

Parcial 2 - Teoria - PRG - ETSInf - Curs 2012-13.

7 de juny de 2013. Duració: 2 hores.

1. **2 punts** Suposant ja implementada una excepció d'usuari anomenada `PilaBuidaException` que deriva de la classe `Exception`, és possible definir la classe `PilaIntEnla`, perquè quan s'executen els mètodes que no poden operar sobre una Pila buida llancen l'excepció assenyalada (en lloc d'indicar que és una precondició d'execució dels mateixos).

Es demana: Implementa els mètodes de la classe `PilaIntEnla` que necessiten operar sobre una Pila no buida per tal que quan s'executen sobre una Pila buida llancen l'excepció `PilaBuidaException`.

Solució:

```
/** Desempila i torna el valor en el cim de la Pila */
public int desempilar() throws PilaBuidaException {
    if (talla==0) throw new PilaBuidaException("Pila buida");
    else {
        int x = this.cim.dada;
        this.cim = this.cim.seguint;
        this.talla--;
        return x;
    }
}

/** Torna el valor en el cim de la Pila */
public int cim() throws PilaBuidaException {
    if (talla==0) throw new PilaBuidaException("Pila buida");
    return this.cim.dada;
}
```

2. **2.5 punts** Donada una seqüència enllaçada d'enters `seq` i un enter `x`, **es demana** un mètode amb el següent perfil:

```
public static int ultimaAparicioDe(NodeInt seq, int x)
```

que torne la posició de l'última aparició de `x` en la seqüència i -1 si no està, entenent 0 com la primera posició.

Solució:

```
public static int ultimaAparicioDe(NodeInt seq, int x) {
    NodeInt actual = seq;
    int comptador = 0, ultimaAparicio = -1;
    while(actual!=null) {
        if (actual.dada==x) ultimaAparicio = comptador;
        comptador++;
        actual = actual.seguint;
    }
    return ultimaAparicio;
}
```

3. 2.5 punts Donada una pila d'enters no buida, **es demana** un mètode **recursiu** amb el següent perfil:

```
public static void esborraBase(PilaIntEnla p)
```

que la modifique eliminant l'element de la seua base (l'element més antic de la pila).

Solució:

```
/** p és una Pila no buida */
public static void esborraBase(PilaIntEnla p) {
    if (p.talla()>1) {
        int x = p.desempilar();
        esborraBase(p);
        p.empilar(x);
    }
    else p.desempilar();
}
```

4. 3 punts Donades dues llistes amb punt d'interés d'enters, llista1 i llista2, ambdues amb els seus elements en ordre estrictament creixent, **es demana** un mètode amb el següent perfil:

```
/** llista1, llista2 estan en ordre estrictament creixent */
public static LlistaPIIntEnla unio(LlistaPIIntEnla llista1, LlistaPIIntEnla llista2)
```

que calcule la unió de les dues llistes. La llista resultant també haurà de quedar en ordre estrictament creixent.

Exemple:

Siga una llista1 amb els valors: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19

Siga una llista2 amb els valors: 1, 4, 7, 10, 13, 16, 19, 21, 27

El resultat de unio(llista1,llista2) ha de ser una llista amb els valors:

1, 3, 4, 5, 7, 9, 10, 11, 13, 15, 16, 17, 19, 21, 27

Solució:

```
/** llista1, llista2 estan en ordre estrictament creixent */
public static LlistaPIIntEnla unio(LlistaPIIntEnla llista1, LlistaPIIntEnla llista2){
    LlistaPIIntEnla li = new LlistaPIIntEnla();
    llista1.inici(); llista2.inici();
    while(!llista1.esFi() && !llista2.esFi()) {
        int i = llista1.recuperar(), j = llista2.recuperar();
        if (i<j) { li.inserir(i); llista1.seguint(); }
        else if (i>j) { li.inserir(j); llista2.seguint(); }
        else { li.inserir(i); llista1.seguint(); llista2.seguint(); }
    }

    while(!llista1.esFi()) {
        li.inserir(llista1.recuperar());
        llista1.seguint();
    }
}
```

```
}

while(!llista2.esFi()) {
    li.inserir(llista2.recuperar());
    llista2.seguint();
}
return li;
}
```