

Ecrit - Correction

Barème (20,5 points)

1 Partie 1

Cette partie développe quelques éléments d'un logiciel destiné à promouvoir le mouvement "zéro déchet", qui consiste à réduire de manière significative les déchets produits par notre consommation quotidienne, par quelques gestes simples, comme de remplacer les bouteilles en plastique par des gourdes en aluminium, l'essuie-tout papier par des pièces textiles, les cotons-tiges par des oriculi japonais, les sacs en papier par des sacs textiles, etc. Dans ce sujet nous nous concentrons sur la problématique de remplacement (de substitution) de produits jetables par des produits durables.

Un *produit* est décrit par un nom, un nombre annuel d'utilisations et un prix à l'unité. Un produit peut avoir plusieurs produits de substitution, appelés substituts. Un produit est connecté à tous ses substituts connus : si un produit *p* a un substitut *s1* qui a lui-même un substitut *s2*, *p* est connecté à *s1* et à *s2*. On distingue deux sortes de produits : les *produits jetables* (usage unique) et les *produits durables dans le temps*. Pour simplifier, la classe *Durabilite* qui représente le critère de spécialisation/généralisation n'est pas considérée dans ce modèle, et ne le sera pas dans le code. Les produits durables dans le temps sont des produits qui ont de plus une durée de vie (mesurée en années). On peut calculer un coût annuel pour chaque produit. Le coût annuel d'un produit jetable se calcule comme la multiplication du nombre annuel d'utilisations par le prix à l'unité, tandis que le coût annuel d'un produit durable dans le temps se calcule comme la division du prix à l'unité par la durée de vie. Un échange relie un produit initial et un substitut : on peut évaluer l'économie réalisée lors de cet échange par la différence de leurs coûts annuels. Elle peut être négative (resp. positive), dans ce cas, l'utilisation du substitut revient plus cher (resp. moins cher) sur une année que l'utilisation du produit initial. Pour un produit, on cherche à trouver les substituts les plus économiques (de coût annuel le plus faible). Les méthodes ne sont pas présentées sur le diagramme.

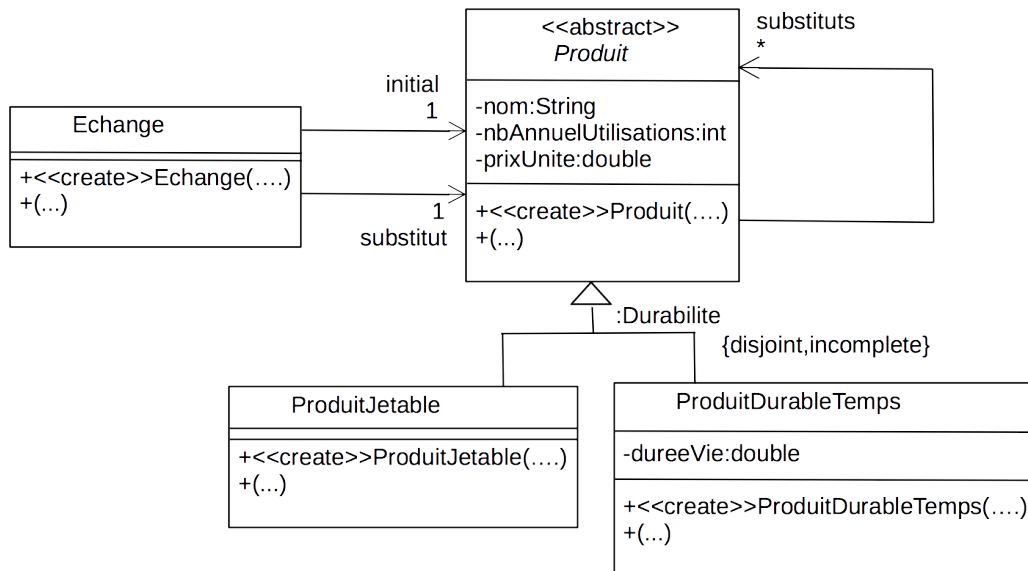


FIGURE 1 – Produits et échanges

Question 1. Dessiner le diagramme d'instances d'un échange entre un coton tige et un oriculi japonais (petit bâtonnet de bambou). Un coton tige est un produit jetable dont le prix à l'unité est 0,01 euros et le nombre d'utilisations annuel est 300. Un oriculi japonais est un produit durable dont le prix à l'unité est 3 euros, le nombre d'utilisations annuel est 300 et la durée de vie est de 5 ans. L'oriculi japonais est un substitut pour le coton tige. Le coton tige a un autre substitut, le cure-oreille (tige métallique avec une boucle à son extrémité), dont le prix à l'unité est de 4 euros, le nombre d'utilisations annuel est 300 et la durée de vie est de 20 ans.

