

Primer Parcial d'IIP (ETSIInf)
31 d'Octubre de 2019. 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 **3,75 punts**

NOMBRE:

GRUP:

1. 6 punts Es vol dissenyar una classe Tipus de Dades denominada **PieceOfNews** per a representar una notícia que serà publicada en un mitjà digital el mateix dia en què es produeix. Cada notícia té associats els següents elements: hora en la qual es produeix; enllaç al fitxer que conté la informació a publicar; nombre de mitjans que s'han fet eco d'ella el mateix dia; tipus, que pot ser text, vídeo o àudio.

Per a representar l'instant del dia en què s'ha produït una notícia, es disposa de la classe d'usuari **TimeInstant**, la documentació del qual es mostra -parcialment- a continuació:

Constructors		
Constructor	Description	
<code>TimeInstant()</code>	Crea un <code>TimeInstant</code> amb el valor de l'instant actual UTC (temps universal coordinat).	
<code>TimeInstant(int iniHours, int iniMinutes)</code>	Crea un <code>TimeInstant</code> amb el valor de les hores i els minuts que rep com arguments, <code>iniHours</code> i <code>iniMinutes</code> , respectivament.	

Method Summary		
All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
<code>int</code>	<code>compareTo (TimeInstant tInstant)</code>	Compara cronològicament l'instant de l'objecte en curs amb el de l'objecte de la classe <code>TimeInstant</code> referenciat per <code>tInstant</code> . El resultat serà un valor negatiu si l'instant de l'objecte en curs és anterior al de <code>tInstant</code> , zero si són iguals, i positiu si l'instant de l'objecte en curs és posterior al del <code>tInstant</code> .
<code>boolean</code>	<code>equals (java.lang.Object o)</code>	Torna <code>true</code> si i o és un objecte de la classe <code>TimeInstant</code> i les seues hores i minuts coincideixen amb els de l'objecte en curs.
<code>int</code>	<code>getHours()</code>	Torna les hores del <code>TimeInstant</code> .
<code>int</code>	<code>getMinutes()</code>	Torna els minuts del <code>TimeInstant</code> .
<code>void</code>	<code>setHours(int hh)</code>	Actualitza les hores del <code>TimeInstant</code> .
<code>void</code>	<code>setMinutes(int mm)</code>	Actualitza els minuts del <code>TimeInstant</code> .
<code>java.lang.String</code>	<code>toString()</code>	Torna el <code>TimeInstant</code> en el format " <code>hh:mm</code> ".

Es demana: implementar la classe **PieceOfNews**, (se suposa que en el mateix paquet que la classe **TimeInstant**) amb els següents atributs i mètodes:

- (0.5 punts) Tres atributs estàtics, públics i constants de tipus `int`, per a representar mitjançant un codi numèric els tres tipus de formats que pot tenir el fitxer que conté una notícia: àudio, vídeo o text. Els seus identificadors i valors són, respectivament: **AUDIO**, amb valor 0; **VIDEO**, amb valor 1; **TEXT**, amb valor 2.
Aquestes "constants Java" s'han d'usar sempre que es requereisca en la classe **PieceOfNews**.
- (0.5 punts) Quatre atributs d'instància i privats, per a representar els elements associats a una **PieceOfNews**. Seguint l'ordre en el qual s'han descrit prèviament, els seus identificadors (i tipus Java) són: **instant** (`TimeInstant`); **link** (`String`); **echoedBy** (`int`); **type** (`int`).
- (0.75 punts) Mètode constructor, que crea una **PieceOfNews** que s'ha produït en l'instant **i**, publicada amb enllaç **l**, eco en **n** mitjans i de tipus **t**. Se suposa com a Precondició que els valors d'aquests paràmetres són correctes.
- (1.25 punts) Mètode **equals**, que sobreescriu el d'`Object` i comprova si una **PieceOfNews** (**this**) és igual a una altra, en concret, si ambdues s'han produït en el mateix instant, han tingut el mateix eco i són del mateix tipus; els enllaços no es tenen en compte.
- (1.75 punts) Mètode **compareTo**, que compara una **PieceOfNews** (**this**) amb una altra notícia **other** sobre la base dels criteris de popularitat que figuren a continuació i retorna un `int` negatiu si **this** és menys popular que **other**, positiu si **this** és més popular que **other** i 0 si **this** i **other** són igual de populars.

Criteris de popularitat:

- En principi, una notícia és menys popular que una altra si ha succeït abans.
- Si han succeït en el mateix instant, llavors una notícia és menys popular que l'altra si el seu eco és menor.
- A igualtat d'instant i eco, una notícia és menys popular si està menys elaborada que l'altra, entenent-se que les notícies menys elaborades vénen com a àudio, les intermèdies vénen com a vídeo, i les més elaborades com a text.

