

TP 1

L. Naert – T. Ferragut – E. Lemonnier

5 juillet 2024

Objectifs du TP

- Apprendre à utiliser l'environnement Java du module R1.01
- Corriger les erreurs de syntaxe
- Construire des algorithmes élémentaires
- Utiliser les variables, l'affectation, la séquentialité, l'instruction conditionnelle

Avant-Propos

Avant de commencer les exercices de cette séance de TP, il est nécessaire de mettre en place l'environnement de développement.

Les opérations à effectuer sont détaillées dans un document disponible sous moodle :

<https://moodle.univ-ubs.fr/course/view.php?id=7518>. La clé est R1.01.

Le rendu est à effectuer sur moodle. Il doit être formaté selon le modèle donné sous moodle.

Exercice 1 (*)

1. Saisir l'algorithme suivant sur Java :

```
/**
 * Essai de la configuration R1.01
 * @author L. Naert
 */
class TestConfig{
    void principal(){
        int maVar;
        maVar = SimpleInput.getInt ("maVar = ");
        maVar = maVar * 2;
        System.out.println ("maVar * 2 = "+ maVar);
    }
}
```

2. Exécuter l'algorithme sur Java
3. Faites en sorte d'afficher "Ce programme fonctionne" une ligne en dessous du premier affichage

Rien n'est à rendre pour cet exercice

Exercice 2 (*)

1. Corriger les erreurs de syntaxe et de convention d'écriture du programme suivant :

```
/*
 * @author ??
 */
class Syntaxe{
    void principal(){

        int val1, Val2;
        int val3;

        val1 = SimpleInput.getInt("Premier entier :");
        Val2 = SimpleInput.getInt("Deuxième entier :");
        val = SimpleInput.getInt("Troisième entier :");

        if (val1<val3&&Val2<val3) {
            System.out.println
                ("Le troisième entier " + val3 + " est le plus grand.");
        } else if (val1<Val2&&val3<Val2) {
            System.out.println
                ("Le deuxième entier " + Val2 + " est le plus grand.");
        } else if (Val2<val1&&val3<val1) {
            System.out.println
                ("Le premier entier " + val1 + " est le plus grand.");
        }
    }
}
```

2. Exécuter le programme corrigé

Rendre le programme corrigé et un exemple d'exécution.

Exercice 3 (*)

Écrire un programme qui calcule le périmètre et l'aire d'un cercle. Le rayon du cercle est saisi par l'utilisateur.

Rendre le texte de l'algorithme et le résultat de plusieurs exécutions.

Exercice 4 (**)

Écrire un programme pour simuler les règles du bowling. Au bowling, on a le droit à deux lancers sur 10 quilles. Le programme demande le nombre de quilles renversées avec chacune des deux boules et affiche :

1. "Strike !" si toutes les quilles sont renversées à la première boule,
2. "Spare" si toutes les quilles sont tombées après la deuxième boule
3. sinon, le nombre de quilles renversées au total.

Attention, le programme doit afficher "Nombre incorrect" si l'un des nombres entrés est incorrect et ne doit pas demander le nombre de boules au deuxième lancer si les quilles sont toutes tombées au premier lancer.

Rendre le code du programme et des tests d'exécution.

Exercice 5 (**) : Exercice facultatif

Cet exercice est optionnel et n'intervient qu'en bonus en cas de notation.

Écrire un programme qui édite une fiche de salaire à partir d'un salaire brut demandé à l'utilisateur. Les prélèvements sont récapitulés dans le tableau suivant :

Intitulés	Pourcentage	Assiette
Assurance maladie	0,75	Totalité du salaire
Assurance vieillesse déplafonnée	0,1	Totalité du salaire
Assurance vieillesse plafonnée	6,75	Totalité du salaire
Frais professionnels	1,75	Totalité du salaire
Contribution sociale généralisée	7,5	Salaire total après déduction des frais professionnels
CRDS	0,5	Salaire total après déduction des frais professionnels
Chômage	2,4	Totalité du salaire

La fiche de salaire présente :

- le salaire brut sur une ligne
- chaque prélèvement sur une ligne
- le total des prélèvements sur une ligne
- le salaire net perçu sur la dernière ligne

Rendre le texte de l'algorithme et le résultat d'une exécution

Exercice 6 (***) : Exercice facultatif

Cet exercice est optionnel et n'intervient qu'en bonus en cas de notation.

Le Programme National de la première année du BUT informatique est construit de la manière suivante :

Compétence	Réf.	Intitulé
Compétence 1	C1	Réaliser un développement d'application
Compétence 2	C2	Optimiser des applications
Compétence 3	C3	Administrer des systèmes informatiques communicants complexes
Compétence 4	C4	Gérer des données de l'information
Compétence 5	C5	Conduire un projet
Compétence 6	C6	Collaborer au sein d'une équipe informatique

Le passage en deuxième année est automatique si :

- la moyenne de 10 est obtenue dans au moins 4 compétences ;
- et il n'y a aucune compétence dont la moyenne est en dessous de 8.

Écrire et exécuter le programme qui demande à l'utilisateur sa note dans chaque compétence, et lui indique :

- s'il a obtenu son année et combien de compétences ont une moyenne entre 8 et 10 ;
- s'il n'a pas obtenu son année et les raisons de cette non obtention.

Rendre le programme et, au moins, le résultat d'une exécution.