



### Základné metódy objektov triedy String:

**int** length()

- o vráti dĺžku reťazca

**char** charAt(**int** index)

- o vráti znak na zadanom indexe v reťazci (znaky sú indexované od 0)

**boolean** equals(String r)

- o vráti *true* práve vtedy, keď tento reťazec sa skladá z tej istej postupnosti znakov ako reťazec referencovaný parametrom r

String trim()

- o vráti referenciu na novovytvorený reťazec vytvorený odstránením počiatočných a koncových medzier

String toLowerCase() resp. String toUpperCase()

- o vráti referenciu na novovytvorený reťazec po zmene znakov v reťazci na malé (veľké) písmena

String substring(**int** zacIndex, **int** konIndex)

- o vráti referenciu na novovytvorený reťazec obsahujúci podreťazec tvorený znakmi na indexoch zacIndex (vrátane) až konIndex (nie je zahrnutý)

**int** indexOf(String podreťazec) resp. **int** indexOf(**char** znak)

- o vráti index prvého výskytu podreťazca resp. znaku v reťazci. Ak sa v reťazci nenachádza vráti -1

### Základné metódy objektov triedy Turtle:

**void** center()

- o presunie korytnačku do stredu plochy, v ktorej sa nachádza (korytačka musí byť v ploche)

**void** setPosition(**double** x, **double** y)

- o presunie korytnačku na pozíciu so súradnicami [x, y], čiara sa nekreslí

**void** step(**double** dlzka)

- o spraví krok v smere natočenia zadanej dĺžky, čiara sa kreslí v závislosti od stavu kresliaceho pera

**void** turn(**double** uhol)

- o otočí korytnačku o zadaný uhol v smere hodinových ručičiek

**void** moveTo(**double** x, **double** y)

- o korytačka spraví krok do bodu na súradniciach [x, y], čiara v závislosti od kresliaceho pera

**void** setDirection(**double** smer)

- o natočí korytnačku zadaným smerom (smer 0 je nahor, 90 doprava, atď.)

**double** getDirection()

- o vráti smer aktuálneho natočenia korytnačky

**void** turnTowards(**double** x, **double** y)

- o natočí korytnačku tak, aby bola natočená smerom k bodu na súradniciach [x, y]

**double** directionTowards(**double** x, **double** y)

- o vráti smer, pri ktorom by bola korytnačka natočená smerom k bodu na súradniciach [x, y]

**double** distanceTo(**double** x, **double** y)

- o vráti vzdialenosť korytnačky k bodu na súradniciach [x, y]

**void** dot(**double** polomer)

- o nakreslí vyplnený kruh (farbou výplne) so zadaným polomerom a stredom v pozícii korytnačky

**void** setFillColor(Color farba)

- o nastaví farbu výplne

**void** setPenColor(Color farba)

- o nastaví farbu kresliaceho pera

**void** penDown() resp. **void** penUp()

- o zapne resp. vypne kresliace pero

## Základné metódy objektov triedy WinPane (kresliaca plocha):

**void** add(Turtle korytnacka)

- o pridá (referencovanú) korytnacku do kresliacej plochy

**void** remove(Turtle korytnacka)

- o odoberie (referencovanú) korytnacku z kresliacej plochy

**int** getWidth() resp. **int** getHeight()

- o vráti šírku, resp. výšku kresliacej plochy

## Java a polia

- o prechod všetkými indexami poľa referencovaného z premennej *pole*:

```
for (int i=0; i<pole.length; i++) { ... }
```

## JPAZ a myšacie udalosti

```
protected void onMouseClicked(int x, int y, MouseEvent detail) {  
    if ((detail.getButton() == MouseEvent.BUTTON1) &&  
        detail.isControlDown()) {  
        // pri zatlačení ľavého tlačidla myši  
        // vo chvíli, keď je zatlačený aj Ctrl  
    }  
}
```

## Farby

Color.red, Color.blue, Color.green, Color.gray, Color.black ... alebo  
**new** Color(**int** r, **int** g, **int** b), kde r, g a b sú celé čísla od 0 po 255.

## Náhodné číslo

Vygenerovanie náhodného čísla z intervalu <0, a): Math.random()\*a

Vygenerovanie náhodného celého čísla od 0 po n: (**int**)(Math.random()\*(n+1))

## Vytvorenie poľa

Vytvorenie poľa 6 celých čísel:

```
int[] pole = new int[6];
```

Vytvorenie poľa 6 celých čísel s inicializáciou hodnôt:

```
int[] pole = {3, 4, 6, 1, 2, 4};
```

Výpis poľa: System.out.println(Arrays.toString(pole));

Kopírovanie prvkov poľa:

```
System.arraycopy(odkiaľ, odAkéhoIndexu, kam, odAkéhoIndexu, koľkoPolíčok);
```

## Čísla

Double.MAX\_VALUE - najväčšie číslo, ktoré možno uložiť v premennej typu double

Double.POSITIVE\_INFINITY - +∞

**double** cislo = Double.parseDouble("3.14"); - prevedie reťazec na číslo

Math.sqrt(c) - vyráta odmocninu zadaného čísla c

## Znaky

Character.isUpperCase(z) - vráti, či znak z predstavuje veľké písmeno

