Segundo Parcial de IIP (ETSInf)
8 de Enero de 2020. Duración: 2 horas y 30 minutos

Nota: El examen se evalúa sobre 10 puntos, pero su peso específico en la nota final de IIP es de **3,75 puntos**

NOMBRE:	GRUPO:

1. 5.5 puntos Se desea implementar una clase *tipo de datos* denominada `BulletinBoard` que modelará un tablón de noticias digital. Para ello, se dispone de la clase `PieceOfNews` que modela una noticia publicada en un medio digital el mismo día en el que se produce con la siguiente descripción:

Fields

Modifier and Type	Field	Description
`static int`	`AUDIO`	Noticia de tipo AUDIO con valor 0
`static int`	`VIDEO`	Noticia de tipo VIDEO con valor 1
`static int`	`TEXT`	Noticia de tipo TEXT con valor 2

Constructors

Constructor	Description
`PieceOfNews(TimeInstant i, String l, int n, int t)`	Crea un objeto `PieceOfNews` que se ha producido en el instante `i`, publicada en el enlace `l`, con eco en `n` medios y de tipo de noticia `t` (AUDIO, VIDEO o TEXT).

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`int`	`compareTo(PieceOfNews other)`	Devuelve un entero negativo si `this` es menos popular que `other`, positivo si `this` es más popular que `other` y 0 si ambas son igual de populares
`TimeInstant`	`getInstant()`	Devuelve el instante de tiempo en que se ha producido la noticia.
`String`	`getLink()`	Devuelve el enlace a la noticia.
`int`	`getType()`	Devuelve el tipo a la noticia.

Se pide: implementar la clase `BulletinBoard`, (se supone que en el mismo paquete que la clase `PieceOfNews`) con los siguientes atributos y métodos:

a) (0.5 puntos) Atributos:

- `MINUTES`: atributo de clase público, estático y constante de tipo `int`, que indica el número de minutos que tiene un día (1440).
- `bBoard`: atributo de instancia privado de tipo array de objetos de la clase `PieceOfNews` y tamaño `MINUTES`, de tal manera que `bBoard[i]` almacena la noticia producida en el minuto `i` del día ($0 \leq$ `i` $<$ `MINUTES`) o `null` si no se ha producido ninguna noticia en dicho minuto. Por ejemplo, las noticias `a` (producida en el instante 00:00), `b` (producida en el instante 01:01) y `c` (producida en el instante 23:59), se almacenarán en `bBoard[0]`, `bBoard[61]` y `bBoard[1439]`, respectivamente, como puede verse en la Figura 1 en la que se representa el objeto `bB` de tipo `BulletinBoard`.
- `numPerType`: atributo de instancia privado de tipo array de `int` y tamaño 3, tal que `numPerType[i]` almacena el número de noticias de tipo `i`, $0 \leq$ `i` < 3, que hay en `bBoard`; así, `numPerType[0]`, `numPerType[1]` y `numPerType[2]` son, respectivamente, el número de noticias de tipo `AUDIO`, `VIDEO` y `TEXT` que hay en `bBoard`. Si las noticias del ejemplo anterior `a` y `c` son de tipo `AUDIO` y `b` es de tipo `TEXT`, entonces `numPerType[0]`, `numPerType[1]` y `numPerType[2]` valen 2, 0 y 1, respectivamente, como en la Figura 1.

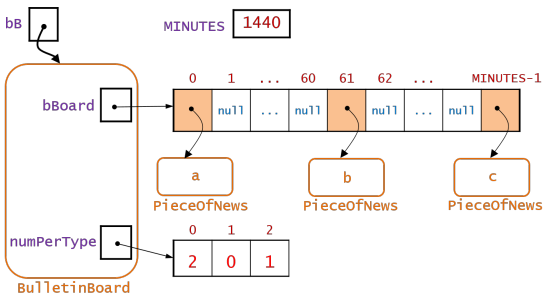

Figura 1: Representación de un objeto de tipo `BulletinBoard`.

b) (0.75 puntos) Método constructor, que crea un `BulletinBoard` sin noticias.

c) (1.5 puntos) Método `add` con la siguiente cabecera o perfil:

```
public boolean add(PieceOfNews p)
```

que, dada una `PieceOfNews p`, la añade al tablón en su instante de tiempo. Si ya existe una noticia en el mismo instante de tiempo, permanecerá la más popular según el método `compareTo`. Si el tablón se ha modificado, devolverá `true`; en caso contrario, `false`. Este método debe actualizar el atributo `numPerType` cuando proceda.

