

Exercices sur la mémoire 2

Exercice 10.1 Qu'affiche ce programme ?

Question 1

```
public class Mem2_2{
    public static void plusDix(int[] tab, int x){
        x = x+10;
        for (int num=0; num<tab.length; num=num+1){
            tab[num] = tab[num] + 10;
        }
    }
    public static void main(String[] args){
        int[] nombres = {1, 2, 3, 4, 5};
        int nb = 3;
        plusDix(nombres,nb);
        System.out.println(nombres[2]);
        System.out.println(nb);
    }
}
```

Question 2

```
public class Mem2_1{
    public static void exchange(int[] tab1, int[] tab2){
        int[] tamp = tab1;
        tab1 = tab2;
        tab2 = tamp;
    }
    public static void main(String[] args){
        int[] un = {1, 2, 3, 4, 5};
        int[] deux = {10, 20};
        exchange(un, deux);
        System.out.println(un.length);
    }
}
```

Question 3

```
public class Mem2_3{
    public static void exchange(int[][] tab1, int[][] tab2){
        int[] tamp = tab1[0];
        tab1[0] = tab2[0];
        tab2[0] = tamp;
    }
    public static void main(String[] args){
        int[][] un = {{1, 2, 3, 4, 5}};
        int[][] deux = {{10, 20}};
        exchange(un, deux);
        System.out.println(un[0].length);
    }
}
```

Exercice 10.2 Dessins

Question 1

```
public class Mem2_4{
    public static int meth2(int x, int y){
        int res;
        x = x + 1;
        y = y *2;
        // faire le dessin à ce point de l'exécution
        return x + y;
    }
    public static int meth1(int x, int y){
        int res = 12;
        x = meth2(y, x);
        res = x -y;
        return res;
    }
    public static void main(String[] args){
        int x = 3;
        int y = 2;
        System.out.println(meth1(x, y));
    }
}
```

1. Dessinez l'état de la pile au moment marqué par un commentaire dans le code. Notez qu'il n'y a rien d'intéressant dans le tas dans ce programme, aussi n'est-il pas nécessaire de le représenter dans le dessin.
2. Qu'affiche ce programme ?

Question 2

```
public class Mem2_5{
    public static char[] meth2(char[] tab){
        char[] res;
        res = new char[tab.length-1];
        for (int i=0; i< res.length; i = i + 1){
            res[i] = tab[i+1];
        }
        return res;
    }
    public static void meth1(char[] tab){
        tab = meth2(tab);
        // faire le dessin à ce point de l'exécution
        tab[tab.length-1] = tab[0];
    }
    public static void main(String[] args){
        char[] abc = {'a', 'b', 'c'};
        meth1(abc);
        System.out.println(abc);
    }
}
```

1. Dessinez l'état de la mémoire (pile et tas) au moment marqué par un commentaire dans le code.
2. Qu'affiche ce programme ?

Exercice 10.3 écureuil et noisettes

On va représenter un damier où des noisettes sont posées sur des cases (chaque noisette est comme le pion d'un jeu, il ne peut y en avoir qu'une par case). Un écureuil est placé au centre du damier. Puis il se met en mouvement de façon aléatoire et quand il arrive sur une case avec une noisette, il la mange.

La représentation du damier sera un tableau de caractères à deux dimensions avec un espace pour les cases vides, un ô pour une noisette et un 2 pour l'écureuil (voyez-vous le panache de sa queue ?). Le damier aura 11x11 cases. Le centre sera donc en ligne 5, colonne 5 (en commençant à compter à 0, comme en Java).

Question 1 création du damier

Le but est de poser les noisettes au hasard, sur des cases différentes à chaque fois. Pour vous aider, on vous fournit le méthode suivante qui tire au sort un nombre compris entre 0 et 10.

```
public static int auSortOnze(){
    return (int) (Math.random() *11);
}
```

Écrivez une méthode qui renvoie un damier avec un écureuil au centre et 25 noisettes disséminées au hasard. Suggestion : tirez au sort les deux coordonnées d'une case et y placer une noisette si la case n'est pas déjà occupée par une noisette ou l'écureuil.

Question 2 affichage

Écrivez une méthode qui affiche le damier. Pour cela, elle doit parcourir toutes les cases du tableau dans le bon ordre (ligne par ligne).

Question 3 déplacement au hasard de l'écureuil

Écrivez une méthode qui prend en paramètre un damier et change la place de l'écureuil pour une des 4 cases voisines (1 pas horizontalement ou verticalement). Pour cela, vous pourrez utiliser la méthode suivante qui tire au sort un nombre entre 0 et 3.

```
public static int auSortQuatre() {  
    return (int) (Math.random() *4);  
}
```

La méthode doit-elle renvoyer le damier modifié ?

Attention : il faut faire attention à ce que l'écureuil reste bien sur le damier. S'il est sur un bord, une direction est impossible et s'il est dans un coin, il y a deux directions impossibles.

Question 4 méthode main

Écrivez une méthode main qui alterne un déplacement de l'écureuil et une pause d'une seconde. Pour cette pause, vous utiliserez la méthode `Thread.sleep` qui prend en paramètre une durée exprimée en millisecondes (il faut 1000 millisecondes pour faire une seconde). Pour pouvoir utiliser cette méthode, il faut donner comme entête à la méthode main :

```
public static void main(String[] args) throws InterruptedException
```

Question 5 méthode main (version 2)

Modifiez le programme pour qu'il s'arrête quand il n'y a plus de noisette sur le damier.