2. feladatsor

Határidő Nincs megadva határidő **Pont** 0 **Kérdések** 1 **Időkorlát** Nincs
Engedélyezett próbálkozások Korlátlan

Kvíz kitöltése újra

Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	1. próbálkozás	151,638 perc	0 az összesen elérhető 0 pontból *

* Néhány kérdés még nem lett értékelve

Beadva ekkor: máj 31, 22:45

incs megválaszolva **érdés** **Még nincs értékelve / 0 pont**

Programozási nyelvek (BSc, 18) Java 2. feladatsor

1. feladat

Készítsen egy `Point` osztályt `double` típusú `x` és `y` mezőkkel. Írja meg a `move(dx,dy)` műveletet, mellyel egy pontot el lehet tolni `dx` és `dy` koordinátákkal, valamint a `mirror(cx,cy)` műveletet, mely egy `cx` és `cy` koordinátájú pontra való tükrözést valósít meg.

Készítsen `PointMain` néven Java programot, amelyben bemutatja a `Point` osztály használatát.

2. feladat

Módosítsa a `Point` osztályban a `mirror(p)` műveletet úgy, hogy paramétere (a tükrözési középpont) egy `Point` objektum legyen!

Írjon `distance(p)` műveletet is, mely kiszámolja az adott pont távolságát egy paraméterként kapott `p` ponttól. Használja a `Math.sqrt(...)` függvényt és a Pitagorasz tételt!

Frissítse a `PointMain` osztályt az új műveletekkel!

3. feladat

Javasoljuk ezt a feladatot önállóan megoldani.

Valósítsa meg a `Complex` osztályt `double` típusú valós és képzetes résszel! Írjon `abs()` metódust, amely kiszámolja a komplex szám abszolút értékét. Valósítsa meg az `add(c)`, a `sub(c)` és a `mul(c)` műveleteket oly módon, hogy az `add` adja hozzá a komplex számhoz a paraméterként kapott `c` komplex számot, a `sub` vonja ki belőle, a `mul` pedig szorozza hozzá.

```
alpha.re = 3
alpha.im = 2
beta.re = 1
beta.im = 2
alpha.add(beta)
// alpha.re == 4 && alpha.im == 4 && beta.re == 1 && beta.im == 2
```

4. feladat

Készítsen `Circle` néven kört reprezentáló osztályt. Egy körnek van középpontja (`x` és `y` nevű, `double` típusú adattag) és sugara (`radius`).

Írjon `enlarge(f)` metódust, amellyel a kör sugarát `f`-szeresére változtatja, illetve `getArea()` metódust, amely megadja a kör területét. Használjuk a `Math.PI` értéket!

5. feladat

Készítsen egy `Distance` programot. Ez a parancscsori paramétereket pontoknak értelmezi: a pontok szóközzel elválasztva vannak felsorolva, minden pontnál elöl az `x`, utána az `y` koordináta (ezek is szóközzel elválasztva).

Feltételezhetjük, hogy páros számú paraméter van, amelyek mind egész számok.

A program a `Point` osztály felhasználásával számítsa ki és adja össze az egymás mellett lévő pontok távolságát (pl. 3 pont esetén az 1. és a 2. pont távolságához hozzá kell adni a 2. és a 3. pont távolságát), majd az eredményt írja ki.

Példák:

```
> java Distance
0.0
> java Distance 1 2
0.0
> java Distance 0 0 3 4
5.0
> java Distance 1 2 4 6
5.0
> java Distance 1 2 4 6 7 6
8.0
```

1. gyakorló feladat

Készítsük el a Complex osztályba a `conjugate()` műveletet, mely a komplex számot átalakítja a komplex konjugáltjára. Készítsük el a `reciprocate()` metódust, mely a komplex számot reciprokára alakítja. Definiáljuk a `div(c)` művelet, mely elosztja a komplex számot a paraméterként kapott `c` komplex számmal.

2. gyakorló feladat

Készítse el a `Line` osztályt, mellyel egy adott sík egyeneseit reprezentálhatjuk. Egy egyenest az $ax + by = c$ összefüggés ír le, ahol `a`, `b` és `c` számok `double` típusúak. (Ezek lesznek az osztály adattagjai.)

Írjon az osztályba egy `contains(p)` műveletet, mely eldönti, hogy egy `p` pont rajta van-e az egyenesen!

