

Contrôle continu HAI7171 – CC2 : 25 novembre 2022
Programmation par objets
(1h)

Université de Montpellier – Faculté Des Sciences
Master informatique (ICO), géomatique, bioinformatique, Physique numérique

Nous étudions la suite des éléments pour un logiciel de gestion des achats de carburant dans une station-service. Les éléments vont vous être donnés au fil des questions.

On vous rappelle les principaux éléments du contrôle précédent.

```
public enum TypeCarburant {Gazole, SP98, SP95}
```

```
public class CarteConso {  
    private String idCarte;  
    private boolean professionnelle;  
    public CarteConso() {}  
    public CarteConso(String idC, boolean pro) {  
        this.idCarte = idC;  
        this.professionnelle = pro;  
    }  
    public String getIdCarte() {return this.idCarte;}  
    public void setIdCarte(String idC) {this.idCarte = idC;}  
    public boolean isProfessionnelle() {return this.professionnelle;}  
    public void setProfessionnelle(boolean pro)  
        {this.professionnelle = pro;}  
    public String toString() {  
        return "CarteConso [idCarte=" + this.idCarte + ",  
            professionnelle=" + this.professionnelle + "];"  
    }  
}
```

```
public class Carburant {  
    private TypeCarburant type;  
    private double prixAuL;  
    public Carburant() {}  
    public Carburant(TypeCarburant type, double prixAuL)  
        {this.type = type; this.prixAuL = prixAuL;}  
    public String toString() {return "Carburant [type=" + this.type + ",  
        prixAuL=" + this.prixAuL + "];"  
    }
```

```
public class AchatCarburant {  
    private CarteConso carteConso;  
    private Carburant carburantAchete;  
    private double quantiteEnLitre;  
    public AchatCarburant() {}  
    public AchatCarburant(CarteConso carteConso, Carburant carburantAchete, double  
        quantiteEnLitre) {  
        this.carteConso = carteConso;  
    }
```

```

        this.carburantAchete = carburantAchete;
        this.quantiteEnLitre = quantiteEnLitre;
    }
    public double prixAPayer() {
        double prixBase = this.quantiteEnLitre * this.carburantAchete.getPrixAuL();
        if (this.carteConso.isProfessionnelle())
            prixBase = prixBase * 0.9;
        return prixBase;
    }
    public String toString() {
        return "AchatCarburant [carteConso=" + this.carteConso + ", carburantAchete=" +
            this.carburantAchete + ", quantiteEnLitre=" + this.quantiteEnLitre + "];"
    }
}

public static void main(String[] args) {
    CarteConso carte = new CarteConso("TR34",true);
    Carburant carbu1 = new Carburant(TypeCarburant.Gazole,2.08);
    AchatCarburant ac1 = new AchatCarburant(carte,carbu1,40);
    Carburant carbu2 = new Carburant(TypeCarburant.SP98,1.96);
    AchatCarburant ac2 = new AchatCarburant(carte,carbu2,35);
}

```

Question 1.

Ajoutez une classe pour représenter les stations-services. Une station-service a un nom, un numéro de SIRET et contient une liste d'achats de carburant.

Vous écrivez pour cette classe uniquement :

- l'entête.
- les attributs.
- un constructeur avec des paramètres pour initialiser les attributs nom et numéro de SIRET. La liste des achats doit être initialement vide.
- l'accessor en lecture de la liste des achats, qui doit permettre de la consulter mais pas de la modifier.
- une méthode permettant d'ajouter un achat.

Question 2.

Ajoutez à la classe représentant les stations-services une méthode permettant de connaître les gains réalisés. Ceux-ci correspondent à la somme des prix des achats de carburant à laquelle on retire 5% de frais de gestion.

Question 3.

Ajoutez à la classe représentant les stations-services une méthode retournant la sous-liste des achats réalisés avec une carte professionnelle.

Question 4.

Continuez la fonction **main** ci-dessus avec la création d'une station-service, l'ajout des deux achats de carburant, l'affichage des gains, et l'affichage de la sous-liste des achats réalisés avec une carte professionnelle. On rappelle que la classe **ArrayList** dispose d'une méthode **toString** qui appelle les méthodes **toString** des différents éléments de la liste.