Réponse à la question 1 : **Barème (1,5 points)**

0,5 pour les 4 instances, 0,5 pour les liens, 0,5 pour les valeurs des attributs.

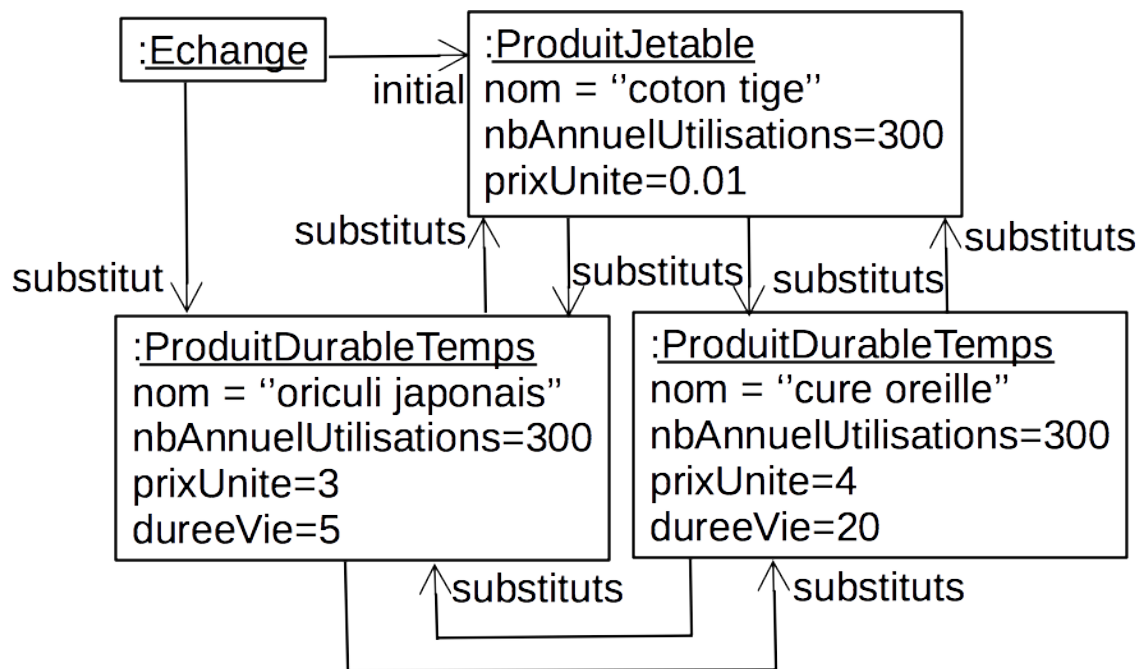


FIGURE 2 – Produits et échanges

Question 2. Ecrire l'entête et les attributs de la classe `Produit`. N'écrivez pas les accesseurs, ni les constructeurs. Dans les questions suivantes, vous pourrez considérer que vous disposez des accesseurs aux attributs qui ne sont pas des collections.

Réponse à la question 2 :

Barème (1,5 points)

```

public abstract class Produit {    //0,5

    private String nom = "";        //0,5 pour les premiers attributs

    private int nbAnnuelUtilisations;

    private int prixUnite;

    // 0,5 pour la liste (meme sans l'initialisation)
    private ArrayList<Produit> listeSubstituts = new ArrayList<>();

    ...
}
  
```

Question 3. Ecrire les entêtes des classes `ProduitJetable` et `ProduitDurableTemps`, ainsi que leurs attributs éventuels. N'écrivez pas les accesseurs, ni les constructeurs. Vous supposerez qu'ils existent pour les prochaines questions.

Réponse à la question 3 :

Barème (0,75 points)

```
// 0,25
public class ProduitJetable extends Produit {

    ...
}

// 0,5
public class ProduitDurableTemps extends Produit {

    private double dureeVie;

    ...
}
```

Question 4. Ecrire, dans la classe `Produit`, une méthode `nbSubstituts():int` retournant, pour un produit, le nombre de ses substituts connus. Vous supposerez que les accesseurs aux attributs qui ne sont pas des collections existent.

Réponse à la question 4 :

Barème (0,5 points) (seulement 0,25 si itération correcte pour calculer)

```
public int nbSubstituts() {

    return this.listeSubstituts.size();

}
```

Question 5. Ecrire, dans la classe `Produit`, une méthode `aPourSubstitut(Produit autreProduit) :boolean` retournant vrai si un objet **de même nom** que `autreProduit` apparaît dans la liste des substituts possibles de l'objet `this` qui représente le receveur de la méthode (faux sinon). Vous supposerez que les accesseurs aux attributs qui ne sont pas des collections existent.

