

## Primer Parcial d'IIP (ETSIInf)

6 de Novembre de 2017. Duració: 1 hora i 30 minuts

**Nota:** L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final d'IIP és de **2,4 punts**

NOM:

GRUP:

1. 6 punts Es vol dissenyar una classe Tipus de dades anomenada **ProgramaRadio** per tal de representar un programa de ràdio. Cada programa de ràdio té associat un títol, una hora d'inici i una hora de fi (sent ambdues del mateix dia i l'hora d'inici anterior a la de fi) i un tipus de programa que pot ser magazine, música o notícies. Per a representar les hores d'inici i de fi es disposa de la classe **Instant** amb la funcionalitat que es mostra en part, a continuació, en la seua documentació:

## Constructor Summary

Constructors	
Constructor and Description	
<b>Instant</b> (int h, int m)	Crea un Instant amb h hores i m minuts.

## Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
int	<b>aMinuts()</b> Torna el número de minuts transcorreguts des de les 00:00 fins l'Instant en curs.	
int	<b>compareTo(Instant altre)</b> Compara cronològicament l'Instant en curs amb l'Instant <code>altre</code> , tornant un valor negatiu si <code>this</code> és anterior a <code>altre</code> , zero si són iguals, i un valor positiu si <code>this</code> és posterior a <code>altre</code> .	
java.lang.String	<b>toString()</b> Torna l'Instant en curs en el format "hh:mm".	

**Es demana:** implementar la classe **ProgramaRadio**, considerant que està en el mateix directori que la classe **Instant**, amb els atributs i mètodes que s'indiquen a continuació:

- a) (0.25 punts) Atributs de classe públics i constants de tipus enter:

- **MAGAZINE**, amb valor 0 que representa el tipus de programa magazine.
- **MUSICA**, amb valor 1 que representa el tipus de programa musical.
- **NOTICIES**, amb valor 2 que representa el tipus de programa notícies.

Aquestes constants han de ser utilitzades sempre que es requereisca (tant a la classe **ProgramaRadio** com a la classe **GestorRadio**).

- b) (0.5 punts) Atributs d'instància privats: **tipus** (int), **títol** (String), **horaInici** (Instant) i **horaFi** (Instant).
- c) (1.25 punts) Un constructor general tal que, donats el tipus de programa, el seu títol, les hores i els minuts de l'inici, i la duració en minuts, inicialitze tots els atributs d'instància. Supposeu que les dades són correctes.
- d) (1 punt) Un mètode **equals** (que sobreescriu el d'**Object**) per comprovar si dos programes de ràdio són iguals, és a dir, si són del mateix tipus i tenen el mateix títol.
- e) (1 punt) Un mètode **toString** (que sobreescriu el d'**Object**) perquè, usant obligatòriament una instrucció **switch** per convertir en text el tipus de programa, torne el resultat amb un format com el mostrat en els següents exemples:

06:30 - Anda ya (Música)  
14:30 - Todo noticias (Notícies)  
16:00 - Julia en la Onda (Magazine)

- f) (2 punts) Dos programes de ràdio (que s'emeten el mateix dia) es consideren ordenats en la graella de programació d'acord amb els següents criteris:
- Va primer el programa que comença abans.
  - A igual hora d'inici, va primer el que acaba abans.
  - A igual horari d'inici i fi, van primer els programes de notícies, després els de música i, per últim, els magazines.
  - En el cas que coincideixen en horari i tipus, és indiferent l'ordre en què apareixen a la graella.

Implementar un mètode **compareTo** que, donat un **ProgramaRadio** passat com paràmetre **p** que s'emet el mateix dia que **this**, torne un **int** negatiu si **this** és anterior a **p** en la graella, positiu si **p** és anterior i 0 si és indiferent.

