# CS5800 – Advanced Software Engineering

# Cal Poly Pomona

# Homework 4

# Spring 2024

## Description:

Creational Design Patterns

Name: Fidelis Prasetyo

Email: (fprasetyo@cpp.edu)

BroncoID: 015765555

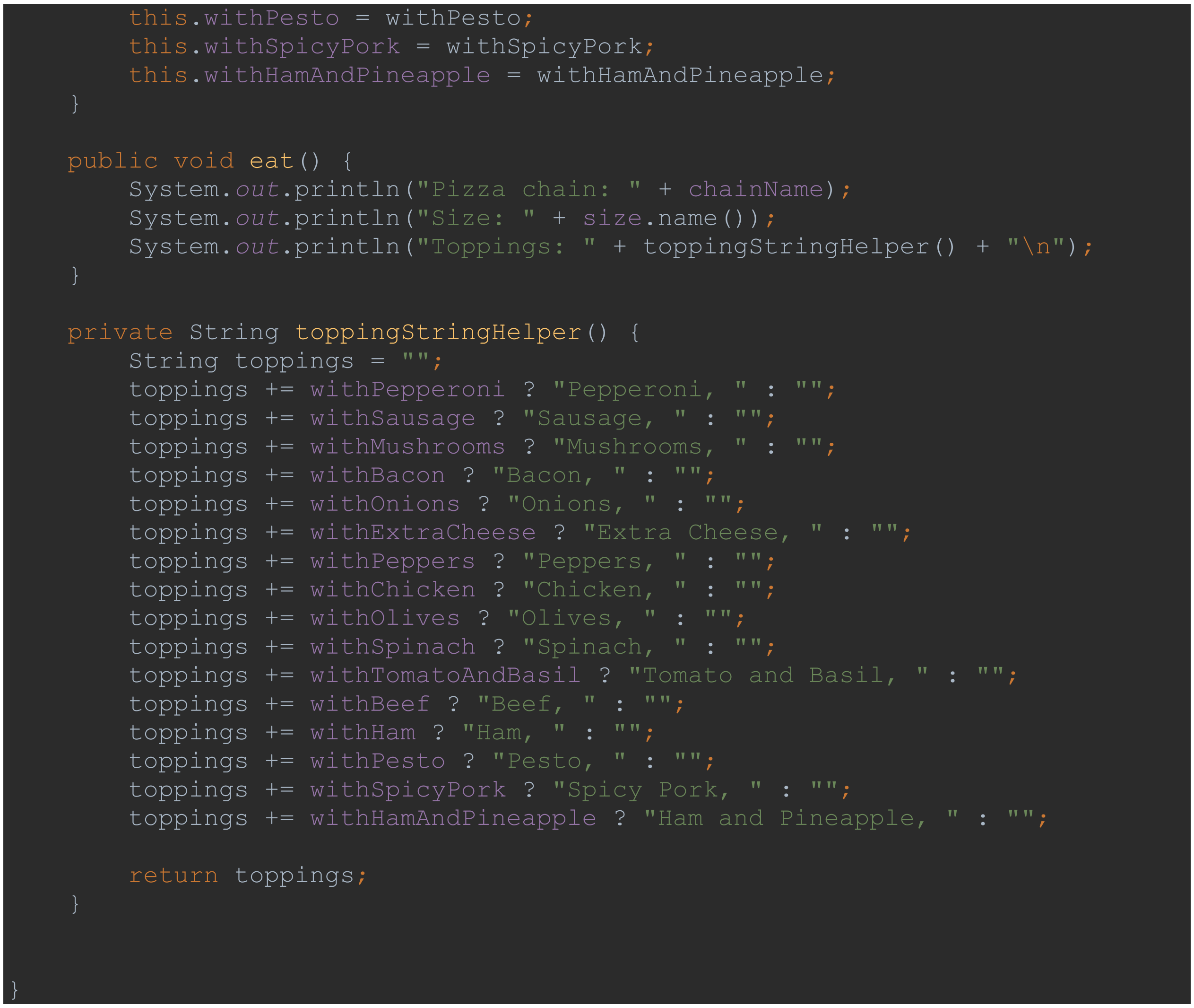
Github & Source code:

