 754



znaménkový bit (s), exponent, mantisa

Nejmenší čísla v normalizovaném tvaru

$$\pm 2^{-1024} \approx \pm 2.2250738585072020 \times 10^{-308}$$

Největší čísla

$$\pm (1 - (1/2)^{52}) 2^{1024} \approx \pm 1.7976931348623157 \times 10^{308}$$

Model reprezentace reálných čísel

Reálná čísla se zobrazují jako aproximace daných rozsahem paměťového místa

Reálná čísla se zobrazují ve tvaru:

$$X = \text{mantisa} * \text{základ}^{\text{exponent}} = m * z^{\text{exponent}}$$

Mantisa musí být normalizována:

$$0,1 \leq m < 1, \text{ důvod: jednoznačnost zobrazení}$$

Model – „dekadický počítač“

- délka mantisy 3 pozice + znaménko
- délka exponentu 2 pozice + znaménko

Příklad

$$X = 77.5 = +0.775 * z^{+02}$$

+|775|+|02

1|775|1|02

- zakódujeme znaménko “+” = 0, “-” = 1
- z je pro názornost 10

Model reprezentace reálných čísel

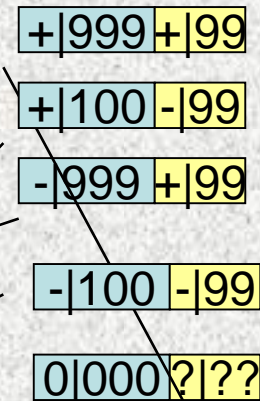
Maximální zobrazitelné kladné číslo

Minimální zobrazitelné kladné číslo

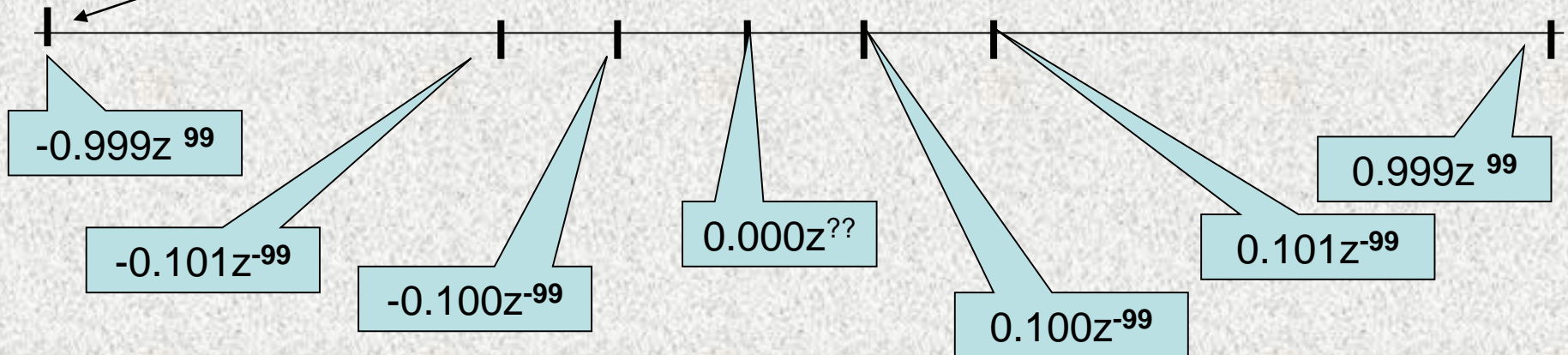
Maximální zobrazitelné záporné číslo (v absolutní hodnotě)

Minimální zobrazitelné záporné číslo (v absolutní hodnotě)

