Polsemestrálny test
Zadanie

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Pravidlá a informácie:

- čas na riešenie úloh je **80 minút**, resp. do 11:20,
- nie je dovolená žiadna (elektronická aj neelektronická) komunikácia s kýmkoľvek okrem dozoru,
- nie je dovolené používať žiadne zdroje ani materiály okrem oficiálneho ťaháku,
- nie je dovolené používať žiadnu inú aplikáciu než Eclipse (s výnimkou webového prehliadača pri odosielaní riešenia), monitorovací softvér musí byť spustený počas celého testu,
- porušenie pravidiel má za následok hodnotenie FX,
- svoje riešenia odovzdávajte cez systém Moodle (http://lms.ics.upjs.sk/).

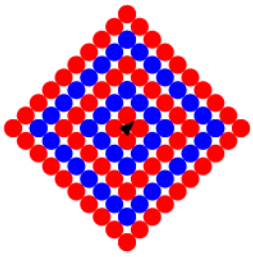
Ktoré úlohy treba riešiť:

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).

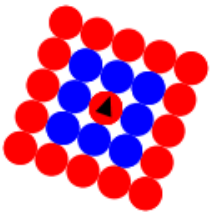
V ***Časti 2*** je len jedna úloha, t.j. v tejto časti nie je možný výber úloh.

Časť 1 (dve úlohy z troch)

Bodkovnica (10 bodov)



Do triedy `Midtermarka` pridajte metódu `bodkovnica`, ktorá nakreslí bodkový štvorec ako na obrázku vpravo. Metóda má dva parametre `n` a `r`. Parameter `n` určuje počet bodiek v jednom riadku, resp. stĺpci bodkového štvorca. Parameter `r` určuje polomer bodiek tvoriacich štvorec. Farby bodiek sa po obvodových „vrstvách“ štvorca postupne striedajú – farba je buď červená alebo modrá (to, ako farbou začnete, je na vašom rozhodnutí). Korytnačka sa na začiatku nachádza v strede štvorca a natočená je rovnobežne s dvoma stranami štvorca. Po vykonaní metódy nech je korytnačka na pozícii a v smere, ako bola pred vykonaním metódy.



```
public void bodkovnica(int n, double r)
```

Rada:

- Odporúčame si vytvoriť pomocnú metódu, ktorá nakreslí jednu obvodovú „vrstvu“ štvorca s farbami bodiek podľa aktuálneho nastavenia. Parameter `n` nech určuje počet bodiek na jednej strane kreslenej „obvodovej“ vrstvy:

```
public void vrstva(int n, double r)
```

Inverzný faktoriál (10 bodov)

Do triedy `Midtermarka` pridajte metódu `inverznyFaktorial`, ktorá pre zadané nezáporné celé číslo n vráti najmenšie také celé číslo k, že $n \leq k!$

```
public int inverznyFaktorial(long n)
```

Boldovač (10 bodov)

Rôzne wiki, či četovacie aplikácie používajú špeciálne značky na formátovanie textov. Napríklad znak * sa zvykne používať na zvýraznenie textu.

Do triedy `Midtermarka` pridajte metódu `bolduj`. Táto metóda dostane ako parameter ne**null**ovú referenciu na reťazec (objekt triedy `String`) a vráti referenciu na novovytvorený reťazec (objekt triedy `String`), ktorý vznikne zo zadaného reťazca zmenou všetkých znakov medzi každou dvojicou za sebou nasledujúcich * (hviezdičiek) na veľké písmena. Zároveň sa hviezdičky ako formátovacie značky odstránia. Predpokladajte korektný vstup, t.j. párny (alebo aj nulový) počet znakov *.

Príklady:

```
bolduj("Dnes bude *midterm* o 9:50") = "Dnes bude MIDTERM o 9:50"
bolduj("*To*to je div*ny te*xt.") = "TOto je divNY TExt."
```

```
public String bolduj(String r)
```

Časť 2

Korytnačie hlasovanie (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 10 korytnačiek triedy `Turtle` na náhodných pozíciách vo viditeľnej časti kresliacej plochy a s náhodným natočením (smerom), ktorý je celočíselným násobkom čísla 15.