https://github.com/fidelisprasetyo/CS5800/tree/hw4

# 1. Builder

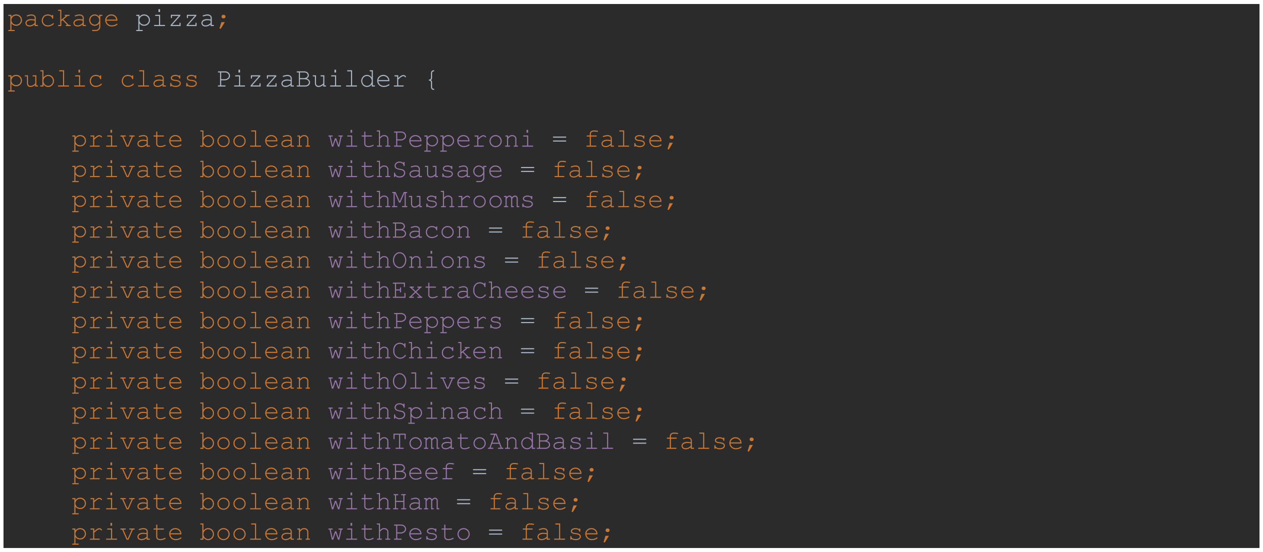
Pizza.java

```
package pizza;

public class Pizza {
    public enum Size {
        SMALL,
        MEDIUM,
        LARGE
    }
    private final boolean withPepperoni;
    private final boolean withSausage;
    private final boolean withMushrooms;
    private final boolean withBacon;
    private final boolean withOnions;
    private final boolean withExtraCheese;
    private final boolean withPeppers;
    private final boolean withChicken;
    private final boolean withOlives;
    private final boolean withSpinach;
    private final boolean withTomatoAndBasil;
    private final boolean withBeef;
    private final boolean withHam;
    private final boolean withPesto;
    private final boolean withSpicyPork;
    private final boolean withHamAndPineapple;
    private final Size size;
    private final String chainName;

    public Pizza(Size size, String chainName,
                 boolean withPepperoni, boolean withSausage, boolean
withMushrooms, boolean withBacon,
                 boolean withOnions, boolean withExtraCheese, boolean
withPeppers, boolean withChicken,
                 boolean withOlives, boolean withSpinach, boolean
withTomatoAndBasil, boolean withBeef,
                 boolean withHam, boolean withPesto, boolean withSpicyPork,
boolean withHamAndPineapple) {
        this.size = size;
        this.chainName = chainName;

        this.withPepperoni = withPepperoni;
        this.withSausage = withSausage;
        this.withMushrooms = withMushrooms;
        this.withBacon = withBacon;
        this.withOnions = withOnions;
        this.withExtraCheese = withExtraCheese;
        this.withPeppers = withPeppers;
        this.withChicken = withChicken;
        this.withOlives = withOlives;
        this.withSpinach = withSpinach;
        this.withTomatoAndBasil = withTomatoAndBasil;
        this.withBeef = withBeef;
        this.withHam = withHam;
```