Nula



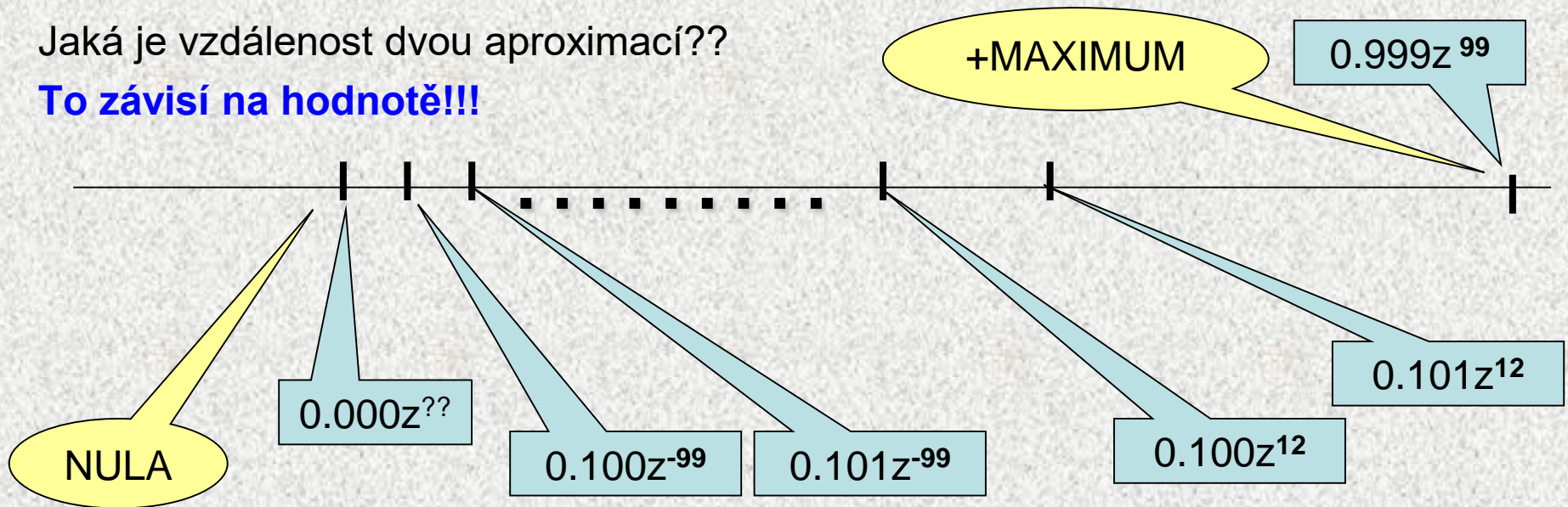
Jak je to se zobrazitelnými hodnotami „okolo nuly“,



Model reprezentace reálných čísel

Jaká je vzdálenost dvou aproximací??

To závisí na hodnotě!!!

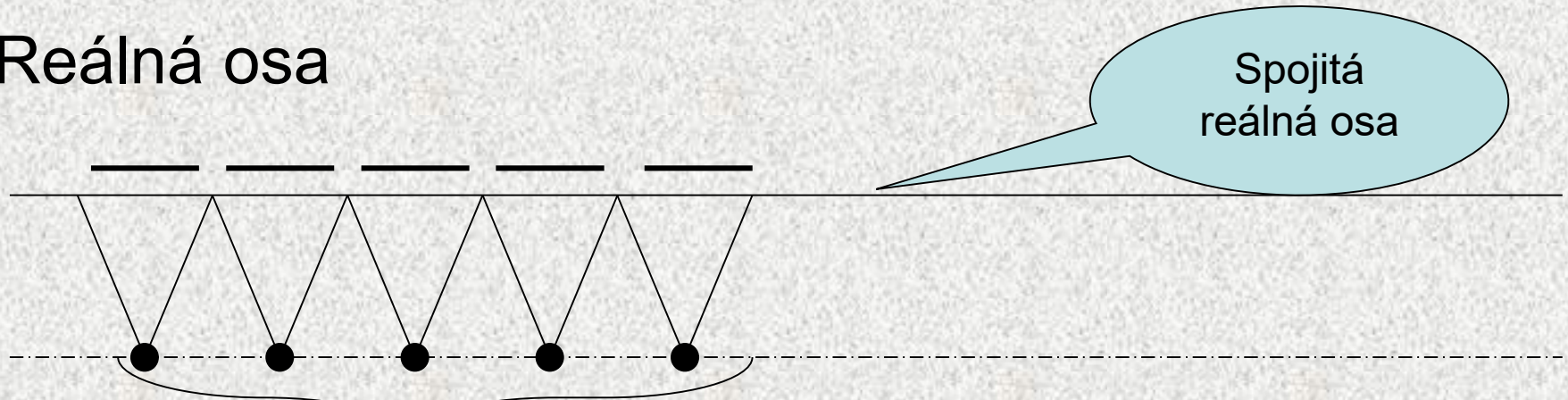


Aproximace reálných čísel: nejsou na číselné ose rovnoměrně rozložené!!



