

Algoritmizace a programování

Typy

- Základní (primitivní) datové typy
- Deklarace

Typy v jazyce Java

- Základní datové typy (primitivní datové typy)
 - Celočíselné – `byte`, `short`, `int`, `long`
 - Reálné – `float`, `double`
 - Znakový – `char`
 - Logický – `boolean`

 - Základní datové typy nejsou třídy
 - Ke každému základnímu primitivnímu datovému typu existuje obalová třída, pomocí které lze hodnotu toho kterého typu převést na objekt a která obsahuje některé nástroje pro práci s hodnotami daného typu

- Referenční datové typy
 - Pole
 - Třídy – `class`
 - Výčet – `enum`
 - Rozhraní – `interface`

Reálné datové typy

Typy

<code>float</code>	4 byte	$\pm 1.4\text{E-}45 \dots \pm 3.4\text{E+}38$	7 – 8 platných číslic
<code>double</code>	8 byte	$\pm 4.9\text{E-}324 \dots \pm 1.7\text{E+}308$	15 – 16 platných číslic

- Hodnoty reálných typů ukládány v paměti v unifikovaném formátu
 $\langle \text{znaménko} \rangle \langle \text{mantisa} \rangle . 10^{\langle \text{exponent} \rangle}$
- Typy vyhovují mezinárodnímu standardu IEEE 754
 - typ `float` 4 byte = 32 bitů, 1 bit znaménko, 8 bitů exponent, 23 bitů mantisa – $\pm 1.mmm \times 2^{eee}$
- Zápis konstantních hodnot reálných typů
 - Znak – znaménko, číslice, oddělovač des. míst, oddělovač exponentu
 - Oddělovač desetinných míst tečka
12.3 -1.847
 - Mohou začínat a končit tečkou
14. .85
 - Oddělovač exponentu – E, e
8e2 5E-3 -4.846E+3
 - Implicitně jsou konstanty typu `double`
 - Konstanty typu `float` končí na F, f
-14.3f 1E-5F

Celočíselné typy

Typy

<code>byte</code>	1B	-128 .. 127
<code>short</code>	2B	-32768 .. 32767
<code>int</code>	4B	-2147483648 .. 2147483647
<code>long</code>	8B	(viz například specifikace jazyka) řád hodnot 10^{18}

- Hodnoty uloženy ve tvaru dvojkového doplňku (viz předmět CIP)
- Jazyk Java nemá neznaménkový celočíselný typ – jistá nevýhoda u jednobytových hodnot – technické aplikace, grafické aplikace

Celočíselné konstanty

- V **desítkové** soustavě
 - Posloupnost číslic 0 . . 9, které nezačíná číslicí 0 (nula)
94, 76, 0, 1
- V **osmičkové** soustavě
 - Posloupnost číslic 0 . . 7, která začíná číslicí 0 (nula)
07, 013
- V **šestnáctkové** soustavě
 - Posloupnost, která začíná znaky 0x nebo 0X následovanými hexadecimálními číslicemi 0 . . 9, a, b, c, d, e, f nebo A, B, C, D, E, F
0xFF, 0X10

Celočíselné konstanty jsou implicitně typu `int`
Konstanta typu `long` ukončená znakem `L`, `l`
9876543210123456L

Logický typ

- Jediným logickým typem jazyka Java je typ `boolean`
- Typ `boolean` je typ, který obsahuje pouze 2 hodnoty
- Tyto hodnoty jsou představované (a v kódu zapisované) logickými konstantami `true` a `false`
 - `true` – logická 1, pravda
 - `false` – logická 0, nepravda
- V jazyce Java nejsou hodnoty logického typu převoditelné na hodnoty jiného datového typu

Znakový typ

- Typ `char`
- Proměnná typu `char` zabírá v paměti 2 byte
- Znaky jsou kódovány čísly – jazyk Java používá kódování Unicode
- Znakové konstanty – uzavřené do apostrofů
 - Jediný **znak v apostrofech** `'A'`
 - **Posloupnost** `'\uXXXX'`, kde `XXXX` jsou šestnáctkové číslice, představující kód znaku, použití například pro zápis akcentovaných znaků `'\u0041'`
 - **Escape sekvence** například `'\n'` pro znak „nová řádka“ (newline, linefeed, LF)
 - **Osmičkový zápis** `'\ooo'`, kde `ooo` jsou osmičkové číslice představující kód čísla, vždy jsou nutné všechny tři číslice `'\101'`
- Typ `char` je kompatibilní s typem `int` (konverze z `char` na `int` je implicitní)

■ Escape sekvence

<code>\b</code>	<code>\u0008</code>	backspace BS
<code>\t</code>	<code>\u0009</code>	horizontal tab HT
<code>\n</code>	<code>\u000a</code>	linefeed LF
<code>\f</code>	<code>\u000c</code>	form feed FF
<code>\r</code>	<code>\u000d</code>	carriage return CR
<code>\"</code>	<code>\u0022</code>	double quote "
<code>\'</code>	<code>\u0027</code>	single quote '
<code>\\</code>	<code>\u005c</code>	backslash \