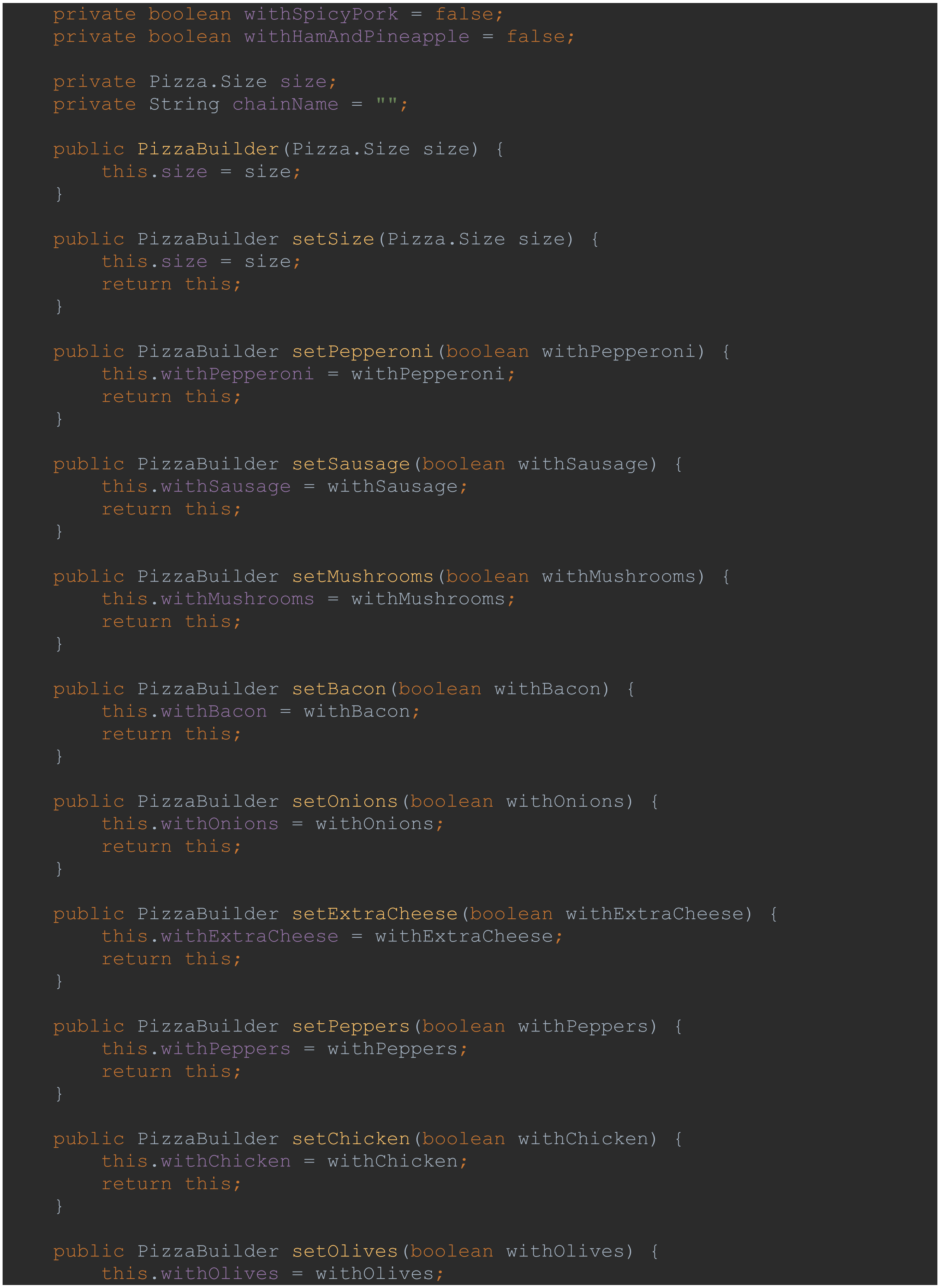


```java
        this.withPesto = withPesto;
        this.withSpicyPork = withSpicyPork;
        this.withHamAndPineapple = withHamAndPineapple;
    }

    public void eat() {
        System.out.println("Pizza chain: " + chainName);
        System.out.println("Size: " + size.name());
        System.out.println("Toppings: " + toppingStringHelper() + "\n");
    }

    private String toppingStringHelper() {
        String toppings = "";
        toppings += withPepperoni ? "Pepperoni, " : "";
        toppings += withSausage ? "Sausage, " : "";
        toppings += withMushrooms ? "Mushrooms, " : "";
        toppings += withBacon ? "Bacon, " : "";
        toppings += withOnions ? "Onions, " : "";
        toppings += withExtraCheese ? "Extra Cheese, " : "";
        toppings += withPeppers ? "Peppers, " : "";
        toppings += withChicken ? "Chicken, " : "";
        toppings += withOlives ? "Olives, " : "";
        toppings += withSpinach ? "Spinach, " : "";
        toppings += withTomatoAndBasil ? "Tomato and Basil, " : "";
        toppings += withBeef ? "Beef, " : "";
        toppings += withHam ? "Ham, " : "";
        toppings += withPesto ? "Pesto, " : "";
        toppings += withSpicyPork ? "Spicy Pork, " : "";
        toppings += withHamAndPineapple ? "Ham and Pineapple, " : "";

        return toppings;
    }


}
```

PizzaBuilder.java



```java
package pizza;

public class PizzaBuilder {

    private boolean withPepperoni = false;
    private boolean withSausage = false;
    private boolean withMushrooms = false;
    private boolean withBacon = false;
    private boolean withOnions = false;
    private boolean withExtraCheese = false;
    private boolean withPeppers = false;
    private boolean withChicken = false;
    private boolean withOlives = false;
    private boolean withSpinach = false;
    private boolean withTomatoAndBasil = false;
    private boolean withBeef = false;
    private boolean withHam = false;
    private boolean withPesto = false;
```