Reprezentace reálných čísel

Reálná osa



Zobrazení aproximací (pro jeden exponent!!)

Reálná čísla se zobrazují jako aproximace daných rozsahem paměťového místa

Diskrétní
aproximace v
paměti počítače

Reprezentace znaků

Java používá typ **char**, znaky jsou kódovány 16 bitovým kódem Unicode (UTF-16, podmnožina UCS-2).

- Odstraňuje problémy češtiny
- Programy v Javě provádí konverzi znaků mezi vnitřní reprezentaci v programech a OS, většinou automaticky podle informací OS (Locale)
- *Poznámky*
 - *Různé OS různá kódování*
 - *sedmibitové ASCII (American Standard Code for Information Interchange), osmibitová Win1250, ISO 8859-2, kód s proměnnou délkou znaku - UTF8(UCS Transformation Format), .*
 - *Java – UCS2*
 - *Universal Character Set - univerzální znaková sada definována normou ISO 10646*

Reprezentace znaků, část reprezentace

Dec	Hex	Znak	Význam
0	00	NUL	
1	01	☉	Start of Header (SOH)
2	02	☉	Start of Text (STX)
3	03	♥	End of Text (ETX)
4	04	♦	End of Transmission (EOT)
5	05	♣	Enquiry (ENQ)
6	06	♠	Acknowledge (ACK)
7	07	•	Bell (BEL)
8	08	▣	Backspace (BS)
9	09	▬	Horizontal Tab (HT)
10	0a	▣	Line Feed (LF)
11	0b	VT	Vertical Tab
12	0c	FF	Form Feed
13	0d	CR	Carriage Return
14	0e	SO	Shift Out

Dec	Hex	Znak
32	20	SPC
33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29)
42	2a	*
43	2b	+
44	2c	,
45	2d	-
46	2e	.

Dec	Hex	Znak
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4a	J
75	4b	K
76	4c	L
77	4d	M
78	4e	N

Dec	Hex	Znak
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6a	j
107	6b	k
108	6c	l
109	6d	m
110	6e	n

