IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Contrôle – INFO-RES-S6

UE Informatique et Réseaux – INFO-RES-S6

26 mai 2025

Prénom:

Nom:

Consignes

Merci de répondre directement sur le sujet dans les cases prévues. Si vous avez besoin d'une feuille supplémentaire, écrivez votre prénom, votre nom et la référence du contrôle : UE Info-Réseaux.

Cet exercice est prévu pour durer 0h45.

Les documents papiers et les dispositifs électroniques (ordinateur, téléphone portable, ...) sont **interdits** pendant l'examen à **l'exception d'une feuille A4 recto-verso**.

Tout échange (qu'il soit vocal, écrit, électronique, ...) sera considéré comme de la fraude.

Nous vous conseillons de lire l'ensemble de l'énoncé de l'exercice avant de commencer.

Merci d'avance et bon travail !

Exercice 1 (*Boutique en ligne*)

Le but de cet exercice est de modéliser une application simple de boutique en ligne sur Internet et développer une partie du système. Un acheteur se connecte à une boutique en ligne via une interface web (le contenu et le fonctionnement de l'interface web ne nous intéressent pas dans le cadre de cet exercice).

Il existe deux catégories d'acheteurs :

- les acheteurs ordinaires et
- les acheteurs privilégiés qui bénéficient de points de fidélité (*avantages*) à chaque achat.

Lorsqu'un acheteur se connecte, il dispose d'un panier. L'acheteur peut alors effectuer les opérations suivantes :

1. sélectionner un produit (parmi les produits existants),
2. ajouter le produit sélectionné à son panier,

3. payer les produits présents dans le panier. Notez qu'un acheteur privilégié peut payer comme un acheteur ordinaire ou payer l'intégralité du montant en utilisant les points de fidélité accumulés lors de précédents achats.

Pour simplifier le problème, nous supposerons que :

— la gestion des acheteurs et des produits présents dans la boutique (par exemple ajout d'un nouveau produit, suppression d'un acheteur) ne fait pas partie des fonctionnalités que vous devez concevoir,
— la connexion, la notion de session et de système externe (par exemple les banques) ne sont pas à traiter dans le cadre de ce problème,
— un acheteur ne possède qu'un seul panier (pendant une séance d'achats),
— un panier n'est associé qu'à un seul acheteur.

▷ **Question 1.1 :**
Identifiez les acteurs et proposez un diagramme de cas d'utilisation du système.

Dans le système que nous désirons modéliser, une boutique dispose d'une URL (*Uniform Resource Locator*) stockée sous forme d'une `String`. Un acheteur dispose d'un nom. Un produit dispose d'un nom et d'un prix en euro. Tout acheteur a un panier qui est vidé chaque fois qu'il paye. Quand l'acheteur paye ses achats, la liste des produits qu'il a achetés s'affiche avec le total à payer.

▷ **Question 1.2 :**
Identifiez les classes nécessaires à la réalisation, leur(s) association(s) et les éventuelles relations d'héritage. Vous présenterez votre réponse sous forme d'un diagramme de classe UML simple (c'est-à-dire avec les attributs mais sans méthode) contenant toutes les annotations (rôle, cardinalités,...) nécessaires.

▷ **Question 1.3 :**
Dans la question précédente, avez-vous réifié la notion de panier (c'est-à-dire introduit une classe spécifique `Panier`) ? Justifiez en 1 ou 2 phrases votre réponse.

▷ **Question 1.4 :**
Programmez en Java la classe `Panier` en complétant le code à trou ci-dessous. Un extrait de la documentation de la classe `ArrayList` est donné en annexe A.

Attention, si vous utilisez d'autres classes identifiées à la question 1.2, il est inutile d'en détailler le code.

```
public class Panier {
private List<Produit> produits;

public Panier() {
    // à compléter



}

public int getNombreProduits() {
    // à compléter
```

```
}

public void ajouterProduit(Produit p) {
    // à compléter

}

public float totalAPayer() {
    // à compléter

}

public String toString() {
    // à compléter

}
```

```
  public static void main(String[] args) {
    Panier p = new Panier();
    p.ajouterProduit(new Produit("carotte",2.5f));
    System.out.println(p.toString());
    p.ajouterProduit(new Produit("courgette",1.5f));
    System.out.println(p.getNombreProduits());
    System.out.println(p.totalAPayer());
  }
}
```

Annexes

A Extrait de la javadoc de la classe `java.util.ArrayList`

Class ArrayList<E>

Constructor Summary
`ArrayList()` Constructs an empty list.

Method Summary	
`void`	`add(int index, E element)` Inserts the specified element at the specified position in this list.
`boolean`	`add(E o)` Appends the specified element to the end of this list.
`E`	`get(int index)` Returns the element at the specified position in this list.
`int`	`size()` Returns the number of elements in this list.