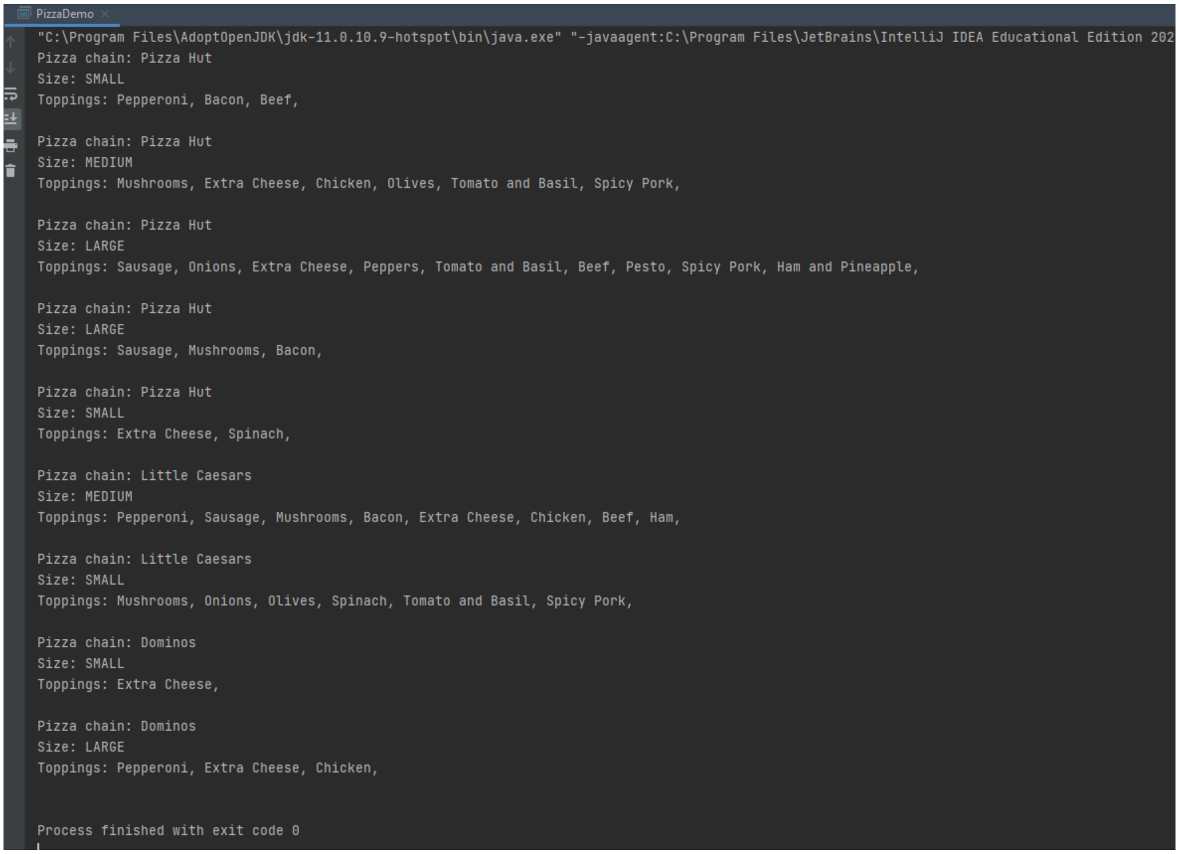
```
    private boolean withSpicyPork = false;
    private boolean withHamAndPineapple = false;

    private Pizza.Size size;
    private String chainName = "";

    public PizzaBuilder(Pizza.Size size) {
        this.size = size;
    }

    public PizzaBuilder setSize(Pizza.Size size) {
        this.size = size;
        return this;
    }

    public PizzaBuilder setPepperoni(boolean withPepperoni) {
        this.withPepperoni = withPepperoni;
        return this;
    }

    public PizzaBuilder setSausage(boolean withSausage) {
        this.withSausage = withSausage;
        return this;
    }

    public PizzaBuilder setMushrooms(boolean withMushrooms) {
        this.withMushrooms = withMushrooms;
        return this;
    }

    public PizzaBuilder setBacon(boolean withBacon) {
        this.withBacon = withBacon;
        return this;
    }

    public PizzaBuilder setOnions(boolean withOnions) {
        this.withOnions = withOnions;
        return this;
    }

    public PizzaBuilder setExtraCheese(boolean withExtraCheese) {
        this.withExtraCheese = withExtraCheese;
        return this;
    }

    public PizzaBuilder setPeppers(boolean withPeppers) {
        this.withPeppers = withPeppers;
        return this;
    }

    public PizzaBuilder setChicken(boolean withChicken) {
        this.withChicken = withChicken;
        return this;
    }

    public PizzaBuilder setOlives(boolean withOlives) {
        this.withOlives = withOlives;
```

```java
        return this;
    }

    public PizzaBuilder setSpinach(boolean withSpinach) {
        this.withSpinach = withSpinach;
        return this;
    }

    public PizzaBuilder setTomatoAndBasil(boolean withTomatoAndBasil) {
        this.withTomatoAndBasil = withTomatoAndBasil;
        return this;
    }

    public PizzaBuilder setBeef(boolean withBeef) {
        this.withBeef = withBeef;
        return this;
    }

    public PizzaBuilder setHam(boolean withHam) {
        this.withHam = withHam;
        return this;
    }

    public PizzaBuilder setPesto(boolean withPesto) {
        this.withPesto = withPesto;
        return this;
    }

    public PizzaBuilder setSpicyPork(boolean withSpicyPork) {
        this.withSpicyPork = withSpicyPork;
        return this;
    }

    public PizzaBuilder setHamAndPineapple(boolean withHamAndPineapple) {
        this.withHamAndPineapple = withHamAndPineapple;
        return this;
    }

    public PizzaBuilder setChainName(String chainName) {
        this.chainName = chainName;
        return this;
    }

    public Pizza build() {
        return new Pizza(size, chainName,
                withPepperoni, withSausage, withMushrooms, withBacon,
                withOnions, withExtraCheese, withPeppers, withChicken,
                withOlives, withSpinach, withTomatoAndBasil, withBeef,
                withHam, withPesto, withSpicyPork, withHamAndPineapple);
    }

}
```

