

Parcial 2 - PRÀCTIQUES - PRG - ETSInf. Curs 2016-17

5 de juny de 2017. Duració: 1 hora

Nota: L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final de l'assignatura és de 1,2 punts.

NOM:	GRUP DE PRÀCTIQUES:
------	---------------------

1. 5 punts Es desitja modificar la classe **Banc** de la pràctica 4, per a permetre la gestió de remeses de transferències. Una remesa ve en un fitxer de text en el que cada línia conté les dades d'una transferència amb tres valors separats per espais:

```
numCompte1 numCompte2 import
```

on `numCompte1`, `numCompte2` són enters entre 10000 i 99999 que corresponen a números de compte existents al banc, i `import` és un valor real positiu.

És a dir, les dades d'una línia indiquen que es vol fer una transferència d'`import` euros des del compte número `numCompte1` al compte número `numCompte2`.

Es demana: suposant que totes les classes necessàries han sigut importades en la classe **Banc**, implementar un mètode amb perfil

```
public void ferTransf(Scanner remesa)
```

tal que, utilitzant el paràmetre **Scanner** ja inicialitzat, ha de llegir les línies amb les dades de cada transferència del fitxer, retirar l'`import` del compte amb número de compte `numCompte1` i ingressar-lo al compte amb número de compte `numCompte2`.

En cas de qualsevol error en una línia, s'ha d'obviar tota la línia mostrant un missatge que indique el motiu del mateix, i seguir tractant la resta de dades.

Nota: Recordeu que el mètode `retirar(double)` de la classe **Compte** pot llançar una excepció de tipus **IllegalArgumentException** si la quantitat a retirar és major que el saldo del compte, i que els mètodes de lectura del **Scanner** poden llançar una excepció **InputMismatchException** si el tipus del token llegit no coincideix amb l'esperat pel mètode.

Recordeu igualment que el mètode `getCompte(int)` de la classe **Banc** retorna, si existeix, el compte el número del qual es passa com a paràmetre, o null si no existeix –encara que en aquest exercici mai passarà, ja que els números de compte del fitxer són de comptes existents al banc–.

Solució:

```
public void ferTransf(Scanner remesa) {
    int numC1 = 0, numC2 = 0; double import = 0;
    while (remesa.hasNext()) {
        try {
            numC1 = remesa.nextInt();
            numC2 = remesa.nextInt();
            import = remesa.nextDouble();
            Compte c1 = this.getCompte(numC1);
            Compte c2 = this.getCompte(numC2);
            c1.retirar(import);
            c2.ingresar(import);
        } catch (InputMismatchException e) {
            System.err.println("línia errònia");
        } catch (IllegalArgumentException e) {
            System.err.println("import erroni");
        } finally { remesa.nextLine(); }
    }
}
```

2. 5 punts Siga la classe `ConjuntString` de la pràctica 5:

```
public class ConjuntString {  
  
    private NodeString primer;  
    private int talla;  
  
    /** Crea un conjunt buit */  
    public ConjuntString() {  
        this.primer = null;  
        this.talla = 0;  
    }  
  
    ....  
  
}
```

on la seqüència enllaçada `primer` conté els elements que formen part del conjunt; tots els mètodes de la classe s'implementen de manera que, per a qualsevol conjunt de la classe, aquesta seqüència està ordenada ascendentment per l'ordre de `String` i sense elements repetits.

Es demana implementar un nou mètode dins de la classe, amb perfil

```
public ConjuntString subconjuntParaules(char c)
```

que retorne un `ConjuntString` amb tots els elements del conjunt que comencen pel caràcter `c`. S'haurà de procurar que el mètode siga $O(n)$, sent n la talla de `this`.

Recordeu que es pot consultar el caràcter que apareix en la posició i -èsima d'un `String` amb el mètode `charAt(int i)`.

Solució:

```
public ConjuntString subconjuntParaules(char c) {  
    ConjuntString cs = new ConjuntString();  
    NodeString aux = primer;  
    NodeString ultCs = null;  
    boolean fi = false;  
    while (aux != null && !fi) {  
        String p = aux.dada;  
        if (p.charAt(0) > c) { fi = true; }  
        else if (p.charAt(0) == c) {  
            // Inserir al final de cs:  
            NodeString nou = new NodeString(p);  
            if (cs.primer == null) { cs.primer = nou; }  
            else { ultCs.seguint = nou; }  
            ultCs = nou;  
            cs.talla++;  
        }  
        aux = aux.seguint;  
    }  
    return cs;  
}
```