



Polsemestrálny test

Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmyšľaj), raz rež (programuj)

Pravidlá a informácie:

- o čas na riešenie úloh je **80 minút**, resp. do 12:15,
- o nie je dovolená žiadna (elektronická aj neelektronická) komunikácia s kýmkoľvek okrem dozoru,
- o nie je dovolené používať žiadne zdroje ani materiály okrem oficiálneho ťaháku,
- o nie je dovolené používať žiadnu inú aplikáciu než Eclipse (s výnimkou webového prehliadača pri odosielaní riešenia), monitorovací softvér musí byť spustený počas celého testu,
- o porušenie pravidiel má za následok hodnotenie FX,
- o svoje riešenia odovzdávajte cez systém Moodle (<http://lms.ics.upjs.sk/>).

Upozornenie:

- o Skontrolujte si, či máte k projektu pripojenú knižnicu `jpez2.jar`.

Ktoré úlohy treba riešiť:

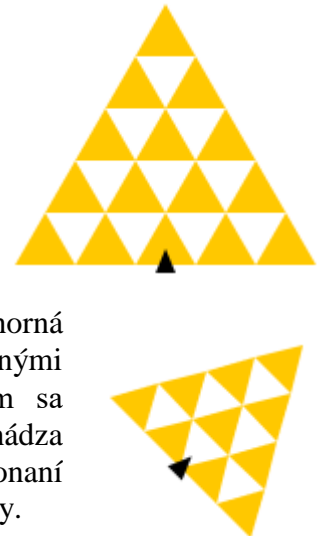
V **Časti 1** je cieľom úloh vytvoriť triedu `Midtermarka`, ktorá rozširuje triedu `Turtle`. Z prvej trojice úloh si **vyberte len 2 úlohy**, ktoré **budete riešiť!!!** To, ktoré úlohy ste sa rozhodli riešiť, uveďte v komentári pri odosielaní riešenia cez Moodle (ak to nie je zrejmé z odoslaného).

V **Časti 2** je len jedna úloha, t.j. v tejto časti nie je možný výber úloh.

Časť 1 (dve úlohy z troch)

Pyramída (10 bodov)

Do triedy `Midtermarka` pridajte metódu `pyramida`, ktorá nakreslí pyramídu ako na obrázku vpravo. Metóda má dva parametre `strana` a `poschodia`. Parameter `strana` určuje dĺžku každej zo strán pyramídy, parameter `poschodia` určuje počet poschodí pyramídy (na obrázku vpravo má horná pyramída 5 poschodí, dolná 4 poschodia). Pyramída je tvorená vyplnenými rovnostrannými trojuholníkmi s dĺžkou strany `strana/poschodia`, pričom sa používa aktuálne nastavená farba výplne. Korytnačka sa na začiatku nachádza v strede jednej zo strán pyramídy a natočená je v smere výšky pyramídy. Po vykonaní metódy nech je korytnačka na pozícii a v smere, ako bola pred vykonaním metódy.



```
public void pyramida(double strana, int poschodia)
```

Rady:

- Odporúčame si vytvoriť metódu, ktorá nakreslí vyplnený trojuholník so zadanou dĺžkou strany a s počiatočnou a koncovou pozíciou ako je naznačené na obrázku.

```
public void vyplnenyTrojuholnik(double strana)
```

- Odporúčame si vytvoriť metódu, ktorá nakreslí jedno poschodie pyramídy skladajúce sa z `n` trojuholníkov so zadanou dĺžkou strany, pričom odporúčaná počiatočná a koncová pozícia sú naznačené na obrázku.

```
public void poschodie(double strana, int n)
```



Počet nul (10 bodov)

Do triedy `Midtermarka` pridajte metódu `pocetNul`, ktorá pre zadané prirodzené číslo c ($c \geq 1$) vráti koľkými nulami končí toto číslo. Uvažujeme zápis čísel v desiatkovej sústave.

```
public int pocetNul(long c)
```

Príklady:

```
pocetNul(10300) = 2
```

```
pocetNul(13000) = 3
```

Hodnotenie: Maximálne 3 body za riešenie, ktoré využije niektorú z metód na prevod čísla na jeho zápis v desiatkovej sústave vo forme postupnosti znakov.

Počet skratiek (10 bodov)

Informatika je plná troj- a viacpísmenových skratiek (OOP, TCP, MQTT, SOAP, XSLT, RMI, CORBA, ...) Do triedy `Midtermarka` pridajte metódu `pocetSkratiek`. Táto metóda dostane ako parameter **nonnull**ovú referenciu na reťazec (objekt triedy `String`) a vráti počet informatických skratiek v tomto reťazci. Informatickou skratkou nazývame každý súvislý podreťazec tvorený aspoň troma veľkými písmenami anglickej abecedy.

Príklady:

```
pocetSkratiek("Čo vieš o TCP a UDP?") = 2
```

```
pocetSkratiek("Neskôr boli CORBA a RMI vytlačené SOAPom.") = 3
```

```
public int pocetSkratiek(String r)
```

Časť 2

Čas do príchodu (10 bodov)

- (2 body) Vytvorte triedu `MidtermPane`, ktorá rozširuje triedu `WinPane`. Po vytvorení kresliacej plochy triedy `MidtermPane` nech sa v nej (automaticky v konštruktore, resp. „inicializačnej metóde“) vytvorí 10 korytnáčiek triedy `Turtle` na náhodných pozíciách a s náhodným natočením vo viditeľnej časti kresliacej plochy.
- (8 bodov) Korytnačka sa dokáže otáčať rýchlosťou 1° za sekundu. Rýchlosť pohybu korytnačky je jednotková vzdialenosť („1 pixel“) za sekundu. Korytnačka sa nedokáže v jednom okamihu súčasne posúvať aj otáčať. Nech C je bod na súradniciach $[x, y]$. Za aký najkratší čas v sekundách sa do bodu C môže dostať prvá korytnačka? Nezapudnite, že korytnačka sa môže otáčať v smere aj proti smeru hodinových ručičiek.

Metóda `casDoPrichodu` má 2 parametre x, y - súradnice bodu C , a vráti najmenší čas v sekundách, za aký sa niektorá z korytnáčiek vie dostať do bodu C .

```
public double casDoPrichodu(double x, double y)
```

Rada (matematika ZŠ): Vonkajší uhol k uhlu α má veľkosť $360^\circ - \alpha$.

Na druhej strane nájdete oficiálny ťahák.



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

