
 Aplikovaná informatika


Podklady předmětu
Aplikovaná informatika
pro akademický rok 2013/2014
Radim Farana

4

 Aplikovaná informatika 2

Obsah

- Datové typy
 - jednoduché,
 - složené,
- Programové struktury,
- Předávání dat.

 Aplikovaná informatika 3

Datové typy

- **Jednoduché**
 - Textové (alfanumerické)
 - Číselné
 - celočíselné
 - neceločíselné
 - s pevnou řádovou čárkou
 - s pohyblivou řádovou čárkou
 - Datum a čas
- **Složené**
 - Pole
 - Záznam



Textové kódy

- **blokové kódy** s různou délkou kódového slova a mohutností abecedy,
- 5 bitů – nejstarší kódy, Baudotův kód, kód CCITT, dvě znakové sady po 32 znaků,
- 7 bitů – American Standard Code for Information Interchange – ASCII – UNIX,
- 8 bitů – **ASCII-2**, znakové sady (1250) pro různé jazyky,
- 16 bitů – **Unicode**.



ASCII, ISO 8859-1 (Latin-1)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			.	f	?	^	%	Š	<	CE				Ÿ
90		.	"	"	"	"	"	"	™	š	>	œ				
A0		ı	é	ê	ë	ì	í	î	ï	š	™	«	»	–	®	–
B0		±	²	³	´	µ	¶	·	,	ı	»	¼	½	¾	¿	
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	



Textové datové typy

- Znak, Char, Character (1 B)
- String – posloupnost Byte (do 255 B)
 - s pevnou délkou
 - s proměnlivou délkou
- Memo (až 64 kB – adresa 16 bitů).
- BLOB (Binary Large Object) – velikost dána prostředím



Aplikovaná informatika 7

Číselné formáty

- **Celočíselné**
 - Bit (logický datový typ)
 - Byte (1 B, rozsah 0 – 255)
 - Integer (2 B)
 - Long Integer (4 B)
- **Racionální**, podíl dvou celých čísel
- **Neceločíselné**
 - s pevnou řádovou čárkou
 - s pohyblivou řádovou čárkou

Aplikovaná informatika 8

Číselné soustavy

- **Binární** (dvojková)
- **Oktalová** (osmičková, 1 znak = 3 b)
- **Šestnáctková** (hexadecimální, 4 b)
- **Dekadická** (desítková)

postupné dělení

18,625

18 : 2 = 9, Zb. 0

9 : 2 = 4, Zb. 1

4 : 2 = 2, Zb. 0

2 : 2 = 1, Zb. 0

1 : 2 = 0, Zb. 1

postupné násobení

0,625 · 2 = 1,250

0,250 · 2 = 0,500

0,500 · 2 = 1,000

10010,101

Aplikovaná informatika 9

Číselné kódy pro celá čísla

- **Přímý kód**
 - znaménko
 - absolutní hodnota

39 ⇒

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

-39 ⇒

1	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---
- **Inverzní kód**
 - záporné + 2ⁿ - 1

39 ⇒

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

-39 ⇒ 2⁸-1-39 ⇒

1	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---
- **Doplňkový kód**
 - záporné + 2ⁿ
 - záporné ⇒ 2⁸-39

39 ⇒

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

-39 ⇒ 2⁸-39 ⇒

1	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Aplikovaná informatika 10

Čísla s pevnou řádovou čárkou

- Omezená velikost celé části.
- Omezená přesnost (počet desetinných míst).

$$2,625 \Rightarrow \begin{array}{cccccccc} \text{zn.} & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \\ \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} \end{array} = 2,625$$

↑
poloha řádové čárky

$$2,6 \Rightarrow \begin{array}{cccccccc} \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} \end{array} = 2,5625$$

- Datový typ **měna** (Currency) (10,4)

Aplikovaná informatika 11

Čísla s pohyblivou řádovou čárkou

- Tvar $m \cdot 2^e$, kde je m – mantisa $|m| < 1$
 e – celočíselný exponent
- Normování mantisy $0,5 \leq m < 1$

$$2,625 = 0,65625 \cdot 2^{+2} = \begin{array}{cccccccc} \text{zn.} & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} & \text{zn.} & 2^1 & 2^0 \\ \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} \end{array} = 2,5$$

mantisa exponent

- Skrytý bit (výsledek normování)
- Uložení hodnoty 0, exponent + 2^{n-1}

Aplikovaná informatika 12

Zaokrouhlovací chyby

- při výpočtech na počítači neplatí základní matematické zákony (distributivní a asociativní zákon): $(a + b) + c \neq a + (b + c)$

A = 0,1234567
B = 2381,325
A+B = 2381,448

Kahanův trik pro B > A

B = 2381,325
-(A+B) = 2381,448
B - (A+B) = -0,123
A = 0,1234567
(B - (A+B)) + A = 0,0004567

- má kumulativní charakter

Aplikovaná informatika 13

Kumulace chyb

Průběh odchylky při načítání hodnoty 0,1 binárně s přesností 24 binárních míst

změna gradientu ovlivněna zejména změnou exponentu uložení průběžného součtu

Aplikovaná informatika 14

Důsledky kumulace chyb

Ekvivalentní algoritmy

<pre> I = 0 DX = 0,1 X = 0 While Not I=1000 { tělo} X = X + DX I = I + 1 End While </pre>	<pre> DX = 0,1 X = 0 While Not X=100 { tělo} X = X + DX End While </pre>
---	--

Algoritmus proběhne 1000x
Pro X = 0; 0,1; 0,2; ...; 99,9

Algoritmus bude probíhat až do přeplnění X

Aplikovaná informatika 15

Číselné kódy

- **Přímý kód**

$$x_{pr} = \begin{cases} x & x \geq 0 \\ 1-x & x < 0 \end{cases} \quad |x| < 1$$
- **Inverzní kód**

$$x_{inv} = \begin{cases} x & x \geq 0 \\ 2-2^{-n} + x & x < 0 \end{cases} \quad |x| < 1$$
- **Doplňkový kód**

$$x_{dop} = \begin{cases} x & x \geq 0 \\ 2+x & x < 0 \end{cases} \quad -1 \leq x < 1$$



Programové struktury

- **Posloupnost** příkazů (činností)
- **Rozhodování a větvení**
 - podmíněná činnost (if – then)
 - rozhodování (if – else – then)
 - větvení (select case)
- **Opakování**
 - s pevným počtem opakování (for)
 - s testem na začátku (do while)
 - s testem na konci (repeat – until)
- **Programové rutiny**



Programové rutiny

- **Funkce**

```
Function fName (Parameters) as DataType
    function body
    fName = value
End Function
...
variable = fName(Parameters)
```
- **Procedury**

```
Sub sName (Parameters)
    subroutine body
End Sub
...
Call sName (Parameters)
```



Proměnné

- **Globální**
 - existují v celém projektu,
 - přístupné ve všech programových rutinách,
- **Modulů**
 - existují v rámci modulu,
 - přístupné ve všech programových rutinách modulu,
- **Lokální**
 - existují pouze po dobu činnosti procedury,



Předávání dat

- Předání **hodnotou**
Parameters: ByVal variable as DataType
 - proměnná existuje jako lokální
- Předání **odkazem**
Parameters: ByRef variable as DataType
 - předává se reference na globální proměnnou
 - pracuje se přímo s touto proměnnou
- **Lokální proměnné**
 - životnost jen v průběhu činnosti
 - STATIC – zachovává hodnotu pro příští volání
