

Informatika 4, 1. zárthelyi (2017-10-30)

1	2	3	4	Σ

A feladatok megoldása elfér a feladat alatt, ha külön lapra írjuk, ezt jelöljük egyértelműen!

1. Írjunk függvényt (metódust), mely a bemeneti n pozitív egész számról eldönti, hogy prím-e. Ha n prím, akkor adjon vissza *true*-t, különben *false*-t. (10 pont)

```
public boolean prime(int n) {
```

```
class Position {
    public Position(); // (0, 0) pont
    public Position(float xn, float yn);

    public float distance(Position other);
        // megadja a az other-től vett távolságot
    public void translate(float xt, float yt);
        // eltolja a pontot az (xt, yt) vektorral
}
```

```
class Circle {
```

```
}
```

2. Írjunk *Circle* nevű osztályt, mely köröket reprezentál síkban. Használjuk az alábbi *Position* osztályt, melynek szándékosan csak a publikus konstruktorainak és metódusainak a fejét írtam ki, ezeket használhatjátok. Nem mondom meg, milyen adattagok kellenek, a lényeg, hogy a metódusok a megfelelő kimenetet adják. A következők legyenek meg:

Konstruktorok: (8 pont)

default: 1 sugarú kör a (0, 0)-ban

Circle(float r, float x, float y): r sugarú kör az (x, y) -ban. Ha r -nek negatív érték érkezik, akkor legyen a kör sugara 0.

Metódusok: (12 pont)

void translate(float x, float y): tolja el a kört az (x, y) vektorral.

void resize(float r): állítsa át a kör sugarát r -re.

float distance(Circle other): adja meg az adott kör(lap) távolságát az *other*-től. Ha metszenek, akkor 0-t adjon vissza.

boolean overlap(Circle other): adja vissza, hogy az adott kör érintkezik-e az *other* körrel. Ha igen, adjon vissza *true*-t, különben *false*-t.

3. Írjunk main függvényt (nem kell a körítés, csak a tartalom), melyben létrehozunk egy `HashSet`-et, feltöltjük 1-től 100-ig egész számokkal. Majd töröljük belőle az összes 3-al osztható számot! (12 pont)

```
public static void main(String[] args) {
```

```
}
```

4. Mit ír ki az alábbi kód? (8 pont)

```
public class Valami {
    private List<String> c = new ArrayList<String>();

    public Valami(int n) {
        for(int i = 0; i < n; i++) {
            c.add("kiskutya");
        }
    }

    public int doSomething(int n) {
        int i = 0;
        Iterator<String> iter = c.iterator();
        while(iter.hasNext()) {
            String s = iter.next();
            if(i > n) {
                System.out.println(s);
            }
            i++;
        }
        return i;
    }

    public static void main(String[] args) {
        Valami v1 = new Valami(10);
        Valami v2 = new Valami(4);
        System.out.println(v1.doSomething(12));
        System.out.println(v2.doSomething(3));
    }
}
```