



ASD – 2023/2024

### Enoncé du TP N°01

#### Exercice 1 :

On souhaite simuler le fonctionnement d'un distributeur de boissons chaudes selon le modèle proposé :

```
public class Boisson{
    private String nom;
    private int prix;

    public Boisson (String nom, int prix) {
        this.nom = nom;
        this.prix = prix; }

    public int getNom() {
        return nom; }

    public void setNom(String nom) {
        this.nom = nom; }

    public int getPrix() {
        return prix; }

    public void setPrix(int prix) {
        this.prix = prix; }
}

public class Distributeur {
    private int choix;

    public Boisson selection(int choix){
        setChoix(choix);
        if(choix == 1){
            return new Boisson('Café', 50); }
        else if (choix == 2){
            return new Boisson('Thé', 40); }
        else {
            return new Boisson('Lait', 30); }
    }

    public int monnaie(int dinars){
        ...
    } }
}
```

1. En tenant en compte le code de la classe **Boisson**, qui est définie par son « nom » et son « prix » :
  - a. Surchargez la classe « Boisson » avec un **constructeur vide**.
  - b. Surchargez-la avec un autre **constructeur** qui met « 0 » comme **prix** par défaut.

2. La classe **Distributeur** ci-dessus est incomplète. Ecrivez les **getters** et les **setters** qui manquent.
3. Sachant que lors de l'insertion d'une pièce de monnaie, le choix de la boisson est déjà sauvegardé. Complétez la méthode « **monnaie** » qui renvoie la monnaie en fonction du prix de la boisson choisie et la pièce insérée dans le distributeur.
4. Ecrivez la classe principale « **Test** » qui contient la méthode **public static void main(...)** dans laquelle vous allez :
  - a. Créer un objet nommé « **D1** » de type « **Distributeur** ».
  - b. à partir de **D1**, commandez du « **Thé** ».
  - c. Commandez une autre boisson en saisissant votre choix à partir du clavier.
  - d. Calculez et affichez la monnaie de 100 dinars.

## **Exercice 2 :**

Créez une classe **Etudiant** qui représente les informations d'un étudiant. Cette classe doit contenir :

- Un attribut privé nom de type **String** qui représente le **nom** de l'étudiant.
- Un attribut privé note de type **float** qui représente la **note** de l'étudiant.
- Un attribut privé groupe de type **int** qui représente le **groupe** où est affecté l'étudiant.
- Un constructeur **Etudiant(String nom, float note, int groupe)** permettant d'initialiser le nom, la note et le groupe de l'étudiant.
- La méthode **String getNom()** permettant de retourner le nom de l'étudiant.
- La méthode **float getNote()** permettant de retourner la note de l'étudiant.
- La méthode **int getGroupe()** permettant de retourner le groupe de l'étudiant.
- La méthode **void setNom(String nom)** permettant de modifier le nom et prénom de l'étudiant.
- La méthode **void setNote(float note)** permettant de modifier la note de l'étudiant.
- La méthode **void setGroupe(int groupe)** permettant de modifier le groupe de l'étudiant.

Définissez la classe **Traitement** qui contient :

- La méthode **comparer(Etudiant a, Etudiant b, Etudiant c)** qui permet de comparer les notes des étudiant(e)s « **a** » et « **b** » et met l'étudiant(e) avec la meilleure note dans l'objet **c**.
- La méthode **admis(Etudiant e, boolean res)** qui vérifie si la note de l'étudiant(e) est supérieure ou égale à 10, puis met le résultat dans la variable **res**.
- La méthode **memegroupe(Etudiant a, Etudiant b)** qui permet de constater si les deux étudiant(e)s appartiennent au même groupe et retourner le résultat.

Définissez la classe **Test** qui contient la méthode **main** dans laquelle vous allez :

- Créer **2** étudiant(e)s en saisissant leurs informations à partir du clavier.
- Créer une instance de la classe **Traitement**.
- Appeler la méthode **comparer** entre les deux étudiants, créés, puis affichez le résultat.
- Appeler la méthode **admis** pour l'un des étudiant(e) créé et afficher le résultat.
- Appeler la méthode **memegroupe** pour les deux étudiant(e)s et afficher le résultat.

***Bonne révision de la Programmation Orientée Objet***