Polsemestrálny test
Zadanie

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Pravidlá a informácie:

- čas na riešenie úloh je **90 minút**, resp. do 14:05,
- 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,
- svoje riešenia odovzdávajte cez systém Moodle (http://moodle.ics.upjs.sk/).

Upozornenie:

- Skontrolujte si, či máte k projektu pripojenú knižnicu `jpaz2.jar`.

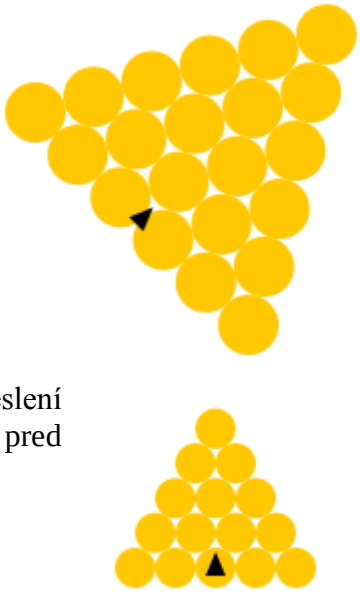
Ktoré úlohy treba riešiť:

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 z kruhov (viď ilustračný obrázok). Metóda má dva parametre `pocetRadov` a `polomer`. Prvý parameter `pocetRadov` určuje počet radov, z ktorých sa pyramída skladá (je to zároveň počet kruhov v najširšom rade). Druhý parameter `polomer` určuje polomer kruhov, z ktorých sa pyramída skladá. Farba kruhov je aktuálne nastavená farba výplne. Je dôležité, aby sa všetky „susediace“ kruhy v pyramíde presne dotýkali. Korytnačka sa na začiatku nachádza v strede najširšieho radu a je natočená v smere výšky pyramídy. Po nakreslení pyramídy nech je korytnačka na pozícii a je natočená v smere, ako bola pred volaním metódy.

```
public void pyramida(int pocetRadov, double polomer)
```

Rada:

- Odporúčame vytvoriť si pomocnú metódu na nakreslenie jedného radu s korytnačkou v strede radu:

```
public void rad(int pocetKruhov, double polomer)
```

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)
```

Pomoc pre hendikepovaných (10 bodov)

Rôzne poruchy nervového a svalového systému výrazne zhoršujú používanie počítačov - zvyčajnými vstupmi sú totiž klávesnica alebo myš. Dôsledkom takýchto porúch (napr. svalový tras a chvenie) môže byť akási obdoba koktavosti. Používateľ vtedy namiesto slova *Java* napíše *Jaaaavvaa*. Naučte korytnačky triedy `Midtermarka` metódu `odstranDuplicitu`, ktorá vráti referenciu na novovytvorený objekt triedy `String`, ktorého obsah vznikne odstránením duplicitných výskytov rovnakého znaku v rade za sebou.

```
public String odstranDuplicitu(String r)
```

Príklady:

- `odstranDuplicitu("Jaaavvaaa") = "Java"`
- `odstranDuplicitu("IIIssiel") = "Isiel"`
- `odstranDuplicitu("IIissiel") = "Iisiel"`
- `odstranDuplicitu("Odddstran duuuupllicitu") = "Odstran duplicitu"`

Časť 2

Bombastické korytnačky (10 bodov)

- *(3 body)* Vytvorte triedu `MidtermPane`, ktorá rozširuje triedu `WinPane`. Po vytvorení kresliacej plochy triedy `MidtermPane` nech sa v nej vytvorí 13 korytnačiek triedy `Turtle` na náhodných pozíciách vo viditeľnej časti kresliacej plochy.

- *(7 bodov)* Keďže korytnačky v JPAZe sú nezničiteľné, ministerstvo obrany za rozhodlo využiť korytnačky v kresliacej ploche na testovanie účinkov explózií. Do kresliacej plochy umiestnili nálož so zadanou silou (parameter `sila`) na pozíciu `[x, y]`. Pri explózii nálože vznikne tlaková vlna, ktorá odhodí korytnačky v smere od miesta explózie. Ak uvážime zjednodušený fyzikálny model, korytnačka vzdialená d od miesta explózie so silou S je odhodená na vzdialenosť S^2/d^4. Budeme predpokladať, že žiadna korytnačka sa nenachádza v mieste explózie. Do triedy `MidtermPane` pridajte metódu `explozia`, ktorá odsimuluje účinky explózie nálože so zadanou silou a umiestnenej na zadanej pozícii. T.j. všetky korytnačky v ploche sa presunú (bez kreslenia trajektórie pohybu) na miesto, kam ich odhodí tlaková vlna, a budú nasmerované v smere od miesta explózie. Metóda vráti najväčšiu vzdialenosť, na akú bola odhodená nejaká z korytnačiek v ploche.

```
public double explozia(double x, double y, double sila)
```

Na druhej 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 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

`void setPenWidth(double hrubka)`
- nastaví hrúbku kresliaceho pera

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.POSSITIVE_INFINITY` – +∞
`double cislo = Double.parseDouble("3.14");` - prevedie reťazec na číslo