

TP 2

M.Adam – N.Delomez – JF.Kamp – L.Naert

16 septembre 2022

Objectifs du TP

- Comprendre la boucle
- Construire une boucle simple avec une seule condition de sortie et sans imbrication

Exercice 1 (reprise du TD 2) (*)

1. Ecrire un programme qui demande à l'utilisateur de saisir une suite de nombres et les affiche. La saisie se termine par l'entrée de -1 ;
2. **Rendre le code du programme et le résultat de son exécution.**
3. Modifier le programme pour qu'il affiche la moyenne des nombres saisis. Le -1 n'est pas comptabilisé.
4. **Rendre le code du programme et des résultats de son exécution.**

Exercice 2 (*)

1. Tester le programme suivant :

```
/**
 * Rôle à deviner
 * @author M.Adam
 */
class PGCD {

    void principal() {
        int val1;
        int val2;

        val1 = SimpleInput.getInt ("Première valeur : ");
        val2 = SimpleInput.getInt ("Deuxième valeur : ");
```

```
        while (val1 != val2) {
            if (val1 > val2) {
                val1 = val1 - val2;
            } else {
                val2 = val2 - val1;
            }
        }
        System.out.println("Le résultat est : " + val1);
    }
}
```

2. **Rendre :**

- le code du programme
 - des exécutions qui se terminent
 - des exécutions qui ne se terminent pas
 - le rôle du programme.
3. Modifier le programme pour que les nombres saisis soient strictement positifs. La saisie se poursuit tant que la valeur n'est pas correcte.
4. **Rendre le code corrigé du programme et le résultat de son exécution.**

Exercice 3 (**)

1. Écrire un programme qui saisit une suite de nombres et qui s'arrête quand le nombre saisi est inférieur au précédent. Par exemple, si le nombre saisi précédemment est 20, la saisie s'arrête, entre autre, avec 10.
2. **Rendre le code du programme et les tests d'exécution.**

Exercice 4 (**)

1. Écrire un programme qui saisit deux valeurs dans deux variables, x un réel et n un entier, et calcule x^n .
Pour rappel :
 - x^n vaut 1 si n vaut 0,
 - x^n vaut $1/x^{-n}$ si n est strictement négatif.Par exemples :
 - 2^3 vaut $2 * 2 * 2$ soit 8
 - 2^0 vaut 1
 - 2^{-3} vaut $1/2^3$ soit $1/8$ ou 0,125.
2. **Rendre le code du programme et les tests d'exécution.**

Exercice 5 (**)

L'expression `(int) (Math.random() * 100)` donne un entier supérieur ou égal à 0 et inférieur strictement à 100.

1. Écrire un programme qui calcule un entier entre 0 et 100, demande à l'utilisateur de deviner le nombre. Le programme ne s'arrête que quand l'utilisateur a trouvé ce nombre. Le programme indique "Trop grand !" ou "Trop petit !" en fonction de la valeur saisie par l'utilisateur. A la fin, le programme indique le nombre d'essais de l'utilisateur.
2. **Rendre le code du programme et les tests d'exécution.**

Exercice 6 (***)

L'objectif de cet exercice est de coder et de tester un programme qui devine un chiffre entre 0 et 1000 choisi par l'utilisateur. Le principe est le suivant :

- l'utilisateur choisi un nombre entre 0 et 1000
 - le programme fait une proposition.
 - l'utilisateur répond + si le nombre proposé est trop petit, - si le nombre proposé est trop grand et = si le nombre proposé est correct.
 - le programme propose des nombres jusqu'à avoir trouvé le nombre choisi.
1. Écrire le programme qui devine le nombre choisi.
 2. **Rendre le code du programme et les tests d'exécution.**
 3. Modifier le programme précédent pour qu'il devine quand le nombre est forcément deviné. Par exemple, pour deviner le nombre 701, le programme si le programme propose 700 et 702, il doit en déduire la bonne réponse.
 4. **Rendre le code du programme et les tests d'exécution.**