Írjon egy `isParallelWith(l)` és egy `isOrthogonalTo(l)` metódust, melyek eldöntik, hogy az egyenes párhuzamos-e a paraméterként kapott `l` egyenessel, illetve merőleges-e rá!

3. gyakorló feladat

Készítse el a `Segment` osztályt, mely egy szakaszt reprezentál. A szakasz objektumok ábrázolásához a két végpont koordinátáit tároljuk el. Az adattagok `x1`, `y1`, `x2`, `y2` legyenek, mind `double` típusú.

Írjon az osztályba egy `line()` metódust, mely visszaad egy olyan `Line` objektumot, amely a szakaszra illeszkedő egyenest reprezentál.

Írjon az osztályba egy `contains(p)` műveletet, mely eldönti, hogy egy `p` pont rajta van-e a szakaszon!

Készítsen `orientation(p)` metódust a `Segment` osztályba, mely eldönti, hogy a szakasz kezdőpontjából a végpontjába mutató vektor, valamint a szakasz végpontjából a paraméterként kapott `p` pontba mutató vektor milyen orientációjú. A metódus adjon vissza 0-t, ha a `p` rajta van a szakasz által meghatározott egyenesen, adjon vissza pozitív értéket, ha a két vektor az óramutató járásával megegyező irányban van egymással, illetve negatív értéket, ha az óramutató járásával ellenkező irányú. Ez elég egyszerű: ha a `p` pont koordinátáit x_3 és y_3 jelöli, akkor a metódus az alábbi kifejezést adja vissza.

$$(y_2 - y_1)(x_3 - x_2) - (y_3 - y_2)(x_2 - x_1)$$

Készítsen egy `intersects(s)` metódust, mely visszaadja, hogy a szakasznak van-e közös pontja a paraméterként kapott `s` szakasszal! A megoldáshoz használja az alábbi segítséget!

http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf (http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf)
<Geometry1x1.pdf>

4. gyakorló feladat

a

Készítsük el a `Rectangle` osztályt, mely a koordinátatengelyekkel párhuzamos oldalú téglalapok ábrázolására alkalmas! A `Rectangle` objektumukban tároljuk el valamelyik csúcspontjuk `x` és `y` koordinátáját, valamint a téglalap szélességét és magasságát. Ez négy adattagot jelent: `x`, `y`, `width` és `height`. Legyen mindegyik típusa dupla-pontosságú lebegőpontos típus.

A szélesség és a magasság felvehet negatív értéket is. Legyen például az `r` négy adattagja rendre 1, 5, 6, -2. Ekkor az `r` bal alsó csúcsának koordinátái 1 és 3 lesznek.

Definiáljuk a `Rectangle` objektumokon a `topLeft()`, a `topRight()`, a `bottomLeft()` és a `bottomRight()` metódusokat, melyek mindegyike egy `Point` objektumot ad vissza, értelemszerűen a téglalap megfelelő csúcsának a koordinátáit.

A `Rectangle` osztályhoz készítsünk egy főprogramot, mely meghatározza a parancssori argumentumaként kapott téglalapok befoglaló téglalapjának csúcspontjait. A főprogram parancssori argumentumai számok legyenek (legalább 4). Minden számnégyes egy `Rectangle` objektumot határoz meg. Ezeket kell feldolgozni, és a végén kiírni a befoglaló téglalap bal alsó és jobb felső csúcsainak koordinátáit.

```
$ java RectangleMain 3 5 1 -7 2 5 8 8
Bounding rectangle: 2.0;-2.0 - 10.0;13.0
```

A fenti példa két téglalap befoglaló téglalapját határozza meg. Az első téglalap egyik csúcsa (3,5) koordinátájú, szélessége 1, magassága -7. A másik téglalap egyik csúcsa (2,5) koordinátájú, szélessége és magassága egyaránt 8. Az eredményként kapott befoglaló téglalap bal alsó csúcsa (2,-2), jobb felső csúcsa (10,13), amely egyébként a második kapott téglalap jobb felső csúcsa is.

Segítség: a befoglaló téglalap bal alsó csúcsának x-koordinátájának meghatározásához keressük meg a legkisebb értéket a kapott téglalapok bal alsó csúcsának x-koordinátái között, stb.

b

Vegyük észre, hogy a `Double.max(,)` és `Double.min(,)` függvényekkel kényelmese(bbe)n tudjuk megoldani a fenti feladatot, mint elágazásokkal és "?:"-kifejezésekkel!