

Exercice 1 – Le tableau noir

Au début des épisodes des *The Simpsons*, Bart écrit des phrases amusantes sur le tableau (Figure 1). Ecrivez un programme qui donne la possibilité à madame Krapabelle de préciser le nombre de fois que Bart doit écrire, p.ex. la phrase "Je ne suis pas allergique à la programmation" (ou une autre phrase de votre choix). Le nombre est lu à partir de la console et la phrase est affichée autant de fois à l'écran.



Figure 1 – Copyright: 20th Century Fox

Exercice 2 – Somme

Ecrivez un programme qui lit un entier positif n de l'entrée standard et qui calcule la somme de tous les entiers de 1 à n . Tant que l'utilisateur n'a pas entré un entier positif continuez à le lui demander. Calculez la somme de trois façons différentes: en utilisant une boucle `for`, une boucle `while` et une boucle `do-while`. Comparez vos résultats au résultat de la formule

$$\sum_{i=1}^n i = \frac{n \cdot (n + 1)}{2}$$

Exercice 3 – Moyenne

Ecrivez un programme qui calcule la moyenne d'une suite de nombres lus de la console. Si l'utilisateur entre le nombre 0, le programme va arrêter de lire des nombres et va afficher la moyenne des nombres entrés.

Exercice 4 – Conversion de nombres

Ecrivez un programme qui lit un entier de la console et affiche chaque chiffre du nombre sous forme de texte. Par exemple, si le nombre 314 est lu, la console affichera trois un quatre.

Indices:

- Utiliser la division d'entiers et l'opérateur modulo (%) pour extraire les différents chiffres du nombre.
- Utiliser un `switch` pour relier les chiffres avec leur représentation sous forme de texte.
- Si le nombre lu est négatif, écrivez moins au début du nombre.

Exercice 5 – Décomposition d'une pièce d'1 €

Ecrivez un programme qui calcule toutes les possibilités de décomposer une pièce d'1 € dans des pièces de 20, 10 et 5 centimes. Affichez toutes les possibilités ainsi que le nombre total de possibilités.

TP 3 – Structures de répétition

Exercice 6 – Devinez le nombre

Ecrivez un programme qui choisit un nombre entre 1 et 100 au hasard et qui laisse l'utilisateur deviner le nombre correct. A chaque essai le programme lui indiquera si le nombre entré est plus petit ou plus grand que le nombre recherché jusqu'à ce que le nombre correct ait été deviné. Le programme va alors afficher le nombre d'essais dont l'utilisateur avait besoin pour trouver le bon nombre.

Nombres Pseudo-aléatoires en Java

Pour générer un nombre pseudo-aléatoire en Java, vous pouvez utiliser la classe `Random` du package `java.util` (import à la ligne 1). Lorsque vous initialisez un générateur de nombres aléatoires, la *graine* (seed) est la valeur initiale de l'état interne du générateur de nombres pseudo-aléatoires^a. Une graine non-constante, comme par exemple l'heure du système, garantit une suite de nombre pseudo-aléatoires différente à chaque exécution du programme (ligne 4). La méthode `nextInt(int bound)` (ligne 5) retourne un nombre entre 0 (inclu) et la limite spécifiée (exclue).

```
1 import java.util.Random;
2 public class RandomExample {
3     public static void main(String[] args) {
4         Random generator = new Random(System.currentTimeMillis());
5         int randomNumber = generator.nextInt(42);
6     }
7 }
```

^a<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

Exercice 7 – Suite de Fibonacci

Ecrivez un programme qui affiche la suite de Fibonacci \mathcal{F} jusqu'au $n^{\text{ième}}$ terme \mathcal{F}_n . Trouvez une solution en utilisant des boucles et non pas la récursivité. Notez que la suite de Fibonacci est définie comme suit:

$$\begin{cases} \mathcal{F}_1 = 1 \\ \mathcal{F}_2 = 1 \\ \mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2} \quad n > 2 \end{cases}$$

Exercice 8 – Réfactorisation: Extraction de variables

Comme vu dans le cours de vendredi, il y a plusieurs raisons pourquoi l'extraction de variables est une technique de réfactorisation utile par rapport à la maintenance de code. Dans Eclipse, pour extraire une variable d'une expression, sélectionnez l'expression, faites un clic droit et choisissez Refactor → Extract Local Variable Essayez ceci en extrayant les variables pour chacune des trois expressions dans la structure if du code suivant.

```
1 String platform = "macOS Sierra";
2 String browser = "Safari";
3 double zoomLevel = 1.5;
4 if (platform.toUpperCase().indexOf("MAC") > -1 && browser.equals("Safari") && zoomLevel >=
5     1.5) {
6     // do something ...
7 }
```