Réponse à la question 5 :

Barème (1,5 points)

```
public boolean aPourSubstitut(Produit autreProduit) {  
  
    for (Produit p : this.listeSubstituts){ // 0,5  
  
        if (autreProduit.nom.equals(p.nom)). // 0,5  
  
            return true; // 0,5 pour les deux return bien placés  
  
    }  
  
    return false;  
}
```

Question 6. Ecrire dans les classes `Produit`, `ProduitJetable` et `ProduitDurableTemps` la méthode `coutAnnuel() :double`. Vous supposerez que les accesseurs aux attributs qui ne sont pas des collections existent.

Réponse à la question 6 :

Barème (2,5 points)

```
// Dans Produit  
  
    abstract public double coutAnnuel(); // 0,5  
  
// Dans ProduitJetable  
  
    public double coutAnnuel() {  
  
        return this.getPrixUnite()*this.getNbAnnuelUtilisations(); // 1  
    }  
  
// Dans ProduitDurableTemps  
  
    public double coutAnnuel() {  
  
        // on peut vérifier que ce n'est pas une division par zéro  
        return this.getPrixUnite()/this.dureeVie; // 1  
    }
```

Question 7. Ecrire dans les classes appropriées la méthode `toString():String`. Elle retourne une chaîne constituée du nom et du coût annuel d'utilisation pour tous les produits, et dans le cas des produits durables dans le temps, on ajoute la durée de vie. Vous supposerez que les accesseurs aux attributs qui ne sont pas des collections existent.

Réponse à la question 7 :

Barème (1,75 points)

```
// Dans Produit

public String toString() { // 0,5

    return "nom = "+this.getNom()

        +" cout annuel = "+this.coutAnnuel();

}

// Dans ProduitJetable
// on ne met rien du tout car la méthode est héritée // 0,5
// et qu'elle n'a pas de particularité
// pour les produits jetables

// Dans ProduitDurableTemps
// Ici on appelle la méthode héritée et on la raffine

public String toString() {

    return super.toString()+" "+this.dureeVie;
    // 0,5 pour super et 0,25 pour duree vie

}
```

Question 8. Ecrire, dans la (les) classe(s) appropriée(s) une méthode `substitutsRentables`. Elle retourne, pour un produit donné, ses substituts qui ont un coût annuel strictement inférieur au produit receveur de la méthode.

Réponse à la question 8 :

Barème (2 points)

```
public ArrayList<Produit> substitutsRentable() { // 0,5

    ArrayList<Produit> result = new ArrayList<>(); // 0,5

    for (Produit p : this.listeSubstituts){ // 0,25

        if (p.coutAnnuel() < this.coutAnnuel()). // 0,25

            result.add(p); // 0,25

    }

    return result; // 0,25

}
```

Question 9. Ecrire l'entête et les attributs de la classe `Echange`. N'écrivez pas les accesseurs, ni les constructeurs. Vous supposerez qu'ils existent pour la prochaine question.

Réponse à la question 9 :

Barème (0,75 points)

```
public class Echange { // 0,25 si ne derive pas bizarrement d'une autre classe

    private Produit initial;    // 0,25

    private Produit substitut;  // 0,25

    ...
}
```

Question 10. Ecrire, dans la classe `Echange` la méthode `economieRealisee():double`.

Réponse à la question 10 :

Barème (1 point)

```
public double economieRealisee() {

    return this.initial.coutAnnuel() - this.substitut.coutAnnuel();

}
```

2 Partie 2

Question 11. Proposez un diagramme de classes UML pour représenter la situation suivante. Un réseau ferroviaire a un nom et deux méthodes pour calculer respectivement sa longueur en km et son émission annuelle de CO₂ (en grammes de CO₂ par voyageur et par km). Il est constitué de plus de deux lignes qui peuvent être partagées entre plusieurs réseaux. Une ligne a une longueur en km. Elle relie une gare de départ à une gare d'arrivée et traverse plusieurs autres gares. Certaines lignes sont électrifiées, tandis que d'autres ne le sont pas. Les lignes électrifiées se composent de câbles conducteurs. Chaque câble conducteur a une longueur et un type de courant qui est une valeur de l'ensemble {continu, alternatif-industriel, alternatif-basse-fréquence}. Un câble conducteur n'appartient qu'à une ligne. Ne mettez pas de constructeurs, ni de noms d'associations, mais indiquez les noms des rôles et la multiplicité, le critère de spécialisation/généralisation et les contraintes associées, parmi disjoint, recouvrement, complet, incomplet.