- f) (1.25 punts) Mètode `toString`, que sobreescriu el d'`Object` i que retorna la descripció de la notícia, és a dir, l'instant, l'enllaç, el seu eco i, entre parèntesi, una paraula que indique el seu tipus: `text`, `video`, `audio`, com en el següent exemple:

10:30 <https://media.com/2019/10/31/climate-change2> 150 (text)

Solució:

```
public class PieceOfNews {
    public static final int AUDIO = 0, VIDEO = 1, TEXT = 2;

    private TimeInstant instant;
    private String link;
    private int echoedBy;
    private int type;

    public PieceOfNews(TimeInstant i, String l, int n, int t) {
        instant = i;
        link = l;
        echoedBy = n;
        type = t;
    }

    public boolean equals(Object o) {
        return o instanceof PieceOfNews
            && this.instant.equals(((PieceOfNews) o).instant)
            && this.echoedBy == ((PieceOfNews) o).echoedBy
            && this.type == ((PieceOfNews) o).type;
    }

    public int compareTo(PieceOfNews other) {
        int res = this.instant.compareTo(other.instant);
        if (res == 0) {
            res = this.echoedBy - other.echoedBy;
            if (res == 0) {
                res = this.type - other.type;
            }
        }
        return res;
    }

    public String toString() {
        String res = "";
        res += instant + " " + link + " " + echoedBy + " (";
        switch (type) {
            case TEXT:
                res += "text)"; break;
            case VIDEO:
                res += "video)"; break;
            default:
                res += "audio)";
        }
        return res;
    }
}
```

2. 2 punts **Es demana:** donada la següent classe Programa `TestPieceOfNews`, completar el mètode `main` perquè realitzi les accions que es descriuen a continuació, suposant que se situa en el mateix paquet que les classes `PieceOfNews` i `TimeInstant`. Aquesta classe programa ha d'usar les "constants Java" de les classes del mateix paquet sempre que es requereisca.

```
public class TestPieceOfNews {

    /** Torna un valor enter aleatori en [ini, fin], 0 <= ini < fin. */
    private static int random(int ini, int fin) {
        return (int) (Math.random() * (fin - ini + 1) + ini);
    }

    public static void main(String[] args) {
        ...
    }
}
```

- a) (0.25 punts) Crear un `TimeInstant` `ti` que represente les 10 hores i 30 minuts.
b) (0.25 punts) Crear una `PieceOfNews` `n1` que s'ha produït en l'instant `ti`, de tipus àudio, de la qual s'han fet eco 200 mitjans i amb l'enllaç "<https://media.com/2019/10/31/climate-change1>".

- c) (0.25 punts) Assignar a una variable `echo2` el resultat de generar aleatòriament un valor en l'interval [2, 500].
- d) (0.25 punts) Crear una `PieceOfNews n2` que s'ha produït en l'instant `ti`, de tipus text, de la qual s'han fet eco `echo2` mitjans i amb l'enllaç "<https://media.com/2019/10/31/climate-change2>".
- e) (0.25 punts) Assignar a una variable `resC` el resultat d'aplicar el mètode `compareTo` a les variables `n1` i `n2`.
- f) (0.75 punts) En funció del valor de `resC`, mostrar per pantalla (amb el format de `toString`) la notícia de major popularitat. Si la popularitat fóra la mateixa, mostrar les dues per pantalla separades pel símbol d'igualtat.

Solució:

```
public class TestPieceOfNews {

    /** Torna un valor enter aleatori en [ini, fin], 0 <= ini < fin. */
    private static int random(int ini, int fin) {
        return (int) (Math.random() * (fin - ini + 1) + ini);
    }

    public static void main(String[] args) {
        TimeInstant ti = new TimeInstant(10, 30);
        String link = "https://media.com/2019/10/31/climate-change1";
        PieceOfNews n1 = new PieceOfNews(ti, link, 200, PieceOfNews.AUDIO);

        int echo2 = random(2, 500);
        String link2 = "https://media.com/2019/10/31/climate-change2";
        PieceOfNews n2 = new PieceOfNews(ti, link2, echo2, PieceOfNews.TEXT);

        int resC = n1.compareTo(n2);
        if (resC == 0) {
            System.out.println(n1 + " = " + n2);
        }
        else if (resC < 0) {
            System.out.println(n2);
        }
        else { System.out.println(n1); }
    }
}
```

3. 2 punts Donada la següent classe `Exercise3`, en la qual s'usa la classe `TimeInstant` de la pregunta anterior, es demana escriure el que mostra per pantalla l'execució del programa.

```
public class Exercise3 {

    public static void main(String[] args) {
        TimeInstant aux = new TimeInstant(5, 6); int j = 1;
        System.out.println("En main: " + j + " " + aux.getHours() + " " + aux.getMinutes());
        m2(aux, j);
        System.out.println("En main: " + j + " " + aux.getHours() + " " + aux.getMinutes());
    }

    private static void m2(TimeInstant aux, int j) {
        System.out.println("En m2:  " + j + " " + aux.getHours());
        int nH = aux.getHours() + j;
        j++; aux.setHours(nH);
        m1(aux, j);
        System.out.println("En m2:  " + j + " " + aux.getHours());
    }

    private static void m1(TimeInstant aux, int j) {
        System.out.println("En m1:  " + j + " " + aux.getMinutes());
        int nM = aux.getMinutes() + j;
        j++; aux.setMinutes(nM);
        System.out.println("En m1:  " + j + " " + aux.getMinutes());
    }
}
```

Solució:

```
En main: 1 5 6
En m2:    1 5
En m1:    2 6
En m1:    3 8
En m2:    2 6
En main: 1 6 8
```