



PROGRAMAZIO MODULARRA ETA OBJEKTUEI BIDERATUTAKOA

SALBUESPENEN AZTERKETA –2017ko Ekainak 26

Ondorengo klaseen inplementazioa kontuan izanda, *Funtzioa* (TAD abstraktua), *Grafikoa* (TAD), eta *Main* (Programa nagusia):

```
public abstract class Funtzioa
{
    public abstract int f(int pX);

    public void marraztu(int pZabalera, int pAltuera)
    {
        int x = 0;
        Grafikoa graf = new Grafikoa(pZabalera, pAltuera);
        while (x < pZabalera)
        {
            graf.marraztuPuntua(x, f(x));
            x++;
        }
        graf.inprimatu();
    }
}

public class Grafikoa
{
    private char[][] puntuenMatrizea;

    public Grafikoa(int pZabalera, int pAltuera)
    {
        this.puntuenMatrizea = new char[pAltuera][pZabalera];

        for (int y = 0; y < pAltuera; y++)
        {
            for (int x = 0; x < pZabalera; x++)
            {
                this.puntuenMatrizea[y][x] = ' '; //Zuriunea
            }
        }
    }

    public void marraztuPuntua(int pX, int pY)
    {
        this.puntuenMatrizea[pY][pX] = '*';
    }

    public void inprimatu()
    {
        for (int y = puntuenMatrizea.length-1; y >= 0; y--) //Lerroa
        {
            for (int x = 0; x < puntuenMatrizea[0].length; x++) //Zutabea
            {
                System.out.print(this.puntuenMatrizea[y][x]);
            }
            System.out.println();
        }
    }
}
```



Funtzioa superklaseak, $y=f(x)$ motako formula matematikoak irudikatzea ahalbidetzen du, non x eta y balioak beti izango diren zenbaki naturalak. *Funtzioa* klasearen azpiklase bakoitzak *int f(int pX)* funtzioa inplementatuko dute, honen helburua, parametro bezala jasotzen duen puntuaren balioa kalkulatu eta itzultzea da. Hauetako azpiklase batzuk, *FuntzioZero*, *FuntzioKonstantea* eta *FuntzioKonplexua* dira eta *Main* klasean erabiltzen dira, nahiz eta kasu honetan, hauen inplementazioak ez duen garrantzirik.

Funtzioa klaseko *marratu(int pZabalera, int pAltuera)* metodoak, $[0, pZabalera-1]$ artaz tarteko balioen errepresentazioa grafikoa pantailaratuko du. Helburu horretarako, lehenik, Grafikoa izeneko objektu bat sortuko du (zurienez osaturiko matrize bat izanik) eta ondoren, $y=f(x)$ funtzioak itzultzen duen balioa marrazten joango da matrizean, hau da, izarra ('*') karakterea idatziko du funtzioak itzultzen duen balioa matrizean. Behin puntu guztiak matrizean marraztu ondoren, *inprimatu()* metodoak grafikoa pantailaratuko du.

Eskatzen dena, zera da, *Funtzioa* eta *Grafikoa* klaseen inplementazioa moldatzea ondoren azaltzen diren salbuespen kasuei aurre egiteko:

- *f(pX)* metodoak, *ArithmeticException* salbuespena jaurti dezake funtzioaren kalkulua burutzerako garaian, adibidez, zenbaki bat zerorengatik zatitzen denean. *ArithmeticException* klasea, jada Javan aurredefiniturik dago, beraz, ez da metodoaren buruan adierazi behar salbuespena jaurti edo propagatzen dela adierazteko. Dena den, honelako salbuespen kasu bat gertatzen denean, grafikoa puntuak marrazteari utziko dio, pantailatik errore hau gertatu dela adieraziz eta programa amaituaz (**0.15 puntu**).
- Grafikoa tartearen barruan ez dagoen puntu bat marrazten saiatzean, bi salbuespen posible jaurti ahalko dira: *XOutOfBoundsException*, *pX* balioa $[0, zabalera-1]$ tartetik kanpo balego; eta/edo *YOutOfBoundsException*, *pY* balioa $[0, altuera-1]$ tartetik kanpo balego (**0.15 puntu**).
Oharra: Matrizearen *altuera* *matrizea.length* bitartez atzitu daiteke, *zabalera* berriz *matrizea[0].length* bitartez.
- *YOutOfBoundsException* salbuespenaren tratamendua, grafiko handiago (altuagoan zehazki) batean marrazten saiatzea izango da. Horretarako, karakterez osatutako *pZabalera* eta *pAltuera* tamainako grafikoa sortu ordez, *pAltuera+10* tamainako grafikoa eraikiko du. Arazoa gertatzen jarraitzen bada, berriz beste 10 karaktereko tamaina gehituko zaio *pAltuerari*, horrela gehienez 12 saiakera gertatu arte. Behin 12 saiakerak eginda *YOutOfBoundsException* salbuespena gertatzen jarraitzen bada, (balioa negatiboa delako edo handiegia delako), arazoa pantailatik erakutsiko da eta programaren exekuzioa amaituko da (**0.4 puntu**).
- *XoutOfBoundsException* salbuespena gertatzen denean, puntu hori alde batera utziko da eta hurrengo puntuak marrazten jarraituko da (**0.2 puntu**).
- *XoutOfBoundsException* klasea inplementatu ezazu (**0.1 puntu**).