OOAD Assignment – 9

Name : Siri Gowri H
SRN : PES1UG21CS599

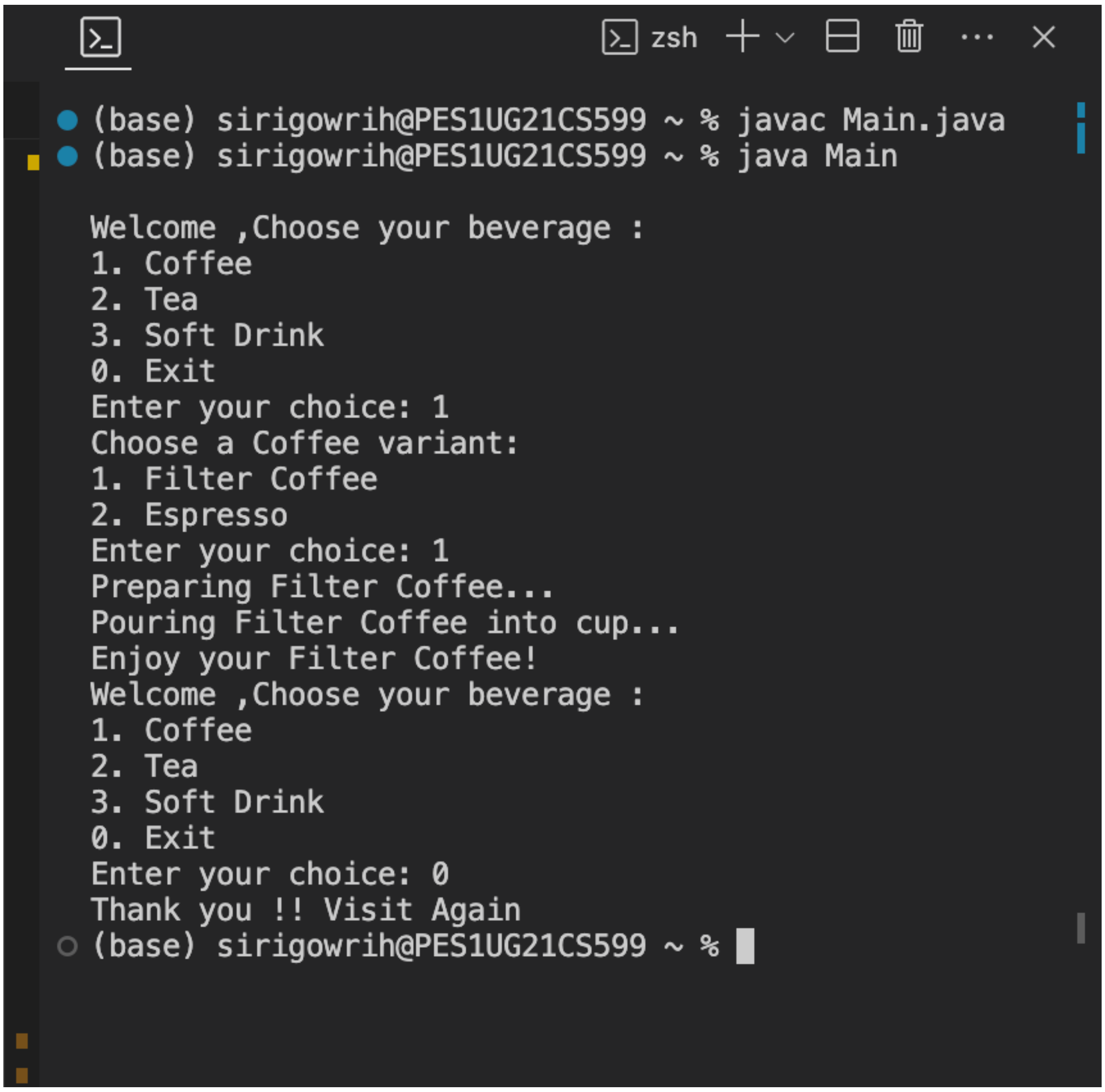
Implementing a Beverage Vending Machine using Abstract Factory Pattern

