

# Fent el pas de Programació I a Programació II

## 1. Objectius

L'objectiu principal d'aquest problema és fer una **transició** de Programació I a Programació II.

L'assignatura de Programació I ha **assentat les bases** que us permeten llegir i escriure codi (sobretot, en JAVA), i transformar un problema real en un conjunt d'accions computacionals que el resolen i que l'ordinador pot entendre i processar (en temps finit).

L'assignatura de Programació II **reforça i amplia** aquestes habilitats centrant-se en el paradigma de la **Programació Orientada a Objectes** (i Esdeveniments).

Per tal de començar a fer la transició, en aquest problema us proposem realitzar una sèrie d'exercicis guiats sobre vectors (*array*) a 'l'estil de Programació I' i a 'l'estil de Programació II', per a que comenceu a veure i apreciar la diferència, sense encara entrar en detalls que més endavant veureu i entendreu.

Els exercicis 1, 2 i 4 estan resolts i són per comentar i analitzar a la sessió de problemes. L'exercici 3 és un 'repte' que us plantegem.

## 2. El problema

El problema a solucionar és el següent: **donat un vector (*array*) de números enters, esborrar totes les entrades que es corresponen a un nombre concret.**

¿Perquè aquest problema? Perquè és un problema força comú, i rellevant al món de la informàtica, per exemple, quan estem programant algorismes, i perquè ens serveix per l'objectiu del problema.

Treballarem amb el següent vector (ho farem senzill, de moment) i les posicions que hem d'eliminar (per simplificar, les posarem amb valor -1) són les que tenen el nombre 5:

1	2	5	3	5
---	---	---	---	---

Comparat amb els exercicis que heu fet a Programació I, sembla que aquest problema és força fàcil, oi? Com veuràs en els següents exercicis, aquest problema presenta una sèrie de característiques força interessants.

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. Universitat de Barcelona.

### 3. Exercici 1

Abans d'escriure aquest codi en el Netbeans, pots dir si esborra totes les entrades amb 5, i perquè?

```
public static void falseRemove5(){
    int size = 5;
    int[] myArray = new int[size];
    myArray[0] = 1;
    myArray[1] = 2;
    myArray[2] = 5;
    myArray[3] = 3;
    myArray[4] = 5;

    for (int i = 0; i < size; i++){
        if (myArray[i] == 5){
            myArray[i] = -1;
            size--;
        }
    }

    for (int j = 0; j < myArray.length; j++){
        System.out.println(myArray[j]);
    }
}
```

Ja tens la resposta? Aquest codi no esborra totes les entrades amb 5 perquè el bucle del primer *for* itera sobre la mida del vector, i cada cop que esborrem un 5, reduïm la mida del vector, simulant la seva eliminació física. Per tant, tenim una entrada (la darrera) que no s'elimina.

Aquesta estratègia no sembla exitosa, provem una altra manera.

### 4. Exercici 2

I si en comptes de treballar amb la mida del vector, ho fem amb la seva *length*?

Recorda que *length* és una propietat dels *arrays* a JAVA i que no canvia. És a dir, en el moment de crear el vector, definim la seva longitud, i aquesta es manté constant, independentment del nombre de posicions no buides (inicialment, un vector és una llista de 0s<sup>1</sup>) que tingui.

---

<sup>1</sup> Si es crea com atribut de la classe; si es crea com a variable local a un mètode, no està inicialitzat. Pots fer la prova si vols.

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. Universitat de Barcelona.

Escriu el següent codi al Netbeans i observa el resultat:

```
public static void dirtyRemove5Length(){

    int size = 5;
    int[] myArray = new int[size];

    myArray[0] = 1;
    myArray[1] = 2;
    myArray[2] = 5;
    myArray[3] = 3;
    myArray[4] = 5;

    for (int i = 0; i < myArray.length; i++){
        if (myArray[i] == 5)
            myArray[i] = -1;
    }

    for (int j = 0; j < myArray.length; j++){
        System.out.println(myArray[j]);
    }
}
```

Si executes aquest codi, aquesta vegada sí que esborra totes les entrades amb el nombre 5. D'altra banda, és un esborrament brut, perquè tenim posicions buides al vector<sup>2</sup>. Això planteja un problema greu, sobretot quan estem processant vectors en algorismes – suposem que les posicions tenen valors, no que estan buides! Millor doncs si no tenim posicions buides al mig del vector.

## 5. Exercici 3

Ara l'exercici és esborrar però fent que no hi hagi posicions buides al mig del vector. Se t'acudeix com es pot fer? Potser fent un intercanvi de posicions... Potser cal crear dos vectors...

Carai, com pots veure, el problema no és tan fàcil com semblava, no hi ha una altra manera de fer-ho?

---

<sup>2</sup> També tenim el fet de que realment no redueix la mida del vector tot i que estem esborrant, però això ho podem ignorar de moment, estem a l'inici del curs!

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. Universitat de Barcelona.

### 6. Exercici 4

Als anteriors exercicis hem treballat amb els vectors i l'operació d'eliminació a l'estil de Programació I'. En aquest exercici, ho farem a l'estil de Programació II.

Executa el següent codi al Netbeans:

```
public static void cleanRemove5(){

    ArrayList myArrayList = new ArrayList(5);
    int aux;

    myArrayList.add(1);
    myArrayList.add(2);
    myArrayList.add(5);
    myArrayList.add(3);
    myArrayList.add(5);

    Iterator it = myArrayList.iterator();

    while (it.hasNext()){
        aux = (int) it.next();
        if (aux == 5)
            it.remove();
    }

    System.out.println(myArrayList.toString());
}
```

Aquest codi esborra les posicions del vector amb un 5 i a més redueix la mida del vector. Bingo!

El codi treballa amb *ArrayList*, que és una classe (tipus) de Java per vectors dinàmics – poden créixer i decreixer dinàmicament.

Per recórrer el vector, utilitzem un *Iterator*, que és una classe (tipus) de JAVA que es pot entendre com una sèrie d'apuntadors a les entrades del vector, i ens permet esborrar elements d'un vector mentre fem un recorregut o una cerca dinàmicament.

