TD 02
Programmation objet - Java

Énoncé :

Une société enregistre les niveaux de stock des produits qu'elle vend. Chaque produit est représenté par une instance de la classe **Produit**, qui enregistre :

- Son identité : un numéro de référence de type entier,
- Son nom : une dénomination en lettres de type chaîne de caractères,
- Sa quantité : le nombre d'unités en stock de type entier.

L'instanciation d'un nouveau produit nécessite des informations sur son identité et son nom.

La quantité en stock initiale est toujours de zéro.

La classe **Produit** définit :

- La méthode **augmenterQuantite()** pour enregistrer les augmentations du stock d'un produit selon la valeur qui lui est passée en paramètre. Si cette valeur est négative, cette méthode affiche un message d'erreur et ne change rien.
- La méthode **vendreUn()** qui enregistre la vente d'une unité en diminuant la quantité du produit de 1. Elle affiche une erreur si le produit est épuisé.

1) De quoi est constitué **l'état** d'un objet de type **Produit** ?
   ↳ les attributs
2) De quoi est constitué son **comportement** ?
   ↳ Les méthodes
3) Écrire les instructions Java qui permettent de définir la classe **Produit** avec les contraintes suivantes :
   - Respecter les principes de **l'encapsulation** de la manière la plus stricte. ; privés
   - L'état d'un produit ne pourra être modifié que par les deux méthodes précédemment citées.
   - Prévoir un moyen d'obtenir les valeurs de chaque attribut d'un produit. ↳ get

a. Écrire les instructions Java qui permettent de définir la classe **ProduitFrais** en évitant les répétitions.

b. Que doit-on changer dans la classe **Produit** ? Expliquer.

c. Que doit-on changer dans la classe **GestionStock** ? Expliquer.

8) Est-il possible de définir la classe **Produit** en tant que classe Abstraite ? Si oui, que doit-on changer ? Expliquer.

- TD . 02 -

```
public class Produit {
    private int id;
    private String nom;
    private int qt = 0;
    public produit (int id, String nom) {
        this.id = id;
        this.nom = nom;
    }
    public void augmenter_qt(int qt){
        if(qt>0)
        { this.qt += qt; }
        else
        { System.out.println("erreur")
        }
    }
Public void vendreUn () {
    if(qt>0) {
        qt --; }
    else {
        System.out.println("stock epuisé")
    } }
```

```
public int getId(){
        return id;
    }
public String getNom(){
        return nom;
    }
Public int getQt(){
    return qt;
    }
```

5)

```
import java.util.List;
class GestionStock {
Private List<Produit> lp =
    new ArrayList<Produit>();
Public void ajouterProduit(Produit p)
{
    lp.add(p);
}
Public void AfficheDétailProduit
    (){
for(int i=0; i<lp.size; i++)
System.out.println(lp.get(i).toString());
```

Question 5. Remplir la 1ère colonne du tableau par les termes correspondant aux descriptions ci-dessous :

Terme	Description
Encapsulation	Principe selon lequel les attributs d'un objet ne [illegible]

this.dp = dp; } pour accéder à un objet sans construct par set

b / constructeur avec 3arg au niveau de class Produit → pour faire appel

c/ liste contient des produits de classe mère donc on peut ajouter des ProduitsFrais

dessous :

Terme	Description

```
Public void livraison (int id, int qt)
  { Produit p = null ;
    p = this.trouverProduit (id);
    if ( p != null )
       { p.setQt (p.getQt + qt) ;
    ou bien
       for (Produit p : lp)
         { if (p.getId() == id )
```

```
Public void Affiche()
    { System.out.println(x + "," + y + "," 3);
        }.

class Test {
        main()
    { Vecteur V1 = new Vecteur (1,2,3);
      Vecteur V2 = new Vecteur (4,5,6);
      Vecteur W = V1.somme (V1, V2);
                  ↓ Lazem AY objet mazmel bih l'appel de méthode
      V1.affiche();
    }.
```