Réponse à la question 11 :

Barème (6,75 point)

- enum : 0,25
- 6 classes, méthodes, attributs (sauf le power type) : 0,25 chacune = 1,5
- association reseau – > ligne (avec multiplicité et noms de rôles) : 0,5 ; on peut accepter un attribut bien fait = 0,25 (idem ci-dessous)
- association ligne – > gares desservies (avec multiplicité et noms de rôles) : 0,5 ; on peut accepter un attribut bien fait
- association ligne – > gare départ (avec multiplicité et noms de rôles) : 0,5 ; on peut accepter un attribut bien fait
- association ligne – > gare arrivée (avec multiplicité et noms de rôles) : 0,5 ; on peut accepter un attribut bien fait
- association ligne électrifiée – > câble (avec multiplicité et noms de rôles) : 0,5 ; on peut accepter un attribut bien fait
- attribut ou association câble – > type de courant (avec multiplicité et noms de rôles) : 0,5
- spécialisation : 0,5
- power type Electrification : 0,5 (0,25 si au moins le nom)
- contraintes : 0,5
- association Ligne – > Electrification : 0,5

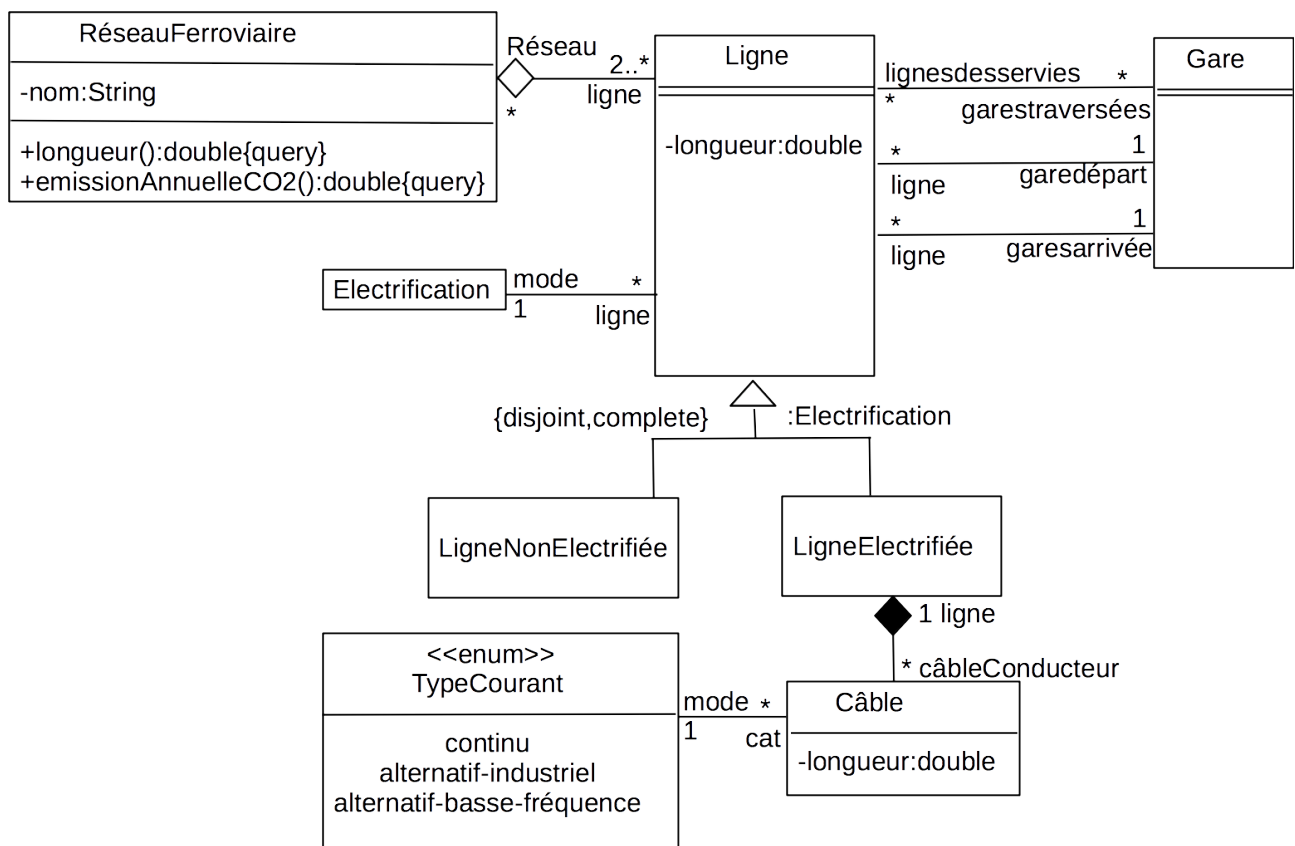


FIGURE 3 – Réseau ferroviaire