

Exercice 1 Soit la classe `Point` telle que donnée en cours :

```
class Point {
    public double abscisse;
    public double ordonnee;
    public Point(double a, double b) {abscisse = a; ordonnee = b;}
}
```

Écrire une méthode retournant :

1. un tableau contenant tous les points $p_i = (i, 0)$ pour $i \in [0, 10[$;
2. un tableau contenant tous les points $p_{i,j} = (i, j)$ pour $i \in [0, 10[$, $j \in [0, 2i[$.

Exercice 2 À la manière du cours, faire des schémas pour illustrer le contenu de la mémoire lors de l'exécution de l'extrait de programme suivant (en faisant des hypothèses pour les adresses des variables, en décidant aussi que le contenu de la mémoire est représenté en hexadécimal, et en détaillant les étapes les plus importantes des constructions opérées) :

```
int i=64;
long k=129;
k=i;
```

Idem :

```
int[] t = new int[5];
for (int i=0; i<5; i++)
    t[i] = i*i;
```

Ou encore :

```
int[][] m = new int[3][4];
for (int i=0; i<3; i++)
    for (int j=0; j<4; j++)
        m[i][j] = i*j;
```

Ou enfin :

```
String s = "a";
for (char c='b'; c<'d'; c++)
    s = c+s+c;
```

Exercice 3 Décrire, par des schémas similaires à ceux vus en cours, l'évolution de la pile et du tas lors de l'exécution du programme suivant, à chaque ligne de code, y compris lors de l'appel des méthodes `m1` et `m2`.

```
public static void main(String[] args){
    int m = 18, n = 5;
    int[] tab = new int[2]; tab[0] = n; tab[1] = m;
    int [] t = m1(tab, m, n);
    System.out.println(m + " " + n);
    System.out.println(tab[0] + " " + tab[1]);
    System.out.println(t[0] + " " + t[1]);
}
public static int[] m1(int[] t, int m, int n){
    m++; n++;
    m2(t);
    t = new int[2];
    t[0] = m; t[1] = n; m = t[0]; n = t[1];
    return t;
}
public static void m2(int[] t) {
    t[0] ++; t[1] ++;
}
```