

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

17 de juny de 2013. Duració: 1 hora i 50 minuts.

1. 2.5 punts Donats el nom d'un fitxer de text `nomFitx` i una paraula `par`, ambdós de tipus `String`, es **demana** implementar un mètode estàtic que copie totes les línies que contenen aquesta paraula en un fitxer de text anomenat *result.txt* al sistema, precedides pel número de línia que ocupen en el fitxer `nomFitx`. El mètode **ha de capturar** l'excepció `FileNotFoundException`, escrivint un missatge en l'eixida estàndard en el cas que aquesta excepció ocorregi.

NOTA: En la resolució d'aquest exercici es pot usar el mètode `contains` de la classe `String` tal que, essent `s1` i `s2` de tipus `String`, `s1.contains(s2)` torna `true` si `s1` conté `s2` com subcadena i, en cas contrari, torna `false`.

Exemple: amb el següent fitxer d'entrada i la paraula `la`:

```
Justa la justa
justícia justa.
Justa la fusta
que ens justifica.
```

El fitxer d'eixida (*result.txt*) seria:

```
1 Justa la justa
3 Justa la fusta
```

Solució:

```
import java.io.*;
import java.util.*;
public class Exercici1 {
    public static void copiar(String nomFitx, String par) {
        try {
            Scanner s = new Scanner(new File(nomFitx));
            PrintWriter pw = new PrintWriter(new File("result.txt"));
            int cont = 0;
            while(s.hasNext()) {
                String linea = s.nextLine();
                cont++;
                if (linea.contains(par)) pw.println(cont + " " + linea);
            }
            s.close();
            pw.close();
        } catch (FileNotFoundException e) {
            System.out.println("Fitxer no trobat.");
        }
    }
}
```

2. 2.5 punts Siga la classe `LlistaPIIntEnlaOrd` una classe molt pareguda a la classe `LlistaPIIntEnla` presentada en el tema 5. En les llistes de la classe `LlistaPIIntEnlaOrd` els valors enters estan en ordre estrictament creixent. El constructor i tots els mètodes d'aquesta classe (excepte `inserir`) tenen la mateixa funcionalitat que en la classe `LlistaPIIntEnla`.

Es demana: implementar, **sense fer ús dels mètodes de la classe (usant només referències)**, el mètode:

```
public void inserir(int x)
```

per tal que inserisca `x` ordenadament en la llista. El punt d'interès es situarà a la dreta de l'element que s'insereix.

Solució:

```
public void inserir(int x) {
    this.PI = this.primer; this.antPI = null;
    while(this.PI!=null && this.PI.dada<x) {
        this.antPI = this.PI;
        this.PI = this.PI.seguint;
    }
    if (this.PI==null || this.PI.dada>x) {
        NodeInt nou = new NodeInt(x,this.PI);
        if (this.PI==this.primer) this.primer = nou;
        else this.antPI.seguint = nou;
        this.antPI = nou;
        this.talla++;
    }
}
```

3. 2.5 punts Donada una llista amb punt d'interès `LlistaPIIntEnla l` i un enter `x`, **es demana** implementar un mètode estàtic **esborrar** que torne una nova `LlistaPIIntEnla` amb els elements de la llista `l` iguals a `x`. A més, haurà d'eliminar aquests valors de la llista `l`.

Per exemple, si la llista `l` conté els valors 2, 3, 4, 3, 7, 5 i 3, després de l'execució de la crida `LlistaPIIntEnla lis = esborrar(l,3)`, la llista `lis` conté els valors 3, 3 i 3 i la llista `l` conté els valors 2, 4, 7 i 5.

S'usaran únicament els mètodes públics de la classe `LlistaPIIntEnla`.

Solució:

```
public static LlistaPIIntEnla esborrar(LlistaPIIntEnla l, int x) {
    LlistaPIIntEnla llista = new LlistaPIIntEnla();
    l.inici();
    while(!l.esFi()) {
        if (l.recuperar()==x) {
            llista.inserir(x) ;
            l.eliminar();
        }
        else l.seguint();
    }
    return llista;
}
```

4. 2.5 punts Donada una `PilaIntEnla p` i un enter `x`, es **demana** implementar un mètode estàtic **recursiu** amb el següent perfil:

```
public static void eliminarMenorsQue(PilaIntEnla p, int x)
```

que elimine de `p` els elements menors que `x`. Per exemple, si la pila `p` conté els valors: 3, 6, 7, 2, 5 i 4, després de l'execució de la crida `eliminarMenorsQue(p,5)`, la pila conté els valors 6, 7 i 5, tal i com es mostra a continuació:

p abans de l'execució		p després de l'execució
3		
6		
7		
2	==== eliminarMenorsQue(p,5) ====>	6
5		7
4		5
---		---

S'usaran únicament els mètodes públics de la classe `PilaIntEnla`.

Solució:

```
private static void eliminarMenorsQue(PilaIntEnla p, int x) {
    if (!p.esBuida()) {
        int aux = p.desempilar();
        eliminarMenorsQue(p,x);
        if (aux>=x) p.empilar(aux);
    }
}
```