

NOM:

GRUP:

1. 5.5 punts Es vol implementar una classe *tipus de dades* anomenada **BulletinBoard** que modelarà un tauler de notícies digital. Per a això, es disposa de la classe **PieceOfNews** que modela una notícia publicada en un mitjà digital el mateix dia en què es produeix amb la següent descripció:

Field Summary**Fields**

Modifier and Type	Field and Description
static int	AUDIO Notícia de tipus AUDIO amb valor 0.
static int	VIDEO Notícia de tipus VIDEO amb valor 1.
static int	TEXT Notícia de tipus TEXT amb valor 2.

Constructor Summary**Constructors****Constructor and Description**

PieceOfNews(`TimeInstant i`, `java.lang.String l`, `int n`, `int t`)

Crea una **PieceOfNews** que s'ha produït en l'instant `i`, publicada amb enllaç `l`, eco en `n` mitjans i de tipus `t` (AUDIO, VIDEO o TEXT).

Method Summary**All Methods****Instance Methods****Concrete Methods**

Modifier and Type	Method and Description
int	compareTo (PieceOfNews other) Torna un enter negatiu si <code>this</code> és menys popular que <code>other</code> , positiu si <code>this</code> és més popular que <code>other</code> i 0 si ambdues són igual de populars.
TimeInstant	getInstant () Torna l'instant de temps en que s'ha produït la notícia.
java.lang.String	getLink () Torna l'enllaç de la notícia.
int	getType () Torna el tipus de la notícia.

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

a) (0.5 punts) Atributs:

- **MINUTES**: atribut de classe públic, estàtic i constant de tipus `int`, que indica el número de minuts d'un dia (1440).
- **bBoard**: atribut d'instància privat de tipus array d'objectes de la classe **PieceOfNews** i grandària **MINUTES**, de tal manera que **bBoard**[*i*] emmagatzema la notícia produïda en el minut *i* del dia ($0 \leq i < \text{MINUTES}$) o `null` si no s'ha produït cap notícia en aquest minut. Per exemple, les notícies **a** (produïda en l'instant 00:00), **b** (produïda en l'instant 01:01) i **c** (produïda en l'instant 23:59), s'emmagatzemaran en **bBoard**[0], **bBoard**[61] i **bBoard**[1439], respectivament, com es pot veure en la Figura 1 en la que es representa l'objecte **bB** de tipus **BulletinBoard**.
- **numPerType**: atribut d'instància privat de tipus array d'`int` i grandària 3, tal que **numPerType**[*i*] emmagatzema el número de notícies de tipus *i*, $0 \leq i < 3$, que hi ha en **bBoard**; així, **numPerType**[0], **numPerType**[1] i **numPerType**[2] són, respectivament, el número de notícies de tipus **AUDIO**, **VIDEO** i **TEXT** que hi ha en **bBoard**. Si les notícies de l'exemple anterior **a** i **c** són de tipus **AUDIO** i **b** és de tipus **TEXT**, aleshores **numPerType**[0], **numPerType**[1] i **numPerType**[2] valen 2, 0 i 1, respectivament, com en la Figura 1.

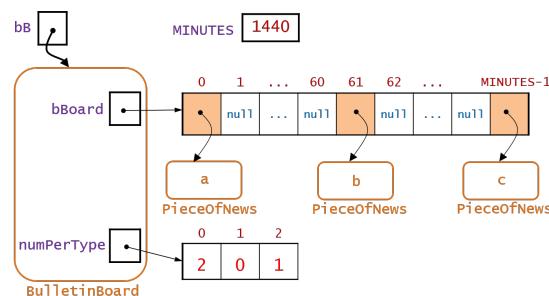


Figura 1: Representació d'un objecte de tipus **BulletinBoard**.

b) (0.75 punts) Mètode constructor, que crea un `BulletinBoard` sense notícies.

c) (1.5 punts) Mètode `add` amb la següent capçalera o perfil:

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

que, donada una `PieceOfNews p`, l'afegeix al tauler en el seu instant de temps. Si ja hi ha una notícia en el mateix instant de temps, quedarà la més popular segons el mètode `compareTo`. Si el tauler s'ha modificat, ha de tornar `true`; en cas contrari, `false`. Aquest mètode ha d'actualitzar l'atribut `numPerType` quan siga procedent.

Recorda que la classe `TimeInstant` té el mètode `toMinutes()` que torna el nombre de minuts transcorreguts des de les 00:00 fins a l'instant representat per l'objecte en curs.

d) (1.25 punts) Mètode `isPosted` amb la següent capçalera o perfil:

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

que torna la notícia l'enllaç de la qual siga `link`. En cas de no trobar-la, ha de tornar `null`.

e) (1.5 punts) Mètode `filterByType` amb la següent capçalera o perfil:

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

que torna un array amb totes les notícies produïdes en el dia de tipus `type`, sent $0 \leq \text{type} < 3$. El tamany de l'array retornat ha de ser igual al número de notícies produïdes en el dia de tipus `type`, sent 0 si no s'ha produït cap notícia d'aquest tipus.

Solució:

```
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;
    }

    /** Precondició: 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 punts **Es demana:** escriure un mètode públic i estàtic tal que, donat un array d'enters positius, cerque i torne el primer valor de l'array que estiga repetit en qualsevol altra posició. En cas que no hi haja repetits, el mètode tornarà -1. Per exemple, per a l'array **a1**, el mètode tornarà 2 (ja que el valor 2 és el primer que està repetit) i per a l'array **a2**, la resposta serà -1 (ja que no hi ha repetits).

	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

Solució:

```
/** Precondició: v és un array d'enters positius */
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]; } // desfem l'ultim increment
    else { return -1; }
}
```

3. 2.25 punts Donat un valor real x i la següent recurrència:

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

Es demana: escriure un mètode públic i estàtic tal que, donats dos valors **x** i **epsilon** ($0 \leq \text{epsilon} < 1$) de tipus **double**:

- si $x \leq 0$, torne -1;
- si no, torne el número de termes calculats per a que l'expressió $1/(C_n - C_{n-1}) \leq \text{epsilon}$ siga certa.

Solució:

```
/** Precondició: 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;
}
```