

Informatika 4, 2. zárthelyi (2018-11-05)

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

1. Írjunk `Function` nevű `interface`-t, ami $\mathbb{R} \mapsto \mathbb{R}$ függvényeket reprezentál. Legyen egy metódusa: `compute`, ami egy `float`-ot kap értékül és azt is ad vissza.

Egy osztály implementálja ezt: `Power`, ami a hátványfüggvény. Ennek legyen egy egyparaméteres konstruktora, ami eltárol egy kitevőt, majd a `compute` metódus ezzel a kitevővel számoljon hatványt. (10 pont)

```
public interface Function
{
```

```
}
```

```
public class Power implements Function
{
```

```
}
```

Valahogy így kell működnie:

```
Function kob = new Power(3); // x -> x^3
System.out.println(kob.compute(2)); // 8
```

2. Írjuk meg az órán tanult `PlanarObject`, `Square` és `Triangle` osztályokat. Az osztályoknál figyelni kell, hogy mi miből öröklődik és hogyan, valamint hogy milyen metódusokat teszünk és hova. A metódusoknak elég csak a szignatúráját megírni és beleírni szóval hogy mit csinál, ahol szükséges. (10 pont)

Segítségül meg van adva a `Point` osztály. Emlékeztetőül, a `PlanarObject` absztrakt osztálynak volt egy `Point` tömb tagváltozója valamint egy absztrakt `area` metódusa és egy nem absztrakt `translate` metódusa (síkbeli objektumot úgy tolunk el, hogy az őt alkotó pontokat mind eltoljuk).

Ebből öröklődött a `Square` és a `Triangle`. Az előbbi a középpontját tárolta és az oldalhosszát, az utóbbi a három csúcspontját. Ezeknek felül kell definiálniuk az `area` metódust (ami a síkidom területét adja vissza)!

```
public class Point
{
    private float x_, y_;
    public Point()
    {
        x_ = 0;
        y_ = 0;
    }
    public Point(float x, float y)
    {
        x_ = x;
        y_ = y;
    }
    public void translate(float x, float y)
    {
        x_ += x;
        y_ += y;
    }
}
```

Példa a működésre:

```
PlanarObject[] objects = new PlanarObject[2];
objects[0] = new Square(0,0,1);
objects[1] = new Triangle(0,0, 0,1, 1,0);

objects[0].translate(1,2);

for (int i = 0; i < objects.length; ++i)
    System.out.println(objects[i].area());

// 1
// 0.5
```