PizzaDemo.java

```
package pizza;

public class PizzaDemo {
    public static void main(String[] args) {
        // Create a driver program to create three pizzas one of each size
with 3, 6, and 9 toppings

        Pizza pizza1 = new PizzaBuilder(Pizza.Size.SMALL).setChainName("Pizza
Hut")
                .setBacon(true)
                .setBeef(true)
                .setPepperoni(true)
                .build();

        Pizza pizza2 = new
PizzaBuilder(Pizza.Size.MEDIUM).setChainName("Pizza Hut")
                .setChicken(true)
                .setMushrooms(true)
                .setOlives(true)
                .setTomatoAndBasil(true)
                .setExtraCheese(true)
                .setSpicyPork(true)
                .build();

        Pizza pizza3 = new PizzaBuilder(Pizza.Size.LARGE).setChainName("Pizza
Hut")
                .setTomatoAndBasil(true)
                .setOnions(true)
                .setSausage(true)
                .setExtraCheese(true)
                .setSpicyPork(true)
                .setHamAndPineapple(true)
                .setPeppers(true)
                .setBeef(true)
                .setPesto(true)
                .build();
        pizza1.eat();
        pizza2.eat();
        pizza3.eat();

        // Assume you purchased another two pizza chains, Little Caesars, and
Dominos.

        Pizza hutPizzaLarge = new
PizzaBuilder(Pizza.Size.LARGE).setChainName("Pizza Hut")
                .setBacon(true)
                .setMushrooms(true)
                .setSausage(true)
                .build();

        Pizza hutPizzaSmall = new
PizzaBuilder(Pizza.Size.SMALL).setChainName("Pizza Hut")
                .setExtraCheese(true)
                .setSpinach(true)
                .build();

        Pizza caesarsPizzaMed = new
PizzaBuilder(Pizza.Size.MEDIUM).setChainName("Little Caesars")
                .setPepperoni(true)
                .setBeef(true)
                .setBacon(true)
                .setChicken(true)
                .setHam(true)
                .setMushrooms(true)
                .setExtraCheese(true)
                .setSausage(true)
                .build();

        Pizza caesarsPizzaSmall = new
PizzaBuilder(Pizza.Size.SMALL).setChainName("Little Caesars")
                .setMushrooms(true)
                .setSpicyPork(true)
                .setSpinach(true)
                .setOlives(true)
                .setOnions(true)
                .setTomatoAndBasil(true)
                .build();

        Pizza dominosPizzaSmall = new
PizzaBuilder(Pizza.Size.SMALL).setChainName("Dominos")
                .setExtraCheese(true)
                .build();

        Pizza dominosPizzaLarge = new
PizzaBuilder(Pizza.Size.LARGE).setChainName("Dominos")
                .setChicken(true)
                .setPepperoni(true)
                .setExtraCheese(true)
                .build();

        hutPizzaLarge.eat();
        hutPizzaSmall.eat();
        caesarsPizzaMed.eat();
        caesarsPizzaSmall.eat();
        dominosPizzaSmall.eat();
        dominosPizzaLarge.eat();
    }
}
```

Output:



```
"C:\Program Files\AdoptOpenJDK\jdk-11.0.10.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Educational Edition 202
Pizza chain: Pizza Hut
Size: SMALL
Toppings: Pepperoni, Bacon, Beef,

Pizza chain: Pizza Hut
Size: MEDIUM
Toppings: Mushrooms, Extra Cheese, Chicken, Olives, Tomato and Basil, Spicy Pork,

Pizza chain: Pizza Hut
Size: LARGE
Toppings: Sausage, Onions, Extra Cheese, Peppers, Tomato and Basil, Beef, Pesto, Spicy Pork, Ham and Pineapple,

Pizza chain: Pizza Hut
Size: LARGE
Toppings: Sausage, Mushrooms, Bacon,

Pizza chain: Pizza Hut
Size: SMALL
Toppings: Extra Cheese, Spinach,

Pizza chain: Little Caesars
Size: MEDIUM
Toppings: Pepperoni, Sausage, Mushrooms, Bacon, Extra Cheese, Chicken, Beef, Ham,

Pizza chain: Little Caesars
Size: SMALL
Toppings: Mushrooms, Onions, Olives, Spinach, Tomato and Basil, Spicy Pork,

Pizza chain: Dominos
Size: SMALL
Toppings: Extra Cheese,

Pizza chain: Dominos
Size: LARGE
Toppings: Pepperoni, Extra Cheese, Chicken,


Process finished with exit code 0
```

