

Design Pattern

Flyweight

```
public class CoffeeOrderContext {
    private int tableNumber;

    public CoffeeOrderContext(int tableNumber) {
        this.tableNumber = tableNumber;
    }

    public int getTableNumber() {
        return tableNumber;
    }
}

public interface ICoffeeOrder {
    void serveCoffee(CoffeeOrderContext context);
}

public class CoffeeFlavor implements ICoffeeOrder {
    private String flavor;

    public CoffeeFlavor(String flavor) {
        this.flavor = flavor;
    }

    public String getFlavor() {
        return flavor;
    }

    public void serveCoffee(CoffeeOrderContext context) {
        System.out.println("serving coffee flavor " + flavor +
            " to table number " + context.getTableNumber());
    }
}

import java.util.HashMap;
import java.util.Map;

public class CoffeeFlavorFactory {
    private Map<String,CoffeeFlavor> flavors = new HashMap<>();

    public CoffeeFlavor getCoffeeFlavor(String flavorName) {
        CoffeeFlavor flavor = flavors.get(flavorName);

        if (flavor == null) {
            flavor = new CoffeeFlavor(flavorName);
            flavors.put(flavorName,flavor);
        }

        return flavor;
    }

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

```

public class Application {
    private static CoffeeFlavor[] flavors = new CoffeeFlavor[100];
    private static CoffeeOrderContext[] tables =
        new CoffeeOrderContext[100];

    private static int ordersMade = 0;
    private static CoffeeFlavorFactory flavorFactory;

    public static void takeOrders(String flavorIn, int table) {
        flavors[ordersMade] = flavorFactory.getCoffeeFlavor(flavorIn);
        tables[ordersMade++] = new CoffeeOrderContext(table);
    }

    public static void main(String... args) {
        flavorFactory = new CoffeeFlavorFactory();

        takeOrders("Cappuccino", 2);
        takeOrders("Cappuccino", 2);
        takeOrders("Frappe", 1);
        takeOrders("Frappe", 1);
        takeOrders("Xpresso", 1);
        takeOrders("Frappe", 897);
        takeOrders("Cappuccino", 97);
        takeOrders("Cappuccino", 97);
        takeOrders("Frappe", 3);
        takeOrders("Xpresso", 3);
        takeOrders("Cappuccino", 3);
        takeOrders("Xpresso", 96);
        takeOrders("Frappe", 552);
        takeOrders("Cappuccino", 121);
        takeOrders("Xpresso", 121);

        for (int i = 0; i < ordersMade; ++i)
            flavors[i].serveCoffee(tables[i]);

        System.out.println(" ");
        System.out.println("total coffeeFlavor objects made: " +
            flavorFactory.getTotalCoffeeFlavorsMade());
    }
}

```