



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://moodle.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).

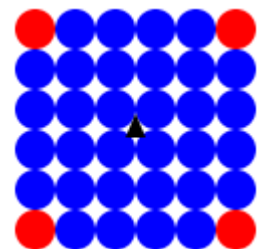
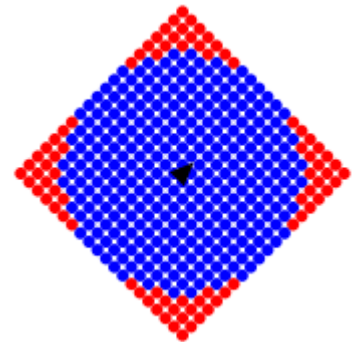
V **Časti 2** je len jedna úloha, t.j. v tejto časti nie je možný výber úloh.

Časť 1 (dve úlohy z troch)

Bodkovnica (10 bodov)

Do triedy `Midtermarka` pridajte metódu `bodkovnica`, ktorá nakreslí „štvorec“ vytvorený zo zadaného počtu farebných bodiek. Metóda má dva parametre `rozmer` a `polomer`. Prvý parameter `rozmer` určuje počet bodiek, ktoré sa nachádzajú v jednom rade. Zároveň je to aj počet radov „bodkového“ štvorca. Druhý parameter `polomer` určuje polomer každej z bodiek. Farby jednotlivých bodiek sú červená alebo modrá tak, ako je to znázornené na obrázku. Modrou farbou sú nakreslené práve tie bodky, ktorých stred je vzdialený od stredu „bodkového“ štvorca menej ako polovicu dĺžky strany tohto „štvorca“.

Korytnačka sa na začiatku nachádza v strede kresleného „bodkového“ štvorca a je natočená v smere jedného zo stĺpcov. Po nakreslení „bodkového“ štvorca nech je korytnačka na pozícii a je natočená v smere, ako bola pri volaní metódy.



```
public void bodkovnica(int rozmer, double polomer)
```

Najväčší mocninový deliteľ (10 bodov – 8 + 2 body)

Do triedy `Midtermarka` pridajte metódu `nmd`, ktorá pre zadané kladné nenulové číslo n vráti najväčšie také číslo c , že $c^2 | n$ (c^2 delí n bezo zvyšku).

```
public int nmd(int n)
```

Hodnotenie: 2 body sa udeľujú podľa efektívnosti riešenia.

Odbodkovanie reťazcov (10 bodov)

Do triedy `Midtermarka` pridajte metódu `odbodkuj`. Táto metóda dostane ako parametre referencie na dva reťazce a vráti referenciu na novovytvorený reťazec, v ktorom je každý výskyt bodky v prvom reťazci (parameter `r1`) nahradený postupne písmenami z druhého reťazca (parameter `r2`). Ak je v prvom reťazci viac bodiek ako znakov v druhom reťazci, postupnosť nahradzovaných znakov z druhého reťazca cyklicky opakujeme. Môžete predpokladať, že ak prvý reťazec obsahuje aspoň jednu bodku, tak druhý reťazec je neprázdny.

Príklady:

```
odbodkuj("Pro....ovanie", "graming") = "Programovanie"  
odbodkuj(".ni...zita", "Uver") = "Univerzita"  
odbodkuj("M..t.r.", "ideme") = "Midterm"  
odbodkuj("M.n.p.st", "o") = "Monopost"  
odbodkuj("....", "ma") = "mama"
```

```
public String odbodkuj(String r1, String r2)
```

Časť 2

Počty v smere (10 bodov)

- (3 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 22 korytnáčiek triedy `Turtle` na náhodných pozíciách vo viditeľnej časti kresliacej plochy. Každá korytnačka nech je náhodne natočená v smere 0° , 90° , 180° alebo 270° .
- (7 bodov) Do triedy `MidtermPane` pridajte metódu `poctyVSmeroch`, ktorá vráti, koľko korytnáčiek v jednotlivých smeroch natočenia sa nachádza v zadanom kruhu.

Metóda ma 3 parametre špecifikujúce kruh (súradnice stredu `x`, `y` a polomer `r`). Zaujímať nás budú len počty korytnáčiek ležiacich v tomto kruhu. Výsledkom metódy je referencia na novovytvorené pole celých čísel dĺžky 4, ktoré bude obsahovať postupne 4 čísla. Na indexe 0 bude počet korytnáčiek v kruhu natočených v smere 0° , na indexe 1 bude počet korytnáčiek v smere 90° , na indexe 2 bude počet korytnáčiek v smere 180° a na indexe 3 bude počet korytnáčiek v smere 270° .

```
public int[] poctyVSmeroch(double x, double y, double r)
```

Na druhej strane nájdete oficiálny ťahák.



Oficiálny t'ahák

Polsemestrálny test



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmyšľaj), raz rež (programuj)

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 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

void setPenWidth(**double** hrubka)

- o nastaví hrúbku kresliaceho pera

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