## 2. Factory

Customer.java

```java
package macronutrients;

public class Customer {

    private String name;
    private Diet diet;

    public Customer(String name, Diet diet) {
        this.name = name;
        this.diet = diet;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Diet getDiet() {
        return diet;
    }
    public void setDiet(Diet diet) {
        this.diet = diet;
    }
}
```

Diet.java

```
package macronutrients;

public enum Diet {
    NO_RESTRICTION("No restriction"),
    PALEO("Paleo"),
    VEGAN("Vegan"),
    NUT_ALLERGY("Nut allergy");

    private final String string;

    Diet(String string) {
        this.string = string;
    }

    @Override
    public String toString() {
        return string;
    }
}
```

Carbs.java

```
package macronutrients;

public abstract class Carbs {
    abstract String getName();
}
```

Cheese.java

```
package macronutrients;

public class Cheese extends Carbs {

    @Override
    public String getName() {
        return "Cheese";
    }
}
```

Bread.java

```
package macronutrients;

public class Bread extends Carbs {
```

```java
    @Override
    public String getName() {
        return "Bread";
    }
}
```

Lentils.java

```java
package macronutrients;

public class Lentils extends Carbs {

    @Override
    public String getName() {
        return "Lentils";
    }
}
```

Pistachio.java

```java
package macronutrients;

public class Pistachio extends Carbs {

    @Override
    public String getName() {
        return "Pistachio";
    }
}
```

Protein.java

```java
package macronutrients;

public abstract class Protein {
    abstract String getName();
}
```

Fish.java

```java
package macronutrients;

public class Fish extends Protein {

    @Override
    public String getName() {
        return "Fish";
    }

}
```

Chicken.java

```java
package macronutrients;

public class Chicken extends Protein {

    @Override
    public String getName() {
        return "Chicken";
    }
}
```

Beef.java

```java
package macronutrients;

public class Beef extends Protein {

    @Override
    public String getName() {
        return "Beef";
    }
}
```

Tofu.java

```java
package macronutrients;

public class Tofu extends Protein {

    @Override
    public String getName() {
        return "Tofu";
    }
}
```

Fats.java

```java
package macronutrients;

public abstract class Fats {
    abstract String getName();
}
```

Avocado.java

```java
package macronutrients;

public class Avocado extends Fats {

    @Override
    public String getName() {
        return "Avocado";
    }
}
```

SourCream.java

```java
package macronutrients;

public class SourCream extends Fats {

    @Override
    public String getName() {
        return "Sour Cream";
    }
}
```

Tuna.java

```java
package macronutrients;

public class Tuna extends Fats {

    @Override
    public String getName() {
        return "Tuna";
    }
}
```

Peanuts.java

```java
package macronutrients;

public class Peanuts extends Fats {

    @Override
    public String getName() {
        return "Peanuts";
    }
}
```

CarbsFactory.java

```java
package macronutrients;

public class CarbsFactory {

    private static CarbsFactory carbsFactory = null;

    private CarbsFactory(){}

    public static CarbsFactory getInstance() {
        if (carbsFactory == null) {
            carbsFactory = new CarbsFactory();
        }
        return carbsFactory;
    }

    public Carbs createCarbs(String type) {
        switch (type) {
            case "cheese":
                return new Cheese();
            case "bread":
                return new Bread();
            case "lentils":
                return new Lentils();
            case "pistachio":
                return new Pistachio();
            default:
                return null;
        }
    }
}
```

ProteinFactory.java

```
package macronutrients;

public class ProteinFactory {

    private static ProteinFactory proteinFactory = null;

    private ProteinFactory(){}

    public static ProteinFactory getInstance() {
        if (proteinFactory == null) {
            proteinFactory = new ProteinFactory();
        }
        return proteinFactory;
    }

    public Protein createProtein(String type) {
        switch (type) {
            case "fish":
                return new Fish();
            case "chicken":
                return new Chicken();
            case "beef":
                return new Beef();
            case "tofu":
                return new Tofu();
            default:
                return null;
        }
    }

}
```

