Parcial 2 - PRÀCTIQUES - PRG - ETSInf. Curs 2015-16
31 de maig de 2016. Duració: 1 hora

(Nota: L'examen s'avalúa sobre 10 punts, però el seu pes específic en la nota final de PRG és de 1.2 punts. D'aquesta manera els valors de les preguntes 3, 3 i 4 punts equivalen en la nota final a 0.35, 0.35 i 0.5, respectivament.)

NOM:	GRUP DE PRÀCTIQUES:

1. [3 punts] Implementar un mètode `toArray()` en la classe `CuaApunts` de la pràctica 5 que torne un array amb els apunts de la cua d'apunts.

   Nota: Recordar que els atributs definits en la classe `CuaApunts` són `talla` (de tipus `int`), `primer` i `ultim` (de tipus `NodeApunt`) i els de la classe `NodeApunt` són `dada` (de tipus `Apunt`) i `seguent` (de tipus `NodeApunt`).

   Solució:

```
public Apunt[] toArray() {
    Apunt[] ap = new Apunt[talla];
    NodeApunt aux = primer;
    for (int i = 0; i < talla; i++) {
        ap[i] = aux.dada;
        aux = aux.seguent;
    }
    return ap;
}
```

2. [3 punts] Escriure un mètode amb perfil: `public double promediApunts()`, en la classe `Banc` de la pràctica 5, que obtinga el número promedi d'apunts dels comptes del banc.

   Nota: Recordar que els atributs definits en la classe `Banc` de la pràctica 5 són `primer` (de tipus `NodeCompte`) i `numComptes` (de tipus `int`), i els de la classe `NodeCompte` són `dada` (de tipus `CompteAp`) i `seguent` (de tipus `NodeCompte`). En la classe `CompteAp` es defineix el mètode `getNumApunts()` que torna el número d'apunts del compte.

   Solució:

```
public double promediApunts() {
    double promedi = 0;
    if (numComptes != 0) {
        NodeCompte aux = primer;
        while (aux != null) {
            promedi += aux.dada.getNumApunts();
            aux = aux.seguent;
        }
        promedi = promedi / numComptes;
    }
    return promedi;
}
```

3. [4 punts] Es desitja modificar la classe `Banc` per a permetre la gestió de remeses de rebuts. Cada remesa ve en un fitxer de text on cada línia té tres valors separats per espais: `numCompte` que és un enter entre 10000 i 90000, `import` que és un valor real, i `numRebut` que és un enter llarg el rang de valors del qual està comprès entre les constants `MIN_NUM_REBUT` i `MAX_NUM_REBUT` ambdues incloses.

Es demana: assumint que totes les classes necessàries han sigut importades en la classe `Banc`, implementar un mètode amb perfil: `public String gestionarRemesa(Scanner remesa)` tal que, utilitzant el paràmetre `Scanner` ja inicialitzat, ha de llegir les línies amb les dades de cada rebut del fitxer, validar les dades i realitzar el càrrec, si és possible, en el compte corresponent, retornant un `String` amb els números de rebuts (`numRebut`) ben processats separats per salts de línia. En cas de qualsevol error s'ha d'obviar tota la línia i mostrar un missatge indicant el motiu del mateix. El `String` resultat només ha de contindre els números de rebuts que sí s'han pogut processar adequadament, açò és: són vàlids, els comptes dels quals existeixen i no han donat error en efectuar-se el càrrec. En cas que el fitxer no continga cap rebut vàlid, ha de retornar un `String` buit.

Nota: Recordar que el mètode `getCompte(int)` de la classe `Banc` de la pràctica 4 retorna, si existeix, el compte el número del qual es passa com a paràmetre o `null` si no existeix. El mètode `retirar(double)` de la classe `Compte` pot llançar l'excepció `IllegalArgumentException` si la quantitat a retirar és major que el saldo del compte, i els mètodes de lectura del `Scanner`, també poden llançar una excepció de tipus `InputMismatchException` si el tipus del token llegit del fitxer no coincideix amb l'esperat pel mètode.

Solució:

```
public String gestionarRemesa(Scanner remesa) {
    String res = "";
    int numC = 0;  double import = 0;  long numR = 0;
    while (remesa.hasNext()) {
        try {
            numC = remesa.nextInt();
            import = remesa.nextDouble();
            numR = remesa.nextLong();
            Compte c = this.getCompte(numC);
            if (c != null) {
                if (numR <= MAX_NUM_REBUT &&  numR >= MIN_NUM_REBUT ) {
                    c.retirar(import);
                    res += numR + "\n";
                } else { System.err.println("numRebut erroni"); }
            } else { System.err.println("compte erroni"); }
        } catch (InputMismatchException e)   {
            System.err.println("línia errònia");
        } catch (IllegalArgumentException e) {
            System.err.println("import erroni");
        } finally { remesa.nextLine(); }
    }
    return res;
}
```