

# 3. gyakorlat anyaga

## Demo

### 1. feladat

a

A korábban készített `Point` osztályához (itt elérhető) készítsen `toString()` metódust, amely visszatér egy pont objektum belső állapotával: `(x,y)` alakú sztring, amely tartalmazza az x és y koordinátát.

Készítsünk főprogramot, amely beolvas a felhasználótól 3 db `Point` koordinátáit, majd példányosít ilyen objektumokat, amelyek referenciáit tömbben tárolja.

A főprogram feladata, hogy kiszámítsa a tárolt pontok tömegközéppontját (ami szintén egy pont), majd az eredményt kiírja a képernyőre.

b

Módosítsuk az **a** megoldást úgy, hogy a tömegközéppont kiszámítását a `Point` osztály egy statikus metódusa végezze, amely a pontokat paraméterként tömbben fogadja, az eredmény pontot visszatérési értéként adja vissza.

A főprogram először kérdezze meg a tárolni kívánt pontok számát, majd ennyi darab pontot kérjen be a felhasználótól.

c

Módosítsuk a **b** megoldást úgy, hogy minden létrehozott `Point` objektumnak legyen egyedi azonosító száma (`id`, egész szám), amely számozás kezdődjön 1-től. Ehhez tárolja a `Point` osztályban egy statikus adattagban, hogy a következő példányosításkor mi legyen a létrehozott pont ID-ja, majd példányosításkor növelje meg ezt az adattagot. A pont `toString()` metódusa tartalmazza a pont ID-jét is.

## Feladatok

### 1. feladat

a

Írjon Java programot, amely az `{1.3, 5.2, 7.7, -2.3, 23.45}` lebegőpontos számokat tartalmazó tömbben megkeresi azt az elemet, amely legkevésbé tér el az átlagtól. Ebben a megoldásban felhasználhatja, hogy tudjuk, hogy 5 darab szám között keresünk.

b

Módosítsa az **a** megoldást úgy, hogy a program a felhasználótól olvassa be a tömb elemeit. Először kérjen be egy `N` darabszám értékét, majd hozzon létre egy ekkora méretű `double` tömböt, amelyet töltsön fel a billentyűzetről beolvasott `N` darab számmal.

### 2. feladat

Main.java:

```

class Foo {
    private int x;

    public Foo(int init_x) {
        x = init_x;
    }
}

public class Main {
    public static void main(String[] args) {
        int counter = 10;

        Foo obj;
        obj = new Foo(5);

        Foo obj2 = new Foo(7);
        obj2 = obj;
    }
}

```

Írja át `Foo` konstruktorát hogy `init_x` helyett `x` legyen a paraméter, tegye egyértelművé a `this` kulcsszóval, hogy melyik `x` azonosítóra hivatkozik.

### 3. feladat

Szervezze a demó során elkészített `Point` osztályt a `mass` csomagba. A főprogram legyen a `mass.Main` osztály, a pont osztály pedig a `mass.util.Point` osztály.

### 4. feladat

Készítsünk egy `utils.DoubleVector` osztályt (valós számokat tartalmazó tömb mint vektor segítségével). Az osztálynak egy konstruktora van, amely `double` tömböt fogad, és lemásolja annak tartalmát. Egy vektornak a következő műveletei vannak: két vektor skaláris szorzatának, összegének, különbségének ill. vektor skalárral való szorzatának kiszámítása, valamint a vektor sztringként történő ábrázolása (`toString()`).

Készítsünk főprogramot is, amely teszteli ezen műveleteket!

### 5. feladat

Bővítse az előző feladat megoldását a következő metódusokkal:

A vektorhoz lehessen hozzáadni egy újabb elemet. Itt figyeljünk rá, hogy a mérete dinamikusan növekedjen (ha megtelt a tömb, akkor csináljunk egy segéd tömböt 2x akkora mérettel, másoljuk át az elemeket és állítsuk át a számokat tároló tömb referenciáját a segéd tömbére).

Legyen egy statikus `sum()` függvénye, amely vár két `IntVector` objektumot és összeadja őket, majd visszatér az eredmény referenciájával.

### 6. feladat

Készítsen `stringutils` néven csomagot. A `stringutils.IterLetter` osztály konstruáláskor fogadjon egy `String` referenciát (kezeljük azt az esetet, ha ez `null`). Az osztálynak legyen egy `printNext()` metódusa, amellyel új sorban a képernyőre írjuk a sztring következő karakterét.

Ha a sztring összes karakterét kiírtuk a képernyőre, akkor a metódus többé ne írjon ki semmit. Az osztálynak legyen egy `restart()` metódusa, amely hatására a következő `printNext()` hívás a sztring elejét kezdi el kiírni. Az osztálynak legyen egy `hasNext()` metódusa, amely `true` értékkel tér vissza, ha van még kiírható elem.

Készítsen `Main` néven főprogramot, amely legyen névtelen csomagban. A főprogram példányosít egy `stringutils.IterLetter` osztályt, majd bemutatja annak használatát.

Bővítse a `stringutils` csomagot a `stringutils.IterWord` osztállyal; az osztály konstruáláskor fogadjon egy sztringet. Az osztály `printNext()` metódusa új sorban a képernyőre írja a sztring következő szavát. Az osztálynak szintén legyen `restart()` és `hasNext()` metódusa.

Bővítse a `Main` főprogramot az új osztály bemutatásával!