FatsFactory.java

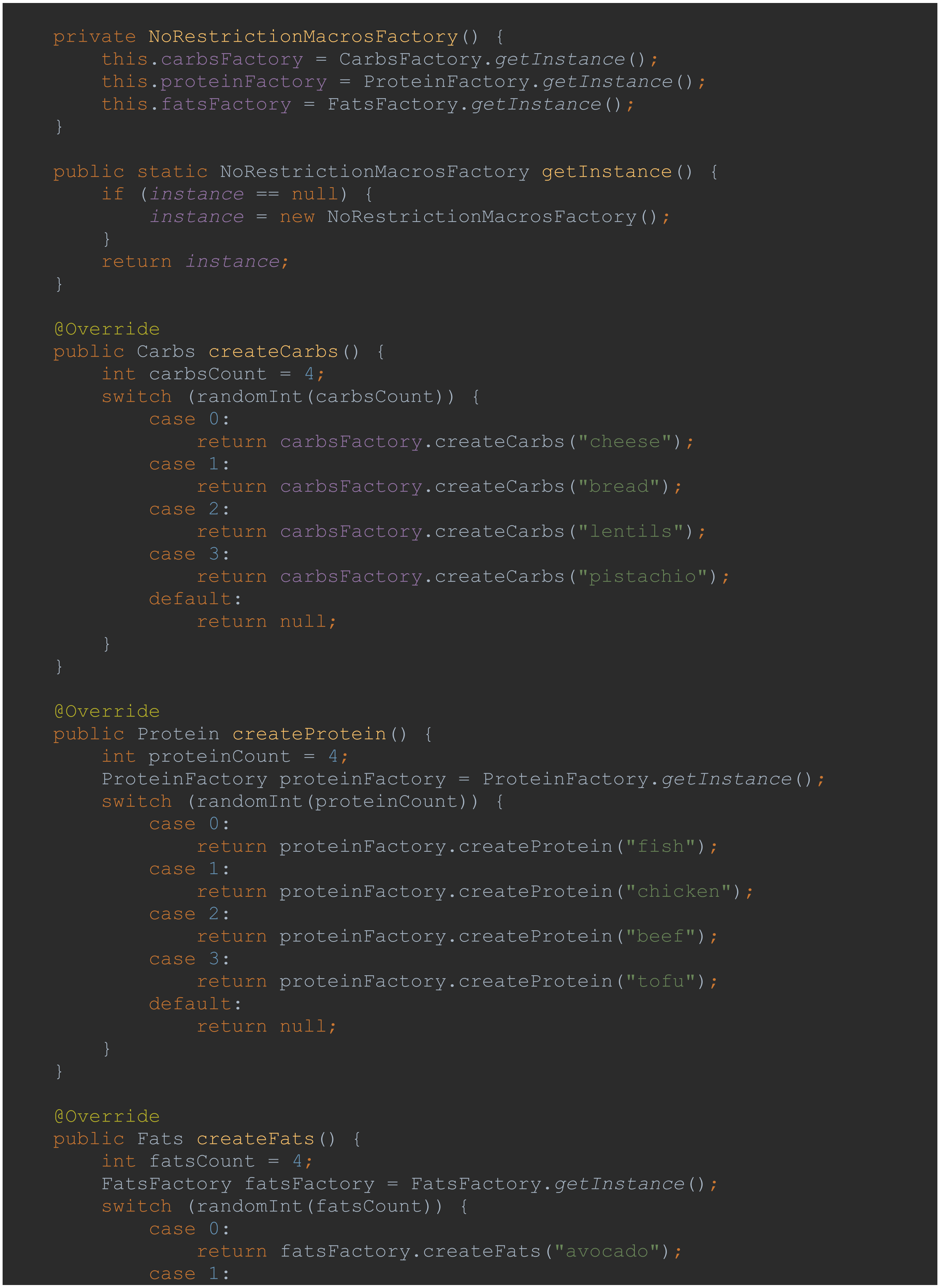
```
package macronutrients;

public class FatsFactory {

    private static FatsFactory fatsFactory = null;
    private FatsFactory(){}

    public static FatsFactory getInstance() {
        if (fatsFactory == null) {
            fatsFactory = new FatsFactory();
        }
        return fatsFactory;
    }

    public Fats createFats(String type) {
        switch (type) {
            case "avocado":
                return new Avocado();
            case "sour cream":
                return new SourCream();
            case "tuna":
                return new Tuna();
            case "peanuts":
                return new Peanuts();
            default:
                return null;
        }
    }
}
```

MacrosFactory.java

```
package macronutrients;

import java.util.Random;

public abstract class MacrosFactory {

    protected CarbsFactory carbsFactory;
    protected ProteinFactory proteinFactory;
    protected FatsFactory fatsFactory;

    abstract Carbs createCarbs();
    abstract Protein createProtein();
    abstract Fats createFats();

    protected int randomInt(int caseCount) {
        Random random = new Random();
        return random.nextInt(caseCount);
    }
}
```

