



# Tvorba tried

Informatika 1

Tvorba tried

# TRETIA ČASŤ



# Cieľ cvičenia

- Formátovanie čísiel.
- Tvorba triedy `Formatovac`.

# String.format

- Metóda požaduje dva druhy parametrov:
  - formátovací reťazec – môže obsahovať fixný text a niekoľko **špecifikátorov**.
  - zoznam argumentov formátovacieho reťazca (tých môže byť viac).
- Špecifikátory (číselných a znakových) argumentov majú tvar  
`%[index argumentu$][flag][šírka][.presnosť]konverzia`
  - Vždy začína **%** a nasleduje zoznam nepovinných upresnení.
  - index argumentu** = ktorý argument zo zoznamu argumentov sa má dosadiť do tohto špecifikátora. Ak chcem prvý argument, napíšem `1$`
  - flag** = ako sa má formátovať. Flag-y je potrebné voliť na základe toho, čo sa formátuje (celé číslo, desatinné číslo..).
  - šírka** = minimálna šírka výstupu.
  - .presnosť** = obmedzuje počet znakov (konkrétny význam závisí od toho, aká konverzia sa použije)
  - konverzia** = aká konverzia argumentu sa použije.

# Konverzie

- `%[index argumentu$][flag][šírka][.presnosť]` **konverzia**
- <https://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html>

Druh argumentu	Konverzia		Význam	Príklad	Výstup
znak	c	C	Unicode znak	<code>String.format("%c", 'a');</code>	<code>"a"</code>
celé číslo	d		celé číslo v desiatkovej sústave	<code>String.format("%d", 169);</code>	<code>"789"</code>
	o		celé číslo v osmičkovej sústave	<code>String.format("%o", 169);</code>	<code>"251"</code>
	x	X	celé číslo v šestnástkovej sústave	<code>String.format("%X", 169);</code>	<code>"A9"</code>
desatinné číslo	e	E	desatinné číslo vo vedeckej notácii	<code>String.format("%e", 3.141);</code>	<code>"3,141000e+00"</code>
	f		desatinné číslo	<code>String.format("%f", 3.141);</code>	<code>"3,141000"</code>

# Konverzie

- `%[index argumentu$][flag][šírka][.presnosť]konverzia`

Flag	Použiteľné konvencie				Popis
	Všeobecné	Znakové	Celočíselné	Neceločíselné	
'-'	áno	všetky	všetky	všetky	zarovnanie doľava
'+'	nie	nie	d o x X	všetky	vždy so znamienkom
' '	nie	nie	d o x X	všetky	kladné hodnoty predchádza medzera
'0'	nie	nie	všetky	všetky	hodnota bude zo začiatku doplnená nulami
','	nie	nie	d	e E f g G	podľa nastavenia systému oddelí cifry do skupín
'('	nie	nie	d o x X	e E f g G	záporné hodnoty budú v zátvorke

# Šírka a presnosť

- `%[index argumentu$][flag][šírka][.presnosť]konverzia`
  - Určuje minimálny počet znakov vo výstupe.
- `%[index argumentu$][flag][šírka][.presnosť]konverzia`
  - Pre konverzie e, E a f znamená počet znakov za desatinnou čiarkou.
  - Nepoužívať pri celých číslach!

# Ako by ste zapísali formát..

- Predpokladajme, že máme dve premenné:
  - `int` `n` = 123
  - `double` `d` = -3.14159
- 1. ..číslo `n` v desiatkovej sústave?
- 2. ..číslo `n` v šestnástkovej sústave, čísla `a–f` malým písmenom.
- 3. ..číslo `n` tak, aby zaberalo 10 znakov, bolo zarovnané doľava, malo zobrazené znamienko?
- 4. ..číslo `d`
- 5. ..číslo `d` vo vedeckej notácii s presnosťou na 2 desatinné miesta?
- 6. ..číslo `d` s presnosťou na 2 desatinné miesta tak, aby zaberalo 15 znakov, zápornosť bola indikovaná uzavretím do `()` a bolo zarovnané doľava?
- 7. ..číslo `d`, aby sa vypísalo 2x za sebou, raz podľa špecifikácie 5 a raz podľa špecifikácie 6?



