

Tercer parcial de PRG - ETSInf  
Data: 8 de juny de 2011. Duració: 2 hores.

1. 3 punts Prenent en consideració la definició de `LlistaIntEnla` vista en classe,

**Es demana:** fer un mètode:

```
public void inserirSenseRepetits(int val, boolean davant)
```

per a inserir un element `val` en la `LlistaIntEnla` en el cas de que no existira prèviament en ella, bé al començament de la llista, com a primer element, si l'argument `davant` és `true`, bé per la part de darrere, a la fi de la llista, cas de que l'argument `davant` siga `false`. La `LlistaIntEnla` romandrà sense canvi en el cas de que l'element `val` ja existira inicialment en ella. **NOTA:** aquesta operació haurà de tindre un **cost lineal**.

**Solució:**

```
/** Insereix un nou element sols si no existeix(sense duplicats).
 * Be al començament (davant es true), be a la fi (davant es false)
 */
public void inserirSenseRepetits(int val, boolean davant) {
    NodeInt p = primer;
    NodeInt q = null;
    while (p!=null && p.dada != val) {
        q = p; p = p.seguint;
    }
    if (p==null) {
        talla++;
        if (q==null) primer = new NodoInt(val);
        else if (davant) primer = new NodoInt(val,primer);
        else q.seguint = new NodoInt(val);
    }
}
```

2. 2 punts Suposant ja fetes les classes:

- `NodeStr`, amb atributs `dada` de tipus `String` i `seguint` de tipus `NodeStr`, i amb les dues operacions constructores habituals definides en aquests tipus de classes.
- `LlistaStrEnla`, mitjançant la qual es manté una llista enllaçada d'elements de tipus `NodeStr` i amb totes les operacions definides en classe per a les llistes.

**Es demana:** Fer una operació, dintre de la classe `LlistaStrEnla`, que determine si una llista d'eixe tipus està o no ordenada ascendentment. Una llista de paraules tal com una `LlistaStrEnla`, està ordenada ascendentment si per a qualsevol parella d'elements consecutius de la llista, el primer element és anterior o igual al segon. A més, per definició, una llista buida o amb tan sols un element, està ordenada. **NOTA:** aquesta operació haurà de tindre un **cost lineal**.

### Solució:

```
/** True sii la llista esta ordenada ascendentment */
public boolean estaOrdenat() {
    if (talla==0 || talla==1) return true;
    else { NodeStr p = primer.seguint;
        NodeStr q = primer;
        while (p!=null && q.dada.compareTo(p.dada)<=0) {
            q = p; p = p.seguint;
        }
        return p==null;
    }
}
```

3. 3 punts Suposant ja definida la classe `NodeInt`, vista en classe, escriure una classe `ExamenIntEnla` amb les operacions següents:

- `ExamenIntEnla()`, constructora de l'estructura buida.
- `inserirAlPrincipi(int)`, insereix al començament de l'estructura el valor que es rep com a argument.
- `inserirAlFi(int)`, insereix al final de l'estructura el valor que es rep com a argument.

**Es demana:** Implementar la classe `ExamenIntEnla`, declarant per a ella el(s) atribut(s) necessari(s), i implementant totes les operacions demanades. **NOTA:** totes les operacions hauran de tindre un **cost constant**.

### Solució:

```
package lineals;

/**
 * @author (PRG - ETSINF - DSIC)
 * @version (examen juny 2011)
 */
public class ExamenIntEnla {
    private NodeInt ini;
    private NodeInt ult;

    public ExamenIntEnla() { ini = ult = null; }

    public void inserirAlPrincipi(int x) {
        if (ini==null) ult = ini = new NodeInt(x);
        else ini = new NodeInt(x,ini);
    }

    public void inserirAlFi(int x) {
        if (ini==null) ult = ini = new NodeInt(x);
        else ult = ult.seguint = new NodeInt(x);
    }
}
```

4. 2 punts Partint de la classe `LlistaIntEnla`, i fent-la servir exclusivament, es desitja fer una nova classe `PilaIntEnla2` que emprant les operacions de `LlistaIntEnla` implemente totes les operacions d'una Pila, aixó és, les següents:

- `PilaIntEnla2()`, constructora.
- `talla()`, torna la talla.
- `esBuida()`, diu si està buida.
- `cim()`, torna, si existeix, el cim.
- `empilar(int)`, empila l'element que se li dona.
- `desempilar()`, desempila i torna, si existeix, l'element al cim.

**Es demana:** Implementar, fent servir `LlistaIntEnla`, la classe `PilaIntEnla2`, declarant per a ella **un únic atribut** de tipus `LlistaIntEnla`, i implementant totes les operacions demanades. **NOTA:** Totes les operacions hauran de tindre un **cost constant**.

#### Solució:

```
package lineals;

/**
 * PilaIntEnla2 implementa una PilaInt mitjancant una LlistaIntEnla
 * @author (PRG - ETSINF - DSIC)
 * @version (examen juny 2011)
 */
public class PilaIntEnla2 {
    private LlistaIntEnla laLlista;

    public PilaIntEnla2() {
        laLlista = new LlistaIntEnla();
    }

    public int talla() { return laLlista.talla(); }
    public boolean esBuida() { return laLlista.esBuida(); }
    public int cim() { return laLlista.recuperar(0); }
    public void empilar(int x) { laLlista.inserir(0,x); }
    public int desempilar() { return laLlista.eliminar(0); }
}
```