



### Exercice 2 :

```
interface ICoffee {
    public void serveCoffee(CoffeeContext context);
}

class Coffee implements ICoffee {
    private final String flavor;

    public Coffee(String newFlavor) {
        this.flavor = newFlavor;
        System.out.println("Coffee is created! - " + flavor);
    }

    public String getFlavor() {
        return this.flavor;
    }

    public void serveCoffee(CoffeeContext context) {
        System.out.println("Serving " + flavor + " to table " +
context.getTable());
    }
}

class CoffeeContext {
    private final int tableNumber;
}
```

```
    public CoffeeContext(int tableNumber) {
        this.tableNumber = tableNumber;
    }

    public int getTable() {
        return this.tableNumber;
    }
}

class CoffeeFactory {

    private HashMap<String, Coffee> flavors = new HashMap<String, Coffee>();

    public Coffee getCoffeeFlavor(String flavorName) {
        Coffee flavor = flavors.get(flavorName);
        if (flavor == null) {
            flavor = new Coffee(flavorName);
            flavors.put(flavorName, flavor);
        }
        return flavor;
    }

    public int getTotalCoffeeFlavorsMade() {
        return flavors.size();
    }
}

public class Waitress {
    //coffee array
    private static Coffee[] coffees = new Coffee[20];
    //table array
    private static CoffeeContext[] tables = new CoffeeContext[20];
    private static int ordersCount = 0;
    private static CoffeeFactory coffeeFactory;

    public static void takeOrder(String flavorIn, int table) {
        coffees[ordersCount] = coffeeFactory.getCoffeeFlavor(flavorIn);
        tables[ordersCount] = new CoffeeContext(table);
        ordersCount++;
    }

    public static void main(String[] args) {
        coffeeFactory = new CoffeeFactory();

        takeOrder("Cappuccino", 2);

        for (int i = 0; i < ordersCount; ++i) {
            coffees[i].serveCoffee(tables[i]);
        }

        System.out.println("\nTotal Coffee objects made: " +
            coffeeFactory.getTotalCoffeeFlavorsMade());
    }
}
```

## Exercice 3 :

A partir de la liste d'exigences exprimées lors du TD1, donnez le diagramme de classes pour la gestion des enseignants, des étudiants, des cours et du catalogue.

### 1. Le système gère des enseignants et des étudiants

#### 1.1. Le système gère des **enseignants**

1.1.1. Un enseignant est **décrit** par son identifiant, nom, prénom, mail

1.1.1.1. Le nom, prénom, mail sont représentés par des chaînes de 50 caractères

1.1.1.2. L'ID est représenté par EXXXX (avec X un entier)

1.1.2. Un enseignant peut être responsable d'un ensemble de cours

1.1.3. Un enseignant peut être responsable d'un ensemble de cursus

#### 1.2. Le système gère des **étudiants**

1.2.1. Un étudiant est décrit par son identifiant, nom, prénom, mail

1.2.2. Un étudiant est affecté à un cursus

1.3. La connexion d'un utilisateur au système doit se faire en moins de 15 secondes.

1.4. La connexion se fait via une interface avec l'annuaire LDAP du système d'information de l'école

### 2. Le système permet la création d'un ensemble de cours

#### 2.1. Le système permet la **création** d'un cours

2.1.1. Chaque cours est décrit par un certain nombre d'informations : le code du cours, le nom, le nom de l'enseignant responsable, le cursus et le résumé du cours.

2.1.2. Un cours n'est affecté qu'à un seul responsable

2.1.3. Seuls les enseignants responsables du cursus peuvent créer des cours dans un cursus

2.1.4. La création d'un cours ne peut se faire que par saisie des informations (code, nom et enseignant responsable)

2.1.5. Avant toute création du cours, il faut choisir le cursus associé

2.2. La création d'un cours déclenche l'envoi d'un **mail** à l'enseignant responsable

2.3. Seul l'enseignant responsable peut saisir le **résumé** du cours.

2.3.1. Le résumé du cours peut être modifié après saisie mais uniquement avant génération du catalogue

### 3. Le système permet la **génération du catalogue**

3.1. Seule la scolarité peut générer le catalogue

3.2. La génération bloque la modification du résumé par les enseignants responsables

3.3. Le catalogue ne peut être généré si tous les cours n'ont pas été complétés.

3.4. Le catalogue de l'année précédente est archivé lors de la génération