# Konštanta

- Syntax:

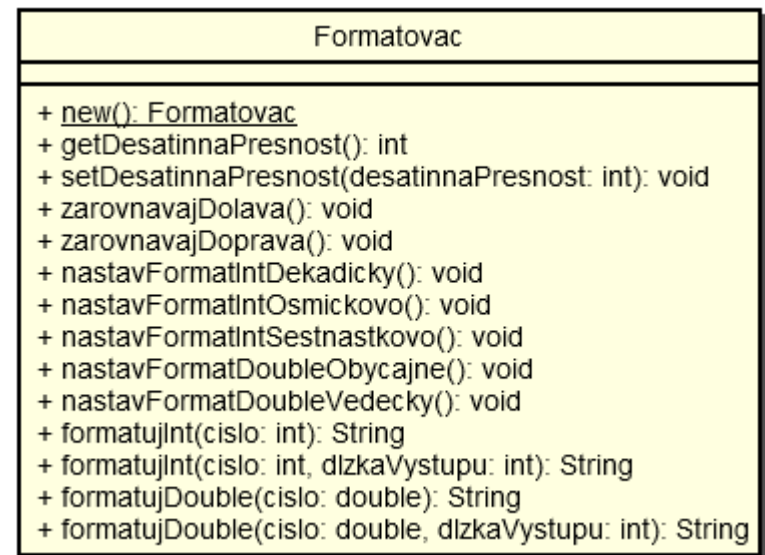
```
public static final typ_premennej názov = hodnota;
```

- **public** = je dostupná z vonku.
- **static** = patrí triede, je definovaná v rámci {tela} triedy, nie metódy ani konštruktora.
- **final** = keď sa jej raz priradí hodnota, už sa nedá zmeniť.
- Príklad: `Matematika.PI`

```
public class Matematika {  
    public static final double PI = 3.14159;  
}
```

# Projekt format

- Vytvorte si projekt **format**.
- Preskúmajte vonkajší pohľad na triedu `Formatovac` v UML diagrame tried :
  - Aká je úloha triedy `Formatovac`?
  - Aké bude mať atribúty?
- Implementujte triedu `Formatovac` podľa danej špecifikácie.



Tvorba tried

# ŠTVRTÁ ČASŤ



# Cieľ cvičenia

- Vytvorenie UML diagramu triedy na základe slovného zadania.
- Implementácia triedy podľa UML diagramu.

# Komplexné číslo

- Vytvorte triedu komplexné číslo, ktoré bude umožňovať:
  - Vytvorenie čísla.
  - Zistenie a nastavenie reálnej a imaginárnej zložky.
  - Vrátiť svoj reťazec v algebraickom a geometrickom tvare.
  - Bude sa vedieť pričítať, odčítať, vynásobiť a vydeliť iným komplexným číslom (parametre metód budú v tvare reálnej a imaginárnej zložky).
  - Vrátiť svoju absolútnu hodnotu

# Komplexné číslo

## Operácie s komplexnými číslami

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

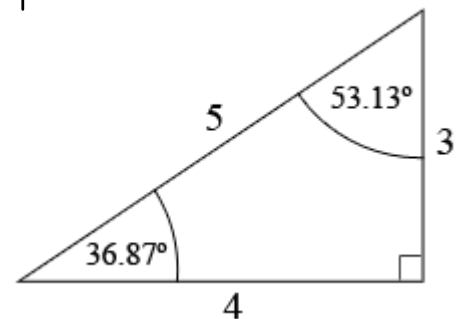
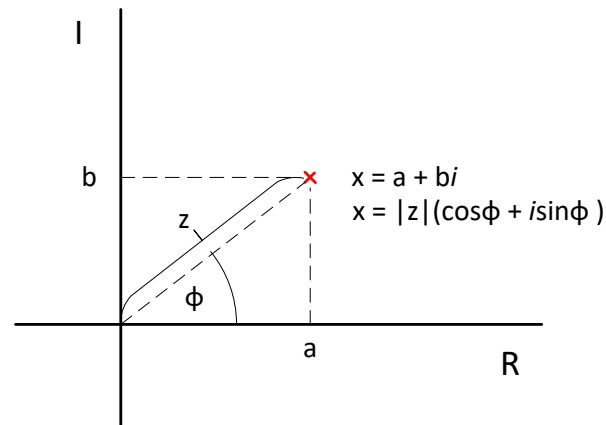
$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

$$(a + bi) \times (c + di) = (ac - bd) + (bc + ad)i$$

$$\frac{(a + bi)}{(c + di)} = \left( \frac{ac + bd}{c^2 + d^2} \right) + \left( \frac{bc - ad}{c^2 + d^2} \right)i$$

$$|(a + bi)| = \sqrt{a^2 + b^2}$$

## Geometrický tvar komplexného čísla



# Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5BI137 Informatika 1 na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.  
Katedra informatiky  
Fakulta riadenia a informatiky  
Žilinská univerzita v Žiline  
[Michal.Varga@fri.uniza.sk](mailto:Michal.Varga@fri.uniza.sk)