Recuerda que la clase `TimeInstant` tiene el método `toMinutes()` que devuelve el número de minutos transcurridos desde las 00:00 hasta el instante representado por el objeto en curso.

d) (1.25 puntos) Método `isPosted` con la siguiente cabecera o perfil:

```
public PieceOfNews isPosted(String link)
```

que devuelve la noticia cuyo enlace es `link`. En caso de no encontrarla, devolverá `null`.

e) (1.5 puntos) Método `filterByType` con la siguiente cabecera o perfil:

```
public PieceOfNews[] filterByType(int type)
```

que devuelve un array con todas las noticias producidas en el día de tipo `type`, siendo $0 \leq$ `type` < 3. El tamaño del array devuelto debe ser igual al número de noticias producidas en el día de tipo `type`, siendo 0 si no se ha producido ninguna noticia de dicho tipo.

Solución:

```
public class BulletinBoard {
    public static final int MINUTES = 1440;
    private PieceOfNews[] bBoard;
    private int[] numPerType;

    public BulletinBoard() {
        bBoard = new PieceOfNews[MINUTES];
        numPerType = new int[3];
    }

    public boolean add(PieceOfNews p) {
        boolean added = false;
        int position = p.getInstant().toMinutes();
        if (bBoard[position] == null || p.compareTo(bBoard[position]) > 0) {
            if (bBoard[position] != null) { numPerType[bBoard[position].getType()]--; }
            bBoard[position] = p;
            numPerType[p.getType()]++;
            added = true;
        }
        return added;
    }

    public PieceOfNews isPosted(String link) {
        PieceOfNews result = null;
        boolean found = false;
        int i = 0;
        while (i < bBoard.length && !found) {
            if (bBoard[i] != null && link.equals(bBoard[i].getLink())) {
                found = true;
                result = bBoard[i];
            }
            i++;
        }
        return result;
    }

    /** Precondición: 0 <= type < 3 */
    public PieceOfNews[] filterByType(int type) {
        PieceOfNews[] result = new PieceOfNews[numPerType[type]];
        for (int i = 0, j = 0; i < bBoard.length && j < numPerType[type]; i++) {
            if (bBoard[i] != null && bBoard[i].getType() == type) {
                result[j++] = bBoard[i];
            }
        }
        return result;
    }
}
```

2. 2.25 puntos **Se pide**: escribir un método público y estático tal que, dado un array de enteros positivos, busque y devuelva el primer valor del array que esté repetido en cualquier otra posición. En caso de que no haya repetidos, el método devolverá `-1`. Por ejemplo, para el array `a1`, el método devolverá `2` (ya que el valor `2` es el primero que está repetido) y para el array `a2`, la respuesta será `-1` (ya que no hay repetidos).

	0	1	2	3	4	5	6	7	8	9
a1	5	2	8	3	7	4	2	7	9	4

	0	1	2	3	4	5	6	7	8	9
a2	5	2	8	3	7	4	6	0	9	1

Solución:

```
/** Precondición: v es un array de enteros positivos */
public static int firstRepeatedElement(int[] v) {
    int i = 0, j;
    boolean found = false;
    while (i < v.length - 1 && !found) {
        j = i + 1;
        while (j < v.length && !found) {
            found = v[i] == v[j];
            j++;
        }
        i++;
    }
    if (found) { return v[i - 1]; }  // deshacemos el ultimo incremento
    else { return -1; }
}
```

3. 2.25 puntos Dado un valor real x y la siguiente recurrencia:

$$C_1 = x \qquad C_n = C_{n-1} * n,\ n > 1$$

Se pide: escribir un método público y estático tal que, dados dos valores `x` y `epsilon` ($0 \leq$ `epsilon` < 1) de tipo `double`:

- si `x` ≤ 0, devuelva `-1`;
- si no, devuelva el número de términos calculados para que la expresión $1/(C_n - C_{n-1}) \leq$ `epsilon` sea cierta.

Solución:

```
/** Precondición: 0 <= epsilon < 1 */
public static int numTerms(double x, double epsilon) {
    if (x <= 0) { return -1; }
    double cPrev = x;
    double cActual = cPrev * 2;
    int i = 2;
    while (1 / (cActual - cPrev) > epsilon) {
        i++;
        cPrev = cActual;
        cActual = cPrev * i;
    }
    return i;
}
}
```