

## **BTS SIO SLAM - première année - Harmonisation (2/2)**

### **TP n°2 - Bases de Java et structures conditionnelles**

#### **Table des matières**

PARTIE 1.....	2
1.Objectifs.....	2
2.Mise en place du projet.....	2
3.Variables et calculs.....	3
3.1.Exercice 1 - Variables et affichage.....	3
3.2.Exercice 2 - Comprendre et corriger des erreurs.....	3
3.3.Exercice 3 - Constantes.....	4
4.Conditions simples.....	5
4.1.Exercice 4 - Test de majorité.....	5
4.2.Exercice 5 - Tarif cinéma.....	5
5.Auto-correction IA.....	6
5.1.Objectifs IA 1.....	6
5.2.Prompt IA 1.....	6
PARTIE 2.....	7
1.Objectifs.....	7
2.Réactivation des acquis.....	7
3.Booléens et logique.....	7
3.1.Exercice 6 - Expressions logiques.....	7
3.2.Exercice 7 - Température.....	8
4.Programme final : menu utilisateur.....	9
5.Auto-correction IA.....	10
5.1.Prompt IA 2.....	10
5.2.Évaluation.....	10

# PARTIE 1

Découverte de Java – Variables, calculs, conditions simples

## 1.Objectifs

- Créer un projet Java fonctionnel
- Déclarer des variables et constantes
- Comprendre les erreurs de typage
- Utiliser une structure if / else
- Compiler sans erreur
- Analyser à l'aide de l'IA

## 2.Mise en place du projet

Créer le projet Java : **TP\_Bases\_Java**

Créer le package : **tp.base**

Créer la classe : **Main.java**

Code de base :

```
package tp.base;  
public class Main {  
    public static void main(String[] args) {  
        ...  
    }  
}
```

### 3.Variables et calculs

#### 3.1.Exercice 1 - Variables et affichage

- Déclarer :

```
double prix1 = 19.99;  
double prix2 = 26.50;
```

- Calculer la somme dans une variable total
- Afficher le total à l'écran

Programme attendu (2 lignes) :

#### 3.2.Exercice 2 - Comprendre et corriger des erreurs

- Corriger le code suivant pour qu'il compile correctement.
- Ajouter des commentaires dans le code expliquant les erreurs corrigées.

Corrections & explications :

```
int nb1 = 5.4;  
  
double nb2 = 10;  
  
double nb3 = 3;  
  
int res = nb2 / nb3;  
  
System.out.println(res);
```

### 3.3.Exercice 3 - Constantes

- Déclarer la constante suivante :

```
final int AGE_MIN = 18;
```

- Essayer de réaffecter une nouvelle valeur à `AGE_MIN`. Que constatez-vous après avoir tenté cette modification ? Quels messages d'erreur ou comportements avez-vous observés ?

- Analyser pourquoi la déclaration `final` est importante pour la sécurité et la maintenabilité du code.

Réponse argumentée :

## 4.Conditions simples

### 4.1.Exercice 4 - Test de majorité

- Demander l'âge de l'utilisateur avec Scanner
- Afficher :

"Mineur" si l'âge est inférieur à 18

"Majeur" sinon

Programme attendu :

### 4.2.Exercice 5 - Tarif cinéma

Afficher le tarif en fonction de l'âge :

- Moins de 5 ans : 2 €
- De 5 à 17 ans : 5 €
- 18 ans et plus : 9 €

**À soumettre à l'enseignant : le code et le programme fonctionnels en Java.**

## 5.Auto-correction IA

### 5.1.Objectifs IA 1

- Vérifier la syntaxe
- Comprendre les erreurs
- Améliorer la lisibilité du code

### 5.2.Prompt IA 1

PROMPT D'AUTO-CORRECTION JAVA (BTS SIO SLAM)

<https://duckduckgo.com/>

<https://mistral.ai/fr>

<https://chatgpt.com/>

Contexte :

Je suis étudiant en BTS SIO SLAM 1re année.

Je débute en Java et je travaille sur les notions suivantes :

- variables et types
- constantes (**final**)
- entrées utilisateur avec **Scanner**
- structures conditionnelles (**if / else**)

Objectif :

Analyser mon fichier **Main.java** pour m'aider à progresser en Java.

Consignes :

- Vérifie si mon code compile correctement
- Identifie les erreurs de syntaxe ou de logique
- Signale les mauvaises pratiques éventuelles
- Explique chaque correction avec des mots simples (niveau débutant)
- Propose des améliorations légères et accessibles
- Ne réécris pas tout le programme, corrige uniquement ce qui est nécessaire

Code à analyser :

**// COLLER ICI VOTRE CODE**

Règle importante :

Je dois être capable de comprendre et de justifier chaque correction proposée.

## PARTIE 2

Conditions multiples, booléens et menu utilisateur

### 1.Objectifs

- Utiliser if / else if / else
- Manipuler des booléens
- Utiliser Random
- Créer un menu avec while
- Structurer un programme simple

### 2.Réactivation des acquis

- variables
- conditions
- Scanner

### 3.Booléens et logique

#### 3.1.Exercice 6 - Expressions logiques

- Recopier et exécuter le code suivant :

Indiquer et expliquer les résultats :

```
int nb1 = 10, nb2 = -5;  
  
System.out.println(8 > 5);  
  
System.out.println(5 == 5);  
  
System.out.println(nb1 > nb2);  
  
System.out.println(nb1 == nb2);
```

### 3.2.Exercice 7 - Température

- Demander une température corporelle
- Si la température est entre 36 et 38, afficher "OK"
- Sinon afficher "Allez voir un médecin"

Programme attendu :



## 4.Programme final : menu utilisateur

Travail demandé

Créer un programme qui :

- Demande un pseudo
- Demande une année de naissance
- Calcule l'âge
- Indique si l'utilisateur est majeur ou mineur
- Affiche le menu suivant :

1 - Générer une température aléatoire

2 - Quitter

- Le menu doit se répéter tant que l'utilisateur ne choisit pas de quitter

Classes utiles : `Scanner`, `Random`, `while`

**À soumettre à l'enseignant : le code et le programme fonctionnels en Java.**

## 5.Auto-correction IA

### 5.1.Prompt IA 2

Contexte :

Je suis étudiant en BTS SIO SLAM 1re année.

J'ai réalisé un programme Java avec :

- des conditions
- des booléens
- la classe Random
- un menu avec while

Consigne :

Analyse mon code Java pour évaluation :

- Vérifie la logique du programme
- Dis si le menu fonctionne correctement
- Propose des améliorations simples (lisibilité, structure)
- Explique chaque suggestion clairement

Code à analyser :

// COLLER ICI VOTRE CODE

### 5.2.Évaluation

- Programme fonctionnel (5 pts)
- Respect des consignes (5 pts)
- Capacité à expliquer son code (5 pts)
- Utilisation raisonnée de l'IA (5 pts)