Typ boolean

- **true**
- **false**

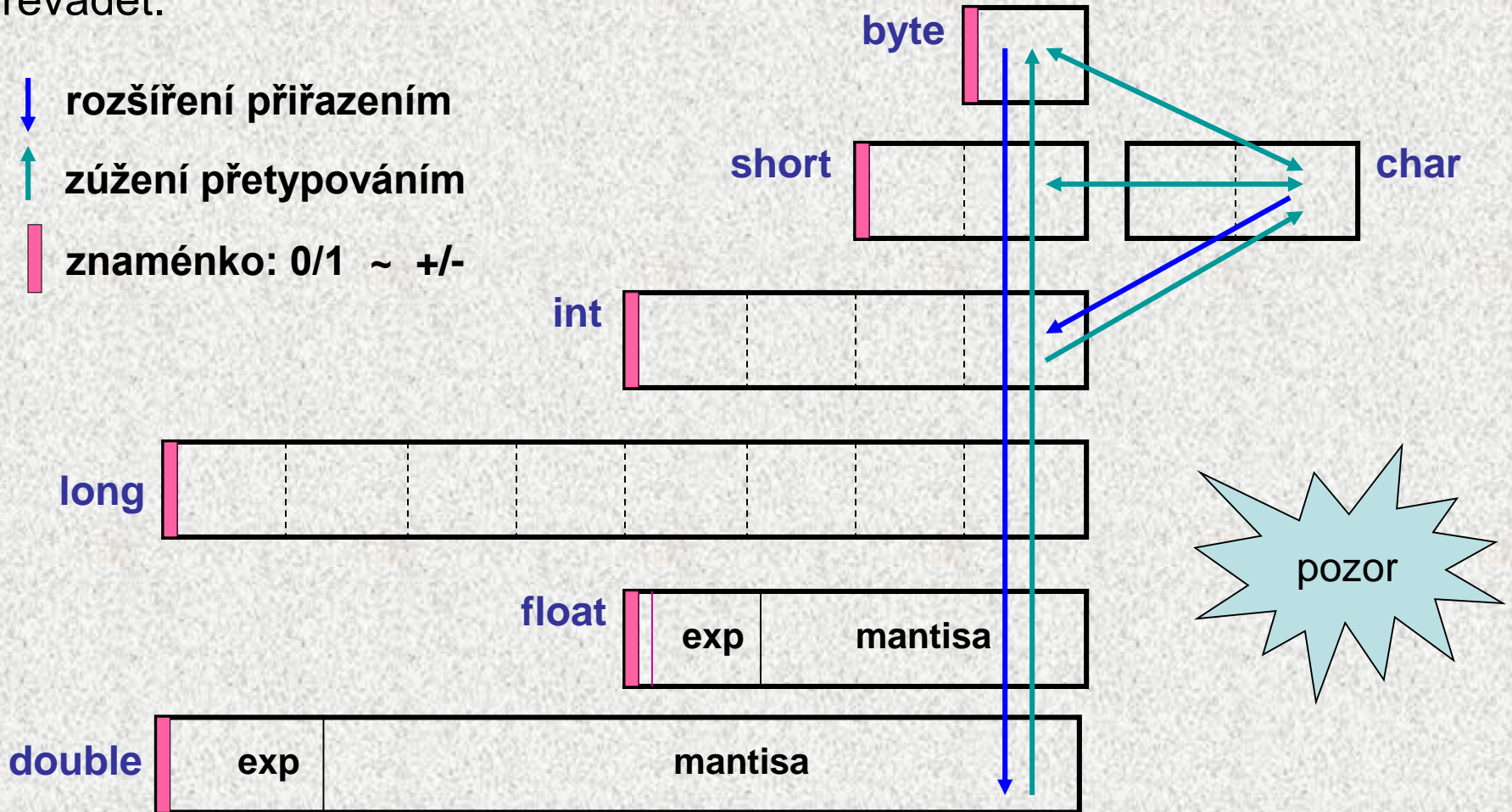
Reprezentace 1/8

Primitivní či základní datové typy

Typ	Bitů	Rozsah	Obal.třída
Celočíselný typ			
byte	8	-128 ... 127	Byte
short	16	-32768 ... 32767	Short
int	32	-2147483648 ... 2147483647	Integer
long	64	-9223372036854775808 ... 9223372036854775807	Long
Reálný typ, IEEE 754 (NaN, infinity)			
float	32	$2^{-149} \dots (2-2^{-23}) \cdot 2^{127}$	Float
double	64	$2^{-1074} \dots (2-2^{-52}) \cdot 2^{1023}$	Double
Znaky, UCS2			
char	16	'\u0000' to '\uffff' 0 ... 65535	Character
Logický typ			
boolean	1/8	true false	Boolean
Pomocný prázdný typ			
void			

Konverze primitivních typů

Primitivní typy jsou vzájemně nekompatibilní, ale jejich hodnoty lze převádět.



Operátory a jejich priorita

priorita	operátor	typ operandu	asociativita	operace
1	++	aritmetický	P	pre/post inkrementace
	--	aritmetický	P	pre/post dekrementace
	-	aritmetický	P	unární plus/minus
	~	celočíselný	P	bitová inverze
	!	logický	P	logická negace
	(typ)	libovolný	P	přetypování
2	*, /, %	aritmetický	L	násobení, dělení, zbytek
3	-	aritmetický	L	odečítání
	+	aritmetický, řetězový	L	sčítání, zřetězení

Operátory a jejich priorita II

priorita	operátor	typ operandu	asociativita	operace
4	<<	celočíslný	L	posun vlevo
	>>	celočíslný	L	posun vpravo
	>>>	celočíslný	L	posun vpravo s doplňováním nuly
	<,>, <=, >=	aritmetický	L	porovnání
	instanceof	objekt	L	test třídy
	==, !=	primitivní	L	rovno, nerovno
7	&	celočíslný nebo logický	L	bitové nebo logické AND
8	^	-"	L	bitové nebo logické XOR
9		-"	L	bitové nebo logické OR

Operátory a jejich priorita III

priorita	operátor	typ operandu	asociativita	operace
10	&&	logický	L	logické AND vyhodnocované zkráceně
11		logický	L	logické OR vyhodnocované zkráceně
12	? :	logický	P	podmíněný operátor
13	=, -=, *=, /=, % =, ==, & =, ^ =, =	libovolný	P	přiřazení