

# Primer Parcial d'IIP - ETSInf

Data: 26 de novembre de 2012. Duració: 1 hora i 30 minuts.

1. 6 punts Una aplicació per a la borsa de valors usa una classe per a definir el valor de les accions. Cada acció queda identificada pel nom de l'empresa i conté, com a dades, quatre valors reals, els d'obertura, el mínim, el màxim i l'actual d'una acció. Cada acció comença la sessió del dia amb el valor d'*obertura*, en un moment determinat té un valor *actual* i durant la sessió ha tingut valors *minim* i *maxim* que poden haver canviat al llarg de la sessió.

Per a aquesta classe **Accio**, **has de**:

- a) (0.5 punts) Definir els atributs d'instància privats **empresa**, **obertura**, **minim**, **maxim** i **actual**.
  - b) (1 punt) Implementar dos constructors:
    - Un que rep tan sols el nom de l'empresa i el valor d'obertura; on **minim**, **maxim** i **actual** tindran el mateix valor que el d'obertura (pots suposar que el valor d'obertura és correcte).
    - Un altre que rep el nom de l'empresa, el valor d'obertura i els valors mínim i màxim; l'atribut **actual** tindrà el mateix valor que el d'obertura (pots suposar que tots els valors rebuts són correctes).
  - c) (1 punt) Escriure un mètode consultor per a cada atribut.
  - d) (1 punt) Escriure el mètode modificador de l'atribut **actual** que ha d'actualitzar adequadament, en el seu cas, els valors de **minim** i **maxim**.
  - e) (0.5 punts) Escriure un mètode **capAMunt** que comprovi si l'acció en curs està donant beneficis (a.e., si el seu valor actual és major que el d'obertura).
  - f) (1 punt) Escriure un mètode **equals** (que sobreescriu el de **Object**) per a comprovar si dues accions són iguals. Dues accions són iguals quan pertanyen a la mateixa empresa.
  - g) (1 punt) Escriure un mètode **toString** (que sobreescriu el de **Object**) que torne una **String** en la que apareix: "**empresa: actual minim maxim**"; p.e., "Iberdrola: 12.30 10.34 13.21".
2. 4 punts Prenent en consideració la classe de la qüestió anterior, **s'ha d'implementar** en Java una classe **BorsaValors** amb els següents mètodes:
- a) (1 punt) Un mètode de classe (**static**) que, donats dos objectes **Accio**, torne aquell que siga més *volàtil*, a.e., tal que presenti un major percentatge de diferència entre els seus valors mínim i màxim. Per exemple, una acció amb mínim 12.3 i màxim 15.7 representa una volatilitat del 27.6% ( $\frac{(15.7-12.3) \times 100}{12.3} = 27.6$ ).
  - b) (1 punt) Un mètode de classe (**static**) que, donats dos nombres reals **a** i **b**, torne un valor aleatori real en l'interval  $[\min(a, b), \max(a, b)]$ .
  - c) (2 punts) Un mètode **main** en el que s'haurà de:
    1. Crear un objecte **Accio** de l'empresa "Iberdrola", amb valor d'obertura 12.30, mostrant a continuació les seves dades en la pantalla.
    2. Crear un objecte **Accio** preguntant-li a l'usuari pel nom de l'empresa i calculant a l'atzar (aleatòriament) els valors d'obertura, mínim i màxim on, necessàriament, s'ha de complir que tots els valors estan compresos en el rang  $[1, 100[$  i són correctes (a.e., mínim  $\leq$  obertura  $\leq$  màxim). A continuació, s'hauran de mostrar les dades de l'acció en la pantalla.
    3. Preguntar a l'usuari per un nou valor actual del primer objecte **Accio** que s'ha creat, actualitzar-lo i mostrar-lo per pantalla indicant si té beneficis o no.
    4. Mostrar per pantalla l'objecte **Accio** que siga més volàtil dels dos que s'han creat, usant per a això, el mètode definit prèviament.

## Solució:

Accio.java

```
public class Accio {
    private String empresa;
    private double obertura, minim, maxim, actual;

    public Accio(String nom, double ob) {
        empresa = nom;
        obertura = minim = maxim = actual = ob;
    }

    public Accio(String nom, double ob, double mi, double ma) {
        empresa = nom;
        obertura = ob;
        minim = mi;
        maxim = ma;
        actual = ob;
    }

    public String getEmpresa() { return empresa; }
    public double getObertura() { return obertura; }
    public double getMinim() { return minim; }
    public double getMaxim() { return maxim; }
    public double getActual() { return actual; }

    public void setActual(double a) {
        if (a<minim) minim = a;
        else if (a>maxim) maxim = a;
        actual = a;
    }

    public boolean capAMunt() { return (actual>obertura); }

    public boolean equals(Object o) {
        return o instanceof Accio &&
            empresa.equals(((Accio) o).empresa);
    }

    public String toString() {
        return empresa + ": " + actual + " " + minim + " " + maxim;
    }
} // de la classe Accio
```

Accio.java

BorsaValors.java

```
import java.util.*;
public class BorsaValors {

    public static Accio mesVolatil(Accio a1, Accio a2) {
        double v1 = (a1.getMaxim()-a1.getMinim())/a1.getMinim();
        double v2 = (a2.getMaxim()-a2.getMinim())/a2.getMinim();
        if (v1>v2) return a1; else return a2;
    }
}
```

```

public static double intervalAleatori(double a, double b) {
    double m1 = Math.min(a,b), m2 = Math.max(a,b);
    return Math.random()*(m2-m1)+m1;
}

public static void main(String[] args) {
    Scanner teclat = new Scanner(System.in).useLocale(Locale.US);

    /** 1 **/
    Accio a1 = new Accio("Iberdrola",12.30);
    System.out.println("Accio 1: " + a1);

    /** 2 **/
    System.out.print("Nom de l'empresa per a l'accio 2: ");
    String nom = teclat.nextLine();
    double min = intervalAleatori(1,100);
    double max = intervalAleatori(min,100);
    double obe = intervalAleatori(min,max);

    Accio a2 = new Accio(nom,obe,min,max);
    System.out.println("Accio 2: " + a2);

    /** 3 **/
    System.out.print("Valor actual per a l'accio 1: ");
    double act1 = teclat.nextDouble();
    a1.setActual(act1);
    if (a1.capAMunt()) System.out.println("Esta donant beneficis " + a1);
    else System.out.println("No esta donant beneficis " + a1);

    /** 4 **/
    System.out.println("La accio mes volatil es: " + mesVolatil(a1,a2));
}

} // de la classe BorsaValors
BorsaValors.java

```