!!! Konstantní znakové hodnoty je nutné uzavřít v apostrofech

■ Kódy některých akcentovaných znaků

Á	<code>\u00C1</code>	á	<code>\u00E1</code>
Č	<code>\u010C</code>	č	<code>\u010D</code>
Ď	<code>\u010E</code>	ď	<code>\u010F</code>
É	<code>\u00C9</code>	é	<code>\u00E9</code>
Ě	<code>\u011A</code>	ě	<code>\u011B</code>
Í	<code>\u00CD</code>	í	<code>\u00ED</code>
Ň	<code>\u0147</code>	ň	<code>\u0148</code>
Ó	<code>\u00D3</code>	ó	<code>\u00F3</code>
Ř	<code>\u0158</code>	ř	<code>\u0159</code>
Š	<code>\u0160</code>	š	<code>\u0161</code>
Ť	<code>\u0164</code>	ť	<code>\u0165</code>
Ú	<code>\u00DA</code>	ú	<code>\u00FA</code>
Ů	<code>\u016E</code>	ů	<code>\u016F</code>
Ý	<code>\u00DD</code>	ý	<code>\u00FD</code>
Ž	<code>\u017D</code>	ž	<code>\u017E</code>

Řetězcové konstanty

- Konstantní textové řetězce – jedná se o instance třídy `String` (nepatří mezi primitivní datové typy)
- Sekvence konstantních znaků uzavřené do uvozovek
- V jednom jediném textovém řetězci je možné kombinovat různé zápisy znaků
 - `"konstantni textovy retezec"`
 - `"Zadej stranu \u010Dtverce "`
 - `"a\nb\nc\n"`
- Java umožňuje zřetězení textových řetězců pomocí operátoru `+`
 - `"kratky text " + "kratky text je dlouhy text "`
 - `"Hodnota cisla pi je " + 3.14`
 - `"Mocnina dvou je " + 4`
 - `"Logicka 1 je " + true`

Deklarace proměnných

- Deklarace proměnné je příkaz, který přidělí proměnné určitého typu **jméno**, **paměťový prostor** a **počáteční hodnotu**

```
<typ> <jmeno> = <hodnota>;  
<typ> <jmeno>;  
<typ> <jmeno>, <jmeno>, <jmeno>;
```

- Příklady deklarací

```
float soucet = 0;  
int i;  
float a, b, c;
```

- Počáteční hodnota

- Explicitně uvedena deklarací
- Členské proměnné tříd – implicitně inicializovány
- Lokální proměnné metod – nejsou implicitně inicializovány

- Deklarace v kódu mohou být – uvnitř třídy, kdekoli v metodě
- Deklarace před prvním použitím
- Každou proměnnou deklarujte zvlášť, okomentujte význam
- Mezi deklaracemi a ostatními příkazy je vhodné dát prázdnou řádku
- Hodnotu proměnné lze v kódu měnit – viz dále operátory přiřazení, načtení ze `stream`

Deklarace konstant

- Deklarace proměnné s konstantní hodnotou (konečnou, neměnnou)
- Deklaraci předchází klíčové slovo **final**

```
final <typ> <jmeno> = <hodnota>;
```

```
final <typ> <jmeno>;
```

- Příklady použití

```
1. final int MAX = 200;
```

```
2. final int MAX;
```

```
MAX = 200;
```

- Konstantě je hodnota přiřazena právě jednou, hodnotu konstanty nelze poté v kódu dále měnit

Shrnutí

- Primitivní typy jazyka Java
 - Reálné typy
 - Celočíselné typy
 - Znakový typ
 - Logický typ
- Zápis konstantních hodnot (literálů) jednotlivých základních typů
- Textové literály, operace zřetězení
- Deklarace proměnných a konstant

Praktické ukázky

- Deklarace proměnných jednotlivých základních typů
- Deklarace proměnných s přiřazením hodnoty
- Změna hodnoty proměnných

- Deklarace konstanty základních typů
- Změna hodnoty konstanty

- Výpis hodnot pomocí
`System.out.println(<parametr>);`

Kontrolní a rozšiřující otázky

- Jaký je rozdíl mezi proměnnou a konstantou?
- Co je to číselná soustava – dvojková, osmičková, desítková, šestnáctková.
- Co je to kódování znaků?
- Co je to UNICODE?
- Jaká jiná kódování znaků znáte?

Úlohy k procvičování

- Deklarujte v programu proměnné jednotlivých základních typů. V dalším kódu změňte hodnotu těchto proměnných přiřazením konstantní hodnoty, operátor přiřazení =. Sledujte případné chyby.
- Deklarujte v programu konstanty (proměnné s konstantní hodnotou) jednotlivých základních typů, vyzkoušejte různý způsob zápisu konstantních hodnot. V dalším kódu změňte hodnotu těchto konstant. Sledujte zvýraznění chyby a nabízená řešení při editaci, chybové hlášení na výstupu po spuštění.
- Deklarujte v programu proměnné jednotlivých základních typů. Deklaraci zapište ve tvaru s přiřazením. Následně zkuste, jaké konverze provádí jazyk Java implicitně – zkuste postupně vzájemným přiřazením hodnot jednotlivých proměnných. Sledujte zvýraznění chyb, chybová hlášení.
- V programu deklarujte proměnné, konstanty jednotlivých základních typů. Dále proveďte výpis hodnot proměnných, konstant nebo konstantních hodnot pomocí: `System.out.println()`. V okrouhlých závorkách musí být uveden parametr, jehož hodnotu požadujeme vypsát. Zkombinujte výpis hodnoty s popisným textovým řetězcem, popřípadě s výpisem dalších hodnot, použití operátoru +.
- Deklarujte v programu proměnnou bez přiřazení počáteční hodnoty. Hodnoty této proměnné následně použijte – přiřazením do jiné proměnné, výpisem hodnoty. Sledujte zvýraznění chyby při editaci, chybové hlášení po spuštění, nabízená řešení při editaci.
- Otestujte různý způsob zápisu znaků pro různé účely – v konstantní znakové hodnotě, v konstantním textovém řetězci, v kódu programu.