Code:

```java
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

// Abstract Product Interface for beverages
interface Beverage {
    void prepare();
    void pour();
    void dispense();
}

// Concrete Product: Coffee
class Coffee implements Beverage {
    private final String name;

    public Coffee(String name) {
        this.name = name;
    }

    @Override
    public void prepare() {
        System.out.println("Preparing " + name + "...");
    }

    @Override
    public void pour() {
        System.out.println("Pouring " + name + " into cup...");
    }

    @Override
    public void dispense() {
        System.out.println("Enjoy your " + name + "!");
    }
}

// Concrete Product: Tea
class Tea implements Beverage {
    private final String name;

    public Tea(String name) {
        this.name = name;
    }

    @Override
    public void prepare() {
        System.out.println("Brewing " + name + "...");
    }

    @Override
    public void pour() {
        System.out.println("Pouring " + name + " into cup...");
    }

    @Override
    public void dispense() {
        System.out.println("Enjoy your " + name + "!");
    }
}

// Concrete Product: Soft Drink
class SoftDrink implements Beverage {
    private final String name;

    public SoftDrink(String name) {
        this.name = name;
    }

    @Override
    public void prepare() {
        System.out.println("Mixing " + name + "...");
    }

    @Override
    public void pour() {
        System.out.println("Pouring " + name + " into cup...");
    }

    @Override
    public void dispense() {
        System.out.println("Enjoy your " + name + "!");
    }
}
```

```java
// Abstract Factory Interface for creating beverages
interface BeverageFactory {
    Beverage createBeverage(String variant);
}

// Concrete Factory: Coffee Factory
class CoffeeFactory implements BeverageFactory {
    @Override
    public Beverage createBeverage(String variant) {
        return new Coffee(variant);
    }
}

// Concrete Factory: Tea Factory
class TeaFactory implements BeverageFactory {
    @Override
    public Beverage createBeverage(String variant) {
        return new Tea(variant);
    }
}

// Concrete Factory: Soft Drink Factory
class SoftDrinkFactory implements BeverageFactory {
    @Override
    public Beverage createBeverage(String variant) {
        return new SoftDrink(variant);
    }
}

// Vending Machine
class VendingMachine {
    private final Map<String, Map<String, BeverageFactory>> factories;
    private final Scanner scanner;

    public VendingMachine() {
        this.factories = new HashMap<>();
        this.scanner = new Scanner(System.in);
        setupFactories();
    }

    private void setupFactories() {
        Map<String, BeverageFactory> coffeeVariants = new HashMap<>();
        coffeeVariants.put("Filter Coffee", new CoffeeFactory());
        coffeeVariants.put("Espresso", new CoffeeFactory());
        factories.put("Coffee", coffeeVariants);

        Map<String, BeverageFactory> teaVariants = new HashMap<>();
        teaVariants.put("Green Tea", new TeaFactory());
        teaVariants.put("Black Tea", new TeaFactory());
        factories.put("Tea", teaVariants);
        Map<String, BeverageFactory> softDrinkVariants = new HashMap<>();
        softDrinkVariants.put("Maaza", new SoftDrinkFactory());
        softDrinkVariants.put("Lemonade", new SoftDrinkFactory());
        factories.put("Soft Drink", softDrinkVariants);
    }

    public void dispenseBeverage(String category) {
        Map<String, BeverageFactory> categoryVariants = factories.get(category);
        if (categoryVariants != null) {
            System.out.println("Choose a " + category + " variant:");
            int index = 1;
            for (String variant : categoryVariants.keySet()) {
                System.out.println(index + ". " + variant);
                index++;
            }
            System.out.print("Enter your choice: ");
            int variantChoice = scanner.nextInt();
            String[] variants = categoryVariants.keySet().toArray(new String[0]);
            if (variantChoice >= 1 && variantChoice <= variants.length) {
                String selectedVariant = variants[variantChoice - 1];
                BeverageFactory factory = categoryVariants.get(selectedVariant);
                if (factory != null) {
                    Beverage beverage = factory.createBeverage(selectedVariant);
                    if (beverage != null) {
                        beverage.prepare();
                        beverage.pour();
                        beverage.dispense();
                        return;
                    }
                }
            }
        }
        System.out.println("Invalid category or variant.");
    }

    public void closeScanner() {
        scanner.close();
    }
}

// Main class to test the vending machine
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        VendingMachine vendingMachine = new VendingMachine();

        int choice;
        do {
            System.out.println("Welcome ,Choose your beverage :");
            System.out.println("1. Coffee");
            System.out.println("2. Tea");
            System.out.println("3. Soft Drink");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    vendingMachine.dispenseBeverage("Coffee");
                    break;
                case 2:
                    vendingMachine.dispenseBeverage("Tea");
                    break;
                case 3:
                    vendingMachine.dispenseBeverage("Soft Drink");
                    break;
                case 0:
                    System.out.println("Thank you !! Visit Again");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 0);

        vendingMachine.closeScanner();
    }
}
```

Output:



```
● (base) sirigowrih@PES1UG21CS599 ~ % javac Main.java
● (base) sirigowrih@PES1UG21CS599 ~ % java Main

  Welcome ,Choose your beverage :
  1. Coffee
  2. Tea
  3. Soft Drink
  0. Exit
  Enter your choice: 1
  Choose a Coffee variant:
  1. Filter Coffee
  2. Espresso
  Enter your choice: 1
  Preparing Filter Coffee...
  Pouring Filter Coffee into cup...
  Enjoy your Filter Coffee!
  Welcome ,Choose your beverage :
  1. Coffee
  2. Tea
  3. Soft Drink
  0. Exit
  Enter your choice: 0
  Thank you !! Visit Again
○ (base) sirigowrih@PES1UG21CS599 ~ %
```