- *(7 bodov)* Korytnačka hlasuje „Áno“, ak je natočená smerom nahor, zdržiava sa hlasovania, ak je natočená vodorovne (jej smer je 90 alebo 270), a hlasuje „Nie“, ak je natočená smerom nadol.

  Metóda `najvyssieAno` vráti y-ovú súradnicu najvyššie umiestnenej korytnačky, ktorá hlasovala „Áno“. Ak takej niet, metóda nech vráti `Double.NaN`.

  ```
  public double najvyssieAno()
  ```

Na zadnej strane nájdete oficiálny ťahák.

Oficiálny ťahák
Polsemestrálny test

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Základné metódy objektov triedy String:

`int length()`
- vráti dĺžku reťazca

`char charAt(int index)`
- vráti znak na zadanom indexe v reťazci (znaky sú indexované od 0)

`boolean equals(String r)`
- vráti *true* práve vtedy, keď tento reťazec sa skladá z tej istej postupnosti znakov ako reťazec referencovaný parametrom `r`

`String trim()`
- vráti referenciu na novovytvorený reťazec vytvorený odstránením počiatočných a koncových medzier

`String toLowerCase()` resp. `String toUpperCase()`
- vráti referenciu na novovytvorený reťazec po zmene znakov v reťazci na malé (veľké) písmena

`String substring(int zacIndex, int konIndex)`
- vráti referenciu na novovytvorený reťazec obsahujúci podreťazec tvorený znakmi na indexoch `zacIndex` (vrátane) až `konIndex` (nie je zahrnutý)

`int indexOf(String podretazec)` resp. `int indexOf(char znak)`
- vráti index prvého výskytu podreťazca resp. znaku v reťazci. Ak sa v reťazci nenachádza vráti -1

Základné metódy objektov triedy Turtle:

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

`void setPosition(double x, double y)`
- presunie korytnačku na pozíciu so súradnicami [x, y], čiara sa nekreslí

`void step(double dlzka)`
- spraví krok v smere natočenia zadanej dĺžky, čiara sa kreslí v závislosti od stavu kresliaceho pera

`void turn(double uhol)`
- otočí korytnačku o zadaný uhol v smere hodinových ručičiek

`void moveTo(double x, double y)`
- korytačka spraví krok do bodu na súradniciach [x, y], čiara v závislosti od kresliaceho pera

`void setDirection(double smer)`
- natočí korytnačku zadaným smerom (smer 0 je nahor, 90 doprava, atď.)

`double getDirection()`
- vráti smer aktuálneho natočenia korytnačky

`void turnTowards(double x, double y)`
- natočí korytnačku tak, aby bola natočená smerom k bodu na súradniciach [x, y]

`double directionTowards(double x, double y)`
- vráti smer, pri ktorom by bola korytnačka natočená smerom k bodu na súradniciach [x, y]

`double distanceTo(double x, double y)`
- vráti vzdialenosť korytnačky k bodu na súradniciach [x, y]

`void dot(double polomer)`
- nakreslí vyplnený kruh (farbou výplne) so zadaným polomerom a stredom v pozícii korytnačky

`void setFillColor(Color farba)`
- nastaví farbu výplne

`void setPenColor(Color farba)`
- nastaví farbu kresliaceho pera

`void penDown()` resp. `void penUp()`
- zapne resp. vypne kresliace pero

Základné metódy objektov triedy WinPane (kresliaca plocha):

`void add(Turtle korytnacka)`
- pridá (referencovanú) korytnačku do kresliacej plochy

`void remove(Turtle korytnacka)`
- odoberie (referencovanú) korytnačku z kresliacej plochy

`int getWidth()` resp. `int getHeight()`
- vráti šírku, resp. výšku kresliacej plochy

Java a polia

- 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
`Character.toUpperCase(z)` – vráti znak, ktorý vznikne zo `z` zmenou na veľké písmeno