

Parcial 2 - Pràctiques - PRG - ETSInf - Curs 2013/14  
23 de juny de 2014 - Duració: 50 minuts

1. **2.5 punts** El mètode `llegirParaulaDesDe`, que figura a continuació, llig una cadena de caràcters i un enter des d'un `Scanner` i torna la subcadena a partir de la posició indicada per l'enter llegit. En invocar al mètode `substring(int)` pot donar-se l'excepció `IndexOutOfBoundsException` si l'enter és negatiu o més gran que la longitud de la cadena llegida.

```
public static String llegirParaulaDesDe(Scanner t) {
    System.out.print("Introdueix una paraula: ");
    String linia = t.nextLine();
    System.out.println("Introdueix una posició: ");
    int pos = t.nextInt();
    t.nextLine();
    String res = linia.substring(pos);
    return res;
}

public static void main(String[] args) {
    Scanner tec = new Scanner(System.in);
    String paraula = llegirParaulaDesDe(tec);
    System.out.println("La paraula llegida és: " + paraula);
}
```

**Es demana:** Modificar el mètode `main` perquè **capture** l'excepció `IndexOutOfBoundsException` i torne a demanar les dades, executant `llegirParaulaDesDe` totes les vegades necessàries, fins que aquesta excepció no es produïska.

**Solució:**

```
public static void main(String[] args) {
    Scanner tec = new Scanner(System.in);
    boolean hiHaError = true;
    do {
        try {
            String paraula = llegirParaulaDesDe(tec);
            System.out.println("La paraula llegida és: " + paraula);
            hiHaError = false;
        } catch (IndexOutOfBoundsException e) {
            System.out.println("Asegura't d'introduir una posició vàlida!");
        }
    } while (hiHaError);
}
```

2. **2.5 punts** **Es demana:** Escriure un mètode que, donat un valor de tipus `double` que representa un saldo i un `Scanner` d'un fitxer de text amb la informació dels comptes d'un banc, lligisca des del `Scanner` i escriga en un fitxer de text "*comptes.txt*" els números de compte amb saldo major que el donat.

Cada línia del fitxer d'on llegir té dos elements d'informació, un número de compte de tipus `int` seguit d'un saldo de tipus `double`. La capçalera del mètode és la que figura a continuació. **No s'ha de tractar cap excepció**, ja que es propaga qualsevol excepció que es pugui produir.

```
public static void triaComptes(double s, Scanner in) throws Exception
```

**Solució:**

```

public static void triaComptes(double s, Scanner in) throws Exception {
    PrintWriter out = new PrintWriter("comptes.txt");
    while(in.hasNext()) {
        int numCompte = in.nextInt();
        double saldo = in.nextDouble();
        if (saldo>s) out.println(numCompte);
    }
    out.close();
}

```

3. **2.5 punts** Donades les estructures de dades **Concordanca** i **NodeCnc**, com les vistes en pràctiques, amb atributs segons les declaracions següents:

| Concordanca                | NodeCnc             |
|----------------------------|---------------------|
| -----                      | -----               |
| private NodeCnc prim;      | String pal;         |
| private int talla;         | CuaIntEnla numLins; |
| private boolean esOrd;     | NodeCnc seguent;    |
| private String separadors; |                     |

**Es demana:** Escriure un mètode dintre de la classe **Concordanca** amb perfil:

```

// PRECONDICIÓ: n >= 1
public int numParaulesMesN(int n)

```

que torne quantes paraules apareixen més de **n** vegades al text amb el que s'ha construït la **Concordanca**.

**Solució:**

```

// PRECONDICIÓ: n >= 1
public int numParaulesMesN(int n) {
    int compt = 0;
    for(NodeCnc aux = prim; aux!=null; aux = aux.seguent)
        if (aux.numLins.talla(>n) compt++;
    return compt;
}

```

4. **2.5 punts** A continuació es mostra una possible implementació incompleta del mètode **insNoOrd(String,int)** de la classe **Concordanca**.

```

private void insNoOrd(String pal, int numLin) {
    NodeCnc aux = prim, ant = null;
    while( /* GUARDA */ ) { ant = aux; aux = aux.seguent; }

    if ( /* CONDICIÓN */ ) // pal ja està en la Concordanca
        /* INSTRUCCIÓN */
    else { // El nou NodeCnc amb la paraula pal ha de ser l'últim i
        // s'ha d'inserir darrere de ant
        NodeCnc nou = new NodeCnc(pal, numLin);
        talla++;
        if (ant==null) // el nou NodeCnc serà el primer i l'últim
            /* INSTRUCCIÓN */
        else // el nou NodeCnc serà l'últim
            /* INSTRUCCIÓN */
    }
}

```

**Es demana:** Completar el codi anterior amb:

- a) [0.6 punts]: la guarda del bucle.

- b) [0.4 punts]: la condició de la instrucció condicional.
- c) [0.7 punts]: la instrucció que falta si la paraula **pal** ja està en la **Concordanca**.
- d) [0.4 punts]: la instrucció que falta si la paraula **pal** ha de ser la primera i l'última de la **Concordanca**.
- e) [0.4 punts]: la instrucció que falta si la paraula **pal** ha de ser l'última de la **Concordanca**.

**Solució:**

```
private void insNoOrd(String pal, int numLin) {  
    NodeCnc aux = prim, ant = null;  
    while(aux!=null && !aux.pal.equals(pal)) { // a)  
        ant = aux;  aux = aux.seguint;  
    }  
  
    if (aux!=null) // b)  
        aux.numLins.encuar(numLin); // c)  
    else {  
        NodeCnc nou = new NodeCnc(pal, numLin);  
        talla++;  
        if (ant==null)  
            prim = nou; // d)  
        else ant.seguint = nou; // e)  
    }  
}
```