

## TD n° 1 - Rappels

### 1 Syntaxe

```
import java.util.*;

public class Rectangle{

    public static void main(String args[]){
        if (args.length<2){
            System.out.println();
            /*System.out.println("Première ligne écrite");*/
        } else{
            int longueur, largeur;
            longueur = Integer.parseInt(args[0]);
            largeur = Integer.parseInt(args[1]);
            String phrase = "La surface de mon rectangle est";
            System.out.print(phrase+longueur*largeur+"\n");
            if (args.length==3){
                largeur =5; longueur =10;
                System.out.print("Rectification (valeur par défaut) : ");
                System.out.print(phrase+longueur*largeur);
                System.out.println();
            }
        }
    }
}
```

1. Dans quel fichier faut-il sauvegarder ce code ?
2. Quelles sont les deux étapes nécessaires pour exécuter ce programme Java ? Qu'est-ce que le bytecode ? Qu'est-ce que la machine virtuelle Java ? Est-ce un abus de langage que de dire que le code est exécutable ?
3. A quoi sert import ? Pourquoi n'est-il pas nécessaire dans le cas présent ?
4. Où est le commentaire ? Quelle est la seconde façon d'en écrire ?
5. Comment indente-t-on un programme ?
6. A quoi servent les parenthèses après print et println et pourquoi faut-il les écrire même s'il n'y a rien dedans ? Dans quelle librairie Java se trouve la commande println ?
7. Quelle est la différence entre = et == ?
8. Qu'est-ce que contient le tableau args ? Quels sont les résultats des commandes suivantes : java Rectangle 5 et java Rectangle 4 6 8 ?

## 2 Structures de contrôle - Petits programmes

1. Ecrivez un `main` qui contienne 2 variables entières `a` et `b`. Puis écrivez quelques lignes qui permettent d'échanger ces valeurs.
2. Créez un tableau d'entiers de longueur 3, contenant les valeurs 1, 2 et 3.
3. Créez un tableau d'entiers de longueur 300, contenant les valeurs 1,2,3 .... 300
4. Ecrivez une méthode (hors de la méthode `main`) qui prenne en argument un tableau et deux de ses indices, et qui inverse leur contenu. Pourquoi peut-on vous imposer que cette méthode ait un type de retour `void`?
5. Ecrivez une méthode qui n'inverse deux indices d'un tableau que si les valeurs correspondantes ne sont pas dans l'ordre croissant.
6. Ecrivez une méthode qui vous dise si un tableau d'entiers est déjà ordonné de manière croissante.
7. Si l'on veut obtenir un nombre au hasard, `x` dans  $[0..100[$  on peut utiliser la commande `int x = (int)(Math.random() * 100)`. Ecrivez une méthode qui prend en argument un tableau, et tant qu'il n'est pas trié, choisira deux indices au hasard et procèdera à un échange de leurs valeurs.

## 3 Plusieurs tableaux

On travaillera dans cet exercice sur deux tableaux d'entiers `tab1` et `tab2`.

1. Ecrivez une méthode qui nous dise si ces deux tableaux ont les mêmes valeurs au même endroit. Ecrivez une méthode qui nous dise si ces deux tableaux sont les mêmes. Quelle est la différence entre ces deux questions?
2. Ecrivez une méthode qui retourne un troisième tableau, résultat de la mise bout à bout des tableaux `tab1` et `tab2`
3. Ecrivez une méthode qui retourne un nouveau tableau, construit en prenant alternativement, et dans l'ordre les éléments de `tab1` et `tab2`. La construction s'arrête lorsqu'il n'y a plus assez d'éléments dans l'un ou l'autre des tableaux.
4. même question, mais lorsqu'on a alterné les valeurs autant que possible, ce qui reste éventuellement est ajouté d'un trait à la fin.
5. même question, on énumère toujours les éléments de `tab1` et `tab2`, mais pour la construction du nouveau tableau on prendra la plus petite des deux valeurs avant de passer à la suite, sur un seul tableau.

## 4 Algorithmes

Dans cette section les questions demandent un peu plus d'analyse.

1. On souhaite savoir s'il existe un élément de `tab1` et un de `tab2` dont la somme est 10000.

2. On souhaite savoir s'il existe deux éléments de **tab1** dont la somme est un élément de **tab2**
3. On souhaite connaître le nombre d'éléments qu'ont en commun **tab1** et **tab2**