## Solució:

```
public class ProgramaRadio {
    public static final int MAGAZINE = 0;
    public static final int MUSICA = 1;
    public static final int NOTICIES = 2;

    private int tipus;
    private String titol;
    private Instant horaInici;
    private Instant horaFi;

    public ProgramaRadio(int tip, String tit, int hora, int min, int duracio) {
        tipus = tip;
        titol = tit;
        horaInici = new Instant(hora, min);
        int fi = horaInici.aMinuts() + duracio;
        horaFi = new Instant(fi / 60, fi % 60);
    }

    public boolean equals(Object o) {
        return o instanceof ProgramaRadio
            && tipus == ((ProgramaRadio) o).tipus
            && titol.equals(((ProgramaRadio) o).titol);
    }

    public String toString() {
        String res = horaInici + " - " + titol + " (";
        switch (tipus) {
            case MAGAZINE:
                res += "Magazine)";
                break;
            case MUSICA:
                res += "Música)";
                break;
            case NOTICIES:
                res += "Notícies)";
        }
        return res;
    }

    public int compareTo(ProgramaRadio p) {
        int res = horaInici.compareTo(p.horaInici);
        if (res == 0) {
            res = horaFi.compareTo(p.horaFi);
            if (res == 0) { res = p.tipus - tipus; }
        }
        return res;
    }
}
```

2. 2 punts **Es demana:** implementar la classe Programa **GestorRadio**, en el mateix directori que **ProgramaRadio**, amb un mètode **main** que realitzi les següents accions:

- (0.25 punts) Crear un objecte **p1** de tipus **ProgramaRadio**, per representar un programa de ràdio magazine, de títol **Julia** en la **Onda**, que comença a les 16:00 i té una duració de 180 minuts.
- (0.25 punts) Crear un objecte **p2** de tipus **ProgramaRadio**, per representar un programa de ràdio musical, de títol **Anda ya**, que comença a les 6:30 i té una duració de 300 minuts.
- (1.5 punts) Comparar **p1** amb **p2** usant el mètode **compareTo** i, en funció del seu resultat, mostrar-los per pantalla segons el seu ordre en la graella.

## Solució:

```
public class GestorRadio {
    public static void main(String[] args) {
        ProgramaRadio p1 = new ProgramaRadio(ProgramaRadio.MAGAZINE, "Julia en la Onda",
                                                16, 0, 180);

        ProgramaRadio p2 = new ProgramaRadio(ProgramaRadio.MUSICA, "Anda Ya",
                                                6, 30, 300);

        if (p1.compareTo(p2) < 0) { System.out.println(p1 + "\n" + p2); }
        else { System.out.println(p2 + "\n" + p1); }
    }
}
```

3. 2 punts Es disposa de la classe **Punt** que defineix un punt en un espai bidimensional real (amb dos atributs representant la seua abscissa i la seua ordenada), amb la funcionalitat que es mostra en part, a continuació, en la seua documentació:

## Constructor Summary

Constructors	
Constructor and Description	
<code>Punt(double abs, double ord)</code>	Crea un Punt amb coordenades (abs, ord).

## Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
<code>void</code>	<code>setX(double nou)</code>	Actualitza a nou el valor de l'abscissa del Punt en curs.
<code>void</code>	<code>setY(double nou)</code>	Actualitza a nou el valor de l'ordenada del Punt en curs.
<code>java.lang.String</code>	<code>toString()</code>	Torna un String amb les dades del Punt en curs en el format (x, y).

Donada la següent classe programa:

```
public class Exercici3 {
    public static void main(String[] args) {
        Punt p = new Punt(1.0, -1.0);
        double x = 1.0, y = -1.0;
        System.out.println("Abans de canviarCoord: x = " + x + ", y = " + y
                           + ", p = " + p);

        canviarCoord(x, y, p);
        System.out.println("Després de canviarCoord una vegada: x = " + x + ", y = " + y
                           + ", p = " + p);

        canviarCoord(x, y, p);
        System.out.println("Després de canviarCoord 2 vegades: x = " + x + ", y = " + y
                           + ", p = " + p);
    }

    public static void canviarCoord(double x, double y, Punt p) {
        p.setX(y);
        p.setY(x);
    }
}
```

Es demana: Completar què es mostra per pantalla després de la seua execució.

Abans de canviarCoord: x = 1.0, y = -1.0, p = (1.0, -1.0)

Després de canviarCoord una vegada: x = 1.0, y = -1.0, p = (-1.0, 1.0)

Després de canviarCoord 2 vegades: x = 1.0, y = -1.0, p = (-1.0, 1.0)