

PRG - ETSInf. TEORIA. Curs 2013-14. Recuperació Parcial 2.
23 de juny de 2014. Duración: 1h 50m.

1. **2.5 punts** Donats un `String nomFitx` amb el nom d'un fitxer de text i certa paraula `pal` (un paraula és un *token* o successió de caràcters no separats per blancs Java), realitzar un mètode que determine si `pal` apareix en el fitxer. En el mètode que es realitze haurà de tractar-se la possible excepció `FileNotFoundException`, comunicant la no existència del fitxer a l'usuari mitjançant un missatge en l'eixida estàndard i retornant `false`.

Solució:

```
public static boolean estaEn(String nomFitx, String pal) {
    try {
        Scanner entrada = new Scanner(new File(nomFitx));
        boolean trobat = false;
        while (entrada.hasNext() && !trobat) {
            String p = entrada.next();
            if (p.equals(pal)) trobat = true;
        }
        entrada.close();
        return trobat;
    } catch (FileNotFoundException e) {
        System.out.println("Fitxer " + nomFitx + " no trobat");
        return false;
    }
}
```

2. **2.5 punts** Considerar la següent classe `NodePersona`:

```
class NodePersona {
    int dni;
    String nom;
    NodePersona seguent;

    NodePersona(int i, String s, NodePersona n) {
        dni = i;
        nom = s;
        seguent = n;
    }
}
```

Es demana implementar els següents mètodes estàtics en la classe `NodePersona`:

- (1.25 punts) Un mètode anomenat `comptar` tal que, donada una seqüència de `NodePersona` i un `String`, torne el número de nodes en la seqüència tals que l'atribut `nom` del node continga l'`String`. En la classe `String` hi ha un mètode, amb perfil boolean `contains(String s)`, que comprova si l'`String` que l'invoca conté a `s`.
- (1.25 punts) Un mètode anomenat `cercar` tal que, donada una seqüència de `NodePersona` i un `int`, busque si en la seqüència existeix algun node amb atribut `dni` igual a aquest `int`. En cas de trobar el node, torna el valor del seu atribut `nom`; en cas de no trobar-lo, torna la cadena `"Persona desconeguda"`.

Solució:

```
public static int comptar(NodePersona n, String s) {
    int compt = 0;
    NodePersona aux = n;
    while (aux!=null) {
        if (aux.nom.contains(s)) compt++;
        aux = aux.seguint;
    }
    return compt;
}

public static String cercar(NodePersona n, int i) {
    NodePersona aux = n;
    while (aux!=null) {
        if (aux.dni==i) return aux.nom;
        aux = aux.seguint;
    }
    return "Persona desconeguda";
}
```

3. 2.5 punts Es desitja afegir a la classe `LlistaPIIntEnla` un nou mètode d'inserció en el punt d'interès (a més del mètode `inserir(int)` ja definit en la mateixa) anomenat `inserirSenseRepetir`, el qual realitzi la inserció d'un element només si no es troba en la llista; en cas contrari ha de llançar una excepció `IllegalArgumentException` (predefinida de Java i derivada de `RuntimeException`), amb el missatge "Ja està l'element " seguit de l'enter que s'ha tractat d'inserir. Per exemple:

- Si la llista és 9 1 4 2 9 i es vol inserir el nou element 0, la llista ha de quedar 9 1 0 4 2 9.
- Si la llista és 9 1 4 2 9 i es vol inserir el nou element 2, la llista ha de quedar 9 1 4 2 9 i s'ha de llançar l'excepció amb el missatge: Ja està l'element 2.

Es demana:

1. (2.25 punts) Escriure el mètode `inserirSenseRepetir` segons la descripció anterior. Notar que si es verifica que `e` no existeix prèviament en la llista en ninguna posició, es pot usar el mètode `inserir` de la classe.

Solució: Solució d'entre les moltes possibles:

```
public void inserirSenseRepetir(int e) {
    NodeInt aux = primer;
    while (aux!=null && aux.dada!=e) aux = aux.seguint;
    if (aux!=null) throw new IllegalArgumentException("Ja està l'element " + e);
    this.inserir(e);
}
```

2. (0.25 punts) Si aquest mètode s'invocara en un mètode `main`, seria obligatori escriure la crida al mètode dins d'una instrucció `try/catch`, o en el seu defecte modificar la capçalera de `main`? Raonar la resposta.

Solució: No, perquè no llança cap excepció *checked*.

4. 2.5 punts Donada una classe **Examen**, es desitja escriure en ella un mètode per a obtenir el valor major d'una **CuaIntEnla** **c**. En acabar l'operació, **c** ha de ser igual que ho era inicialment. Si la cua està buida ha de llançar l'excepció **NoSuchElementException** amb el missatge "Cua buida: màxim no definit". Per exemple:
- Si **c** és `<- 4 -2 9 8 <-`, ha de tornar 9 i deixar **c** com `<- 4 -2 9 8 <-`.
 - Si **c** és `<- -2 <-`, ha de tornar -2 i deixar **c** com `<- -2 <-`.
 - Si **c** és `<- <-`, **c** ha de seguir sent buida, i es llança l'excepció.

Solució:

```
public static int maxim(CuaIntEnla c) {
    int n = c.talla();
    if (n==0) throw new NoSuchElementException("Cua buida: màxim no definit");
    int e = c.desencuar();
    int maxim = e;
    c.encuar(e); n--;
    while (n>0) {
        e = c.desencuar();
        if (e>maxim) maxim = e;
        c.encuar(e); n--;
    }
    return maxim;
}
```