Contrôle continu HAI717I – CC2 : 25 novembre 2022
Programmation par objets
(30 min)
Correction

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 idCarte, boolean professionnelle) {
                this.idCarte = idCarte;
                this.professionnelle = professionnelle;
        }
        public String getIdCarte() {return idCarte;}
        public void setIdCarte(String idCarte) {this.idCarte = idCarte;}
        public boolean isProfessionnelle() {return professionnelle;}
        public void setProfessionnelle(boolean professionnelle)
                {this.professionnelle = professionnelle;}
        public String toString() {
                return "CarteConso [idCarte=" + idCarte + ",
                        professionnelle=" + 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=" + type + ",
              prixAuL=" + 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=" + carteConso + ", carburantAchete=" +
                        carburantAchete+ ", quantiteEnLitre=" + 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 (7 points).

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'accesseur 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.

```
public class StationService {
        private String nom, numeroSIRET;  //0,5
        private ArrayList<AchatCarburant> listeAchats = new ArrayList<>(); //1+1

        public StationService(String nom, String numeroSIRET) { // 1
                this.nom = nom;
                this.numeroSIRET = numeroSIRET;
        }

        public ArrayList<AchatCarburant> getListeAchats(){ // 2
                return (ArrayList<AchatCarburant>)
                                Collections.unmodifiableList(this.listeAchats) ;
        }
```

```
        public void ajouteAchat(AchatCarburant a)  //1,5
        {if (! this.listeAchats.contains(a)) this.listeAchats.add(a);}

}
```

Question 2 (4,5 points).
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.

```
public double gains() {
        double sommeAchats=0;
        for (AchatCarburant a : this.listeAchats)
                sommeAchats += a.prixAPayer();
        return sommeAchats * 0.95;
}
```

Question 3 (4,5 points).
Ajoutez à la classe représentant les stations-services une méthode retournant la sous-liste des achats réalisés avec une carte professionnelle.

```
public ArrayList<AchatCarburant> sousListePro(){
        ArrayList<AchatCarburant> achatPro = new ArrayList<>();
        for (AchatCarburant a : this.listeAchats)
                if (a.getCarteConso().isProfessionnelle())
                        achatPro.add(a);
        return achatPro;
}
```

Question 4 (4 points).
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.

```
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);
        StationService st = new StationService("LeBeauVoyage", "ISID56"); //1
        st.ajouteAchat(ac1);st.ajouteAchat(ac2); //1
        System.out.println(st.gains()); //1
        System.out.println(st.sousListePro()); //1
    }
```