

TD N° 1

Exercice 1

1. Java est un langage

- (a) Compilé
- (b) Interprété
- (c) Compilé et interprété
- (d) Ni compilé ni interprété

2. Java est un langage développé par

- (a) Hewlett-Packard
- (b) Sun Microsystems
- (c) Microsoft
- (d) Oracle

3. L'interprétation des programmes Java est effectuée par

- (a) API
- (b) JDK
- (c) JVM
- (d) AWT

4. Quels keywords sont utilisés pour spécifier la visibilité des propriétés et des méthodes ?

- (a) final
- (b) private
- (c) abstract
- (d) protected
- (e) public

5. Quel concept de Java est utilisé en combinant des méthodes et des attributs dans une classe?

- (a) Polymorphisme
- (b) Encapsulation
- (c) Abstraction
- (d) Héritage

6. Si le constructeur n'est pas défini dans une classe, celle-ci

- (a) n'est pas instanciable
- (b) est dite « virtual »
- (c) en fournit un par défaut
- (d) ne fonctionne pas

7. La création d'un nouvel objet à partir d'une classe s'appelle :

- (a) instantiation
- (b) construction
- (c) surcharge
- (d) implémentation

8. Pour la classe B définie comme suit :

```
class B {
    public B()
        {System.out.print("Hello ");};
    public B(int i)
        {this();
        System.out.println("Bonjour "+i);};
}
```

Qu'affichera l'instruction suivante : B monB=new B(2024);

- (a) erreur de compilation
- (b) erreur d'exécution
- (c) Hello Bonjour 2024

Exercice 2

Définir la classe **Rectangle** caractérisée par :

- Deux attributs privés (**Longueur**, **Largeur**).
- Un constructeur permettant d'initialiser la longueur et la largeur d'un objet de la classe Rectangle.
- Les méthodes publiques suivantes permettant :
 - D'obtenir la longueur d'un rectangle.
 - D'obtenir la largeur d'un rectangle.
 - De modifier la longueur d'un rectangle.
 - De modifier la largeur d'un rectangle.
 - De calculer la surface d'un rectangle.
 - De calculer le périmètre d'un rectangle.
 - D'afficher un rectangle.
 - De comparer deux rectangles.

Ecrire un programme JAVA permettant de créer des objets de la classe Rectangle et d'utiliser les méthodes définies dans cette classe.

Exercice 3 :

Écrivez une classe **Article** pour représenter les articles vendus dans un supermarché. Chacun comporte quatre attributs :

- **private long reference** – un numéro qui caractérise l'article de manière unique.
- **private String intitule** – la description de l'article sous forme de texte.
- **private float prixHT** – le prix unitaire hors taxes de l'article.
- **private int quantiteEnStock** – le nombre d'unités de l'article disponibles en stock.

et dispose d'un certain nombre de méthodes :

- **public Article(long reference, String intitule, float prixHT, int quantiteEnStock)** – constructeur paramétré qui donne une valeur à chacun des attributs.
- **public long getReference(), public String getIntitule(), public int getQuantiteEnStock(), public float getPrixHT()** – les « accesseurs » permettant d'obtenir la valeur de chaque attribut.
- **public void approvisionner(int nombreUnites)** – méthode pour augmenter la quantité disponible de l'article.
- **public boolean vendre(int nombreUnites)** – cette méthode enregistre la vente d'un certain nombre d'unités de l'article. Si nombreUnites est supérieur à la quantité disponible alors le stock n'est pas modifié et la méthode renvoie false ; autrement elle effectue les mises à jour nécessaires et renvoie true.

- **public float MontantHT()** – cette méthode calcule et renvoie le Montant HT de l'article avec $\text{MontantHT} = \text{prixHT} * \text{quantiteEnStock}$.
- **public float MontantTTC()** – cette méthode calcule et renvoie le Montant TTC de l'article étant donné que le taux de la tva est =0.2 (tva est une constante à définir). $\text{MontantTTC} = \text{MontantHT} * (1 + \text{tva})$
- **public String toString()** – cette méthode renvoie une chaîne de caractères exprimant la référence, l'intitulé et le prix de l'article.
- **public boolean equals(Article A)** – teste si l'article A passé en paramètre est équivalent à l'objet courant.

a.equals(b) est true si et seulement si a et b représentent le même article (c'est-à-dire s'ils ont la même référence).

Pour tester cette classe écrivez une méthode main qui crée un tableau comportant trois articles et qui teste toutes ces méthodes en adoptant le scénario suivant :

référence	intitulé	prixHT	quantitéEnStock
111635	A1	120	30
111636	A2	20	220
111637	A3	100	5

- Afficher les différents articles.
- Augmenter la quantité de chaque article par 20 pièces.
- Vendre 30 pièces pour chaque article si c'est possible sinon, afficher un message d'erreur.
- Afficher le prix HT et le prix TTC de chaque article.
- Tester l'égalité du premier et du dernier article.

Exercice 4

On voudrait gérer chaque étudiant d'une institution à l'aide d'une classe Etudiant, définie par les attributs suivants :

- NomPren: nom et prénom d'un étudiant
- TabNotes: tableau contenant les notes d'un étudiant
- NbNotes: nombre de notes dans le tableau

Et les méthodes suivantes :

- Etudiant() (constructeur non paramétré)
- void saisie(), permettant la saisie d'un étudiant
- void affichage(), permettant l'affichage d'un étudiant
- float moyenne(), retourne comme résultat la moyenne des notes d'un étudiant
- boolean admis(), retourne vrai, si un étudiant est admis et faux, sinon. Un étudiant est

considéré comme étant admis lorsque la moyenne de ses notes est supérieure ou égale à 10.

- int Nb_Crédits(), retourne comme résultat :
 - la valeur 0, si un étudiant est Admis
 - le nombre de notes qui sont inférieures à la moyenne, si l'étudiant n'est pas admis.

1) Donnez la définition de la classe Etudiant dans le langage java.

2) On voudrait gérer un groupe d'étudiants faisant partie d'une même classe, par une classe java que l'on nommera *UneClasse* et que l'on implémentera à l'aide d'un tableau d'étudiants. Cette classe sera composée des attributs suivants :

- Niveau : niveau d'études et spécialité (chaîne de caractères)
- Tab: Tableau contenant les étudiants de la classe
- N: Nombre d'étudiants de la classe

Et des méthodes suivantes :

- UneClasse() (constructeur non paramétré)
- void Ajout(Etudiant E), permettant d'ajouter un étudiant E dans la classe
- void Suppr(int i), permettant de supprimer le ième étudiant de la classe.
- void Résultats(), permettant l'affichage des résultats obtenus par tous les étudiants de la classe, c'est-à-dire listage de leur nom et prénom ainsi que leur moyenne.
- int NbAdmis(), retourne le nombre d'étudiants admis dans la classe.

3) Donnez la définition de la classe UneClasse dans le langage java.