



**L'Institut International de Technologie Sfax (IIT)**

**Année Universitaire : 2020-2021**

**Module : POO**

**Responsable du cours : Fairouz Fakhfakh**

**Série N° : 4**

## **Exercice n°1**

On considère la définition de la classe suivante :

```
public class B{  
    int c;  
    B succ()  
    {    c+= 1;  
        return this;  
    }  
    int twice(){ return  c*2; }  
  
    void fill(int[] a)  
    {  
        for(int i=0; i<a.length;i++)  
            a[i]= c;  
    }  
    void fill(int k)  
    { k= c;  }  
  
    public static void main(String []args){  
        B x= new B();  
        System.out.println(x.c);  
        System.out.println(x.succ().c);  
        System.out.println(x.c);  
        System.out.println(x.twice());  
        System.out.println(x.c);  
        System.out.println(x.succ().succ().succ().twice());  
        System.out.println(x.c);  
        x.c=22;    int[] a= {7,8};  
        x.fill(a);  
        System.out.println(a[1]);  
        int z= 5;  
        x.fill(z);  
  
        System.out.println(z);  
    }  
}
```

## Exercice n°2

1. Ecrire une classe **Repertoire** permettant de gérer un répertoire téléphonique associant un numéro de téléphone (chaîne de caractères) à un nom. Pour faciliter les choses, on prévoira une classe **Abonne** destinée à représenter un abonné et disposant des fonctionnalités indispensables (nom de l'abonné et son numéro de téléphone).

- La classe Repertoire a comme attributs nbAbonnes le nombre d'abonnés dans le répertoire, le nbMax nombre maximal d'abonnés que le répertoire peut contenir, et la liste des abonnés.
- La classe Repertoire devra disposer des méthodes suivantes :
  - constructeur recevant un argument de type entier précisant le nombre maximum d'abonnés,
  - méthode getNbAbonnes qui fournit le nombre d'abonnés figurant dans le répertoire,
  - méthode getAbonne fournissant l'abonné dont le rang est fourni en argument,
  - méthode addAbonne permettant d'ajouter un nouvel abonné ; elle renverra la valeur false si le répertoire est plein ou la valeur true sinon,
  - méthode getNumero fournissant le numéro associé à un nom d'abonné fourni en argument,
  - méthode getAbonnesTries qui permet de ranger les différents abonnés par ordre alphabétique (pour simplifier, on supposera que les noms sont écrits en minuscules et sans caractères accentués).
- NB: Tous les attributs doivent être déclarés privés.

2. Ecrire la classe **TestRepertoire** qui permet de créer un répertoire téléphonique, d'y ajouter des abonnés et de chercher le numéro de téléphone d'un abonné dont le nom est saisi par le clavier.

## Exercice n°3

Le **Bandit Manchot** est un jeu de hasard que l'on joue avec une machine comportant quatre roues qui tournent et qui comportent chacune un symbole (par exemple abricot, banane, cerise, datte). Quand les roues s'arrêtent, si 4 roues ont un symbole identique on a gagné.

### **On se propose :**

De réaliser ce jeu de hasard en mode texte. On figurera les symboles par un chiffre ou par une lettre. Les gains possibles seront les suivants : 10 € pour deux symboles identiques, 100 € pour 3 symboles identiques, 1000 € pour 4 et 0 € sinon.

1. Créer la classe **Roue** pour le choix des symboles avec les méthodes suivantes:

- public void lance() : Génère un nombre aléatoire et le stocke dans un attribut figurant la valeur affichée sur la roue.

- public int get() : renvoie la valeur stockée.

2. Créer une classe **BanditManchot** qui comporte un tableau de 4 roues, un constructeur pour créer ce tableau et une méthode public int joue() qui affichera les 4 roues et renverra le gain au jeu: 10 € si deux symboles identiques, 100 € si 3 symboles identiques, 1000 € pour 4 symboles sont identiques et 0 € sinon.

3. Ecrire un programme **TestBanditManchot.java** qui contient une méthode main qui permet de tester les classes créées.