



# TP 1: prise en main et premiers codes

## Pour ceux qui n'ont jamais utilisé Linux

Pour ce cours, vous utiliserez deux outils:

- une *console* (ou *terminal*): c'est une fenêtre où vous pouvez taper des commandes. Vous pouvez ouvrir une console en cliquant sur le bouton se trouvant à gauche de l'écran. 
- un *éditeur de texte* qui vous servira à écrire votre code. Dans les salles `unix`, vous avez le choix d'utiliser `gedit` (éditeur par défaut) ou `GNU Emacs`. Pour ouvrir un éditeur, vous pouvez accéder au menu en utilisant la touche `windows` ou bien en cliquant sur le bouton . Une fois dans le menu, cliquez sur l'icône de l'éditeur, si elle est absente, tapez le nom de l'éditeur.

Voici quelques commandes très utiles que vous pouvez utiliser dans un terminal:

- `cd` change directory, permet de changer de répertoire courant
  - `cd ..` permet d'accéder au répertoire père
  - `cd monRepertoire` permet d'accéder au répertoire `monRepertoire`
- `mkdir` make directory, permet de créer un répertoire: `mkdir Java` créé un répertoire `Java`.
- `ls` list, permet de connaître la liste des fichiers et répertoires contenus dans le répertoire courant.
- `more nomFichier` permet d'afficher le contenu du fichier `nomFichier`

## Premier programme

Ouvrez un fichier `Hello.java` dans un répertoire de votre choix (e.g. `Java/`) et copiez le texte ci-dessous:

```
1 class Hello{
2     public static void main(String [] args){
3         System.out.println("Hello world!");
4     }
5 }
```

Ouvrez une console. Naviguez pour que le répertoire courant soit celui qui contient le fichier `Hello.java`. Pour *compiler* le programme, tapez la ligne suivante et tapez entrée

```
monLogin@B048-18:~/Java/$ javac Hello.java
```

Cette instruction lance un programme appelé `javac` (Java Compiler) qui est un *compilateur* de Java. Un compilateur traduit un code écrit dans un langage haut niveau en un code écrit dans un langage bas niveau qui peut être exécuté par la machine, en ce cas, la machine virtuelle de Java.

Pour exécuter votre code, vous pouvez maintenant taper:

```
monLogin@B048-18:~/Java/$ java Hello
```

Cette instruction appelle la machine virtuelle de Java, qui exécute la fonction main contenu dans la classe qui a le même nom que le fichier (en ce cas Hello). Si tout se passe bien, la ligne Hello World s'affiche sur votre terminal.

```
monLogin@B048-18:~/Java/$ java Hello
Hello World!
```

## Exercice 1: Collatz

En mathématiques, on appelle suite de Syracuse une suite d'entiers naturels définie de la manière suivante

On part d'un nombre entier plus grand que zéro ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite d'entiers positifs dont chacun ne dépend que de son prédécesseur. (Crédit : Wikipedia)

Par exemple, à partir de 14, on construit la suite des nombres : 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2 C'est ce qu'on appelle la suite de Syracuse du nombre 14.

Evidemment, une fois qu'on a atteint 1, la suite se répète. La **Conjecture de Collatz** ou Conjecture de Syracuse dit que  $\forall n$ , la suite de Syracuse de  $n$  atteint 1.

Pour cette exercice vous devez écrire un programme qui, étant donné un entier positif  $n$ , affiche la suite de Syracuse à partir de  $n$ , jusqu'à 1.

## Exercice 2: Devinette

Ecrire un programme qui fait deviner un entier entre 0 et 20. Pour ce faire, votre code doit d'abord choisir aléatoirement un nombre cible entre 0 et 20. Puis il doit demander à l'utilisateur d'entrer un nombre jusqu'à ce que celui-ci entre le nombre cible. A chaque fois que l'utilisateur échoue, le code peut lui indiquer si le nombre cible est supérieur ou inférieur à la tentative de l'utilisateur.

**NB:** Vous pouvez utiliser la classe Random pour tirer un nombre aléatoire avec nextInt(). Lisez sa description sur une référence <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

## Exercice 3: Nombre premier

Implementer une fonction statique estPremier qui prend en argument un entier et qui retourne un booléen indiquant si l'entier est un nombre premier ou non.

### Exercice 4: Monte Carlo $\pi$

Ecrire un programme qui calcule une approximation de  $\pi$  avec la méthode de Monte-Carlo. Pour cela on choisit au hasard un point dans un carré de côté 2 centré sur l'origine et on regarde s'il est dans le cercle de rayon 1 centré sur l'origine. La surface du disque de rayon 1 étant  $\pi$ , la probabilité qu'un point tiré au hasard dans le carré soit dans le cercle est égale à la surface du disque divisée par la surface du carré, soit  $\frac{\pi}{4}$ . Utiliser la méthode `nextDouble` de la classe `Random`.