NoRestrictionMacrosFactory.java

```
package macronutrients;

public class NoRestrictionMacrosFactory extends MacrosFactory {

    private static NoRestrictionMacrosFactory instance = null;
```



```java
    private NoRestrictionMacrosFactory() {
        this.carbsFactory = CarbsFactory.getInstance();
        this.proteinFactory = ProteinFactory.getInstance();
        this.fatsFactory = FatsFactory.getInstance();
    }

    public static NoRestrictionMacrosFactory getInstance() {
        if (instance == null) {
            instance = new NoRestrictionMacrosFactory();
        }
        return instance;
    }

    @Override
    public Carbs createCarbs() {
        int carbsCount = 4;
        switch (randomInt(carbsCount)) {
            case 0:
                return carbsFactory.createCarbs("cheese");
            case 1:
                return carbsFactory.createCarbs("bread");
            case 2:
                return carbsFactory.createCarbs("lentils");
            case 3:
                return carbsFactory.createCarbs("pistachio");
            default:
                return null;
        }
    }

    @Override
    public Protein createProtein() {
        int proteinCount = 4;
        ProteinFactory proteinFactory = ProteinFactory.getInstance();
        switch (randomInt(proteinCount)) {
            case 0:
                return proteinFactory.createProtein("fish");
            case 1:
                return proteinFactory.createProtein("chicken");
            case 2:
                return proteinFactory.createProtein("beef");
            case 3:
                return proteinFactory.createProtein("tofu");
            default:
                return null;
        }
    }

    @Override
    public Fats createFats() {
        int fatsCount = 4;
        FatsFactory fatsFactory = FatsFactory.getInstance();
        switch (randomInt(fatsCount)) {
            case 0:
                return fatsFactory.createFats("avocado");
            case 1:
                return fatsFactory.createFats("sour cream");
            case 2:
                return fatsFactory.createFats("tuna");
            case 3:
                return fatsFactory.createFats("peanuts");
            default:
                return null;
        }
    }
}
```

PaleoMacrosFactory.java

```
package macronutrients;

public class PaleoMacrosFactory extends MacrosFactory {

    private static PaleoMacrosFactory instance = null;

    private PaleoMacrosFactory() {
        this.carbsFactory = CarbsFactory.getInstance();
        this.proteinFactory = ProteinFactory.getInstance();
        this.fatsFactory = FatsFactory.getInstance();
    }

    public static PaleoMacrosFactory getInstance() {
        if (instance == null) {
            instance = new PaleoMacrosFactory();
        }
        return instance;
    }

    @Override
    public Carbs createCarbs() {
        return carbsFactory.createCarbs("pistachio");
    }

    @Override
    public Protein createProtein() {
        int PROTEIN_COUNT = 3;
        switch (randomInt(PROTEIN_COUNT)) {
            case 0:
                return proteinFactory.createProtein("fish");
            case 1:
                return proteinFactory.createProtein("chicken");
            case 2:
                return proteinFactory.createProtein("beef");
            default:
                return null;
        }
    }

    @Override
    public Fats createFats() {
        int FATS_COUNT = 3;
        switch (randomInt(FATS_COUNT)) {
            case 0:
                return fatsFactory.createFats("avocado");
            case 1:
                return fatsFactory.createFats("tuna");
            case 2:
                return fatsFactory.createFats("peanuts");
            default:
                return null;
        }
    }
}
```

VeganMacrosFactory.java

```
package macronutrients;

public class VeganMacrosFactory extends MacrosFactory {

    private static VeganMacrosFactory instance = null;

    private VeganMacrosFactory() {
        this.carbsFactory = CarbsFactory.getInstance();
        this.proteinFactory = ProteinFactory.getInstance();
        this.fatsFactory = FatsFactory.getInstance();
    }

    public static VeganMacrosFactory getInstance() {
        if (instance == null) {
            instance = new VeganMacrosFactory();
        }
        return instance;
    }

    @Override
    public Carbs createCarbs() {
        int carbsCount = 3;
        switch (randomInt(carbsCount)) {
            case 0:
                return carbsFactory.createCarbs("bread");
            case 1:
                return carbsFactory.createCarbs("lentils");
            case 2:
                return carbsFactory.createCarbs("pistachio");
            default:
                return null;
        }
    }

    @Override
    public Protein createProtein() {
        return proteinFactory.createProtein("tofu");
    }

    @Override
    public Fats createFats() {
        int fatsCount = 2;
        switch (randomInt(fatsCount)) {
            case 0:
                return fatsFactory.createFats("avocado");
            case 1:
                return fatsFactory.createFats("peanuts");
            default:
                return null;
        }
    }
}
```

NoNutMacrosFactory.java

```
package macronutrients;

public class NoNutMacrosFactory extends MacrosFactory {

    private static NoNutMacrosFactory instance = null;

    private NoNutMacrosFactory() {
        this.carbsFactory = CarbsFactory.getInstance();
        this.proteinFactory = ProteinFactory.getInstance();
        this.fatsFactory = FatsFactory.getInstance();
    }

    public static NoNutMacrosFactory getInstance() {
        if (instance == null) {
            instance = new NoNutMacrosFactory();
        }
        return instance;
    }

    @Override
    public Carbs createCarbs() {
        int carbsCount = 3;
        switch (randomInt(carbsCount)) {
            case 0:
                return carbsFactory.createCarbs("cheese");
            case 1:
                return carbsFactory.createCarbs("bread");
            case 2:
                return carbsFactory.createCarbs("lentils");
            default:
                return null;
        }
    }

    @Override
    public Protein createProtein() {
        int proteinCount = 4;
        switch (randomInt(proteinCount)) {
            case 0:
                return proteinFactory.createProtein("fish");
            case 1:
                return proteinFactory.createProtein("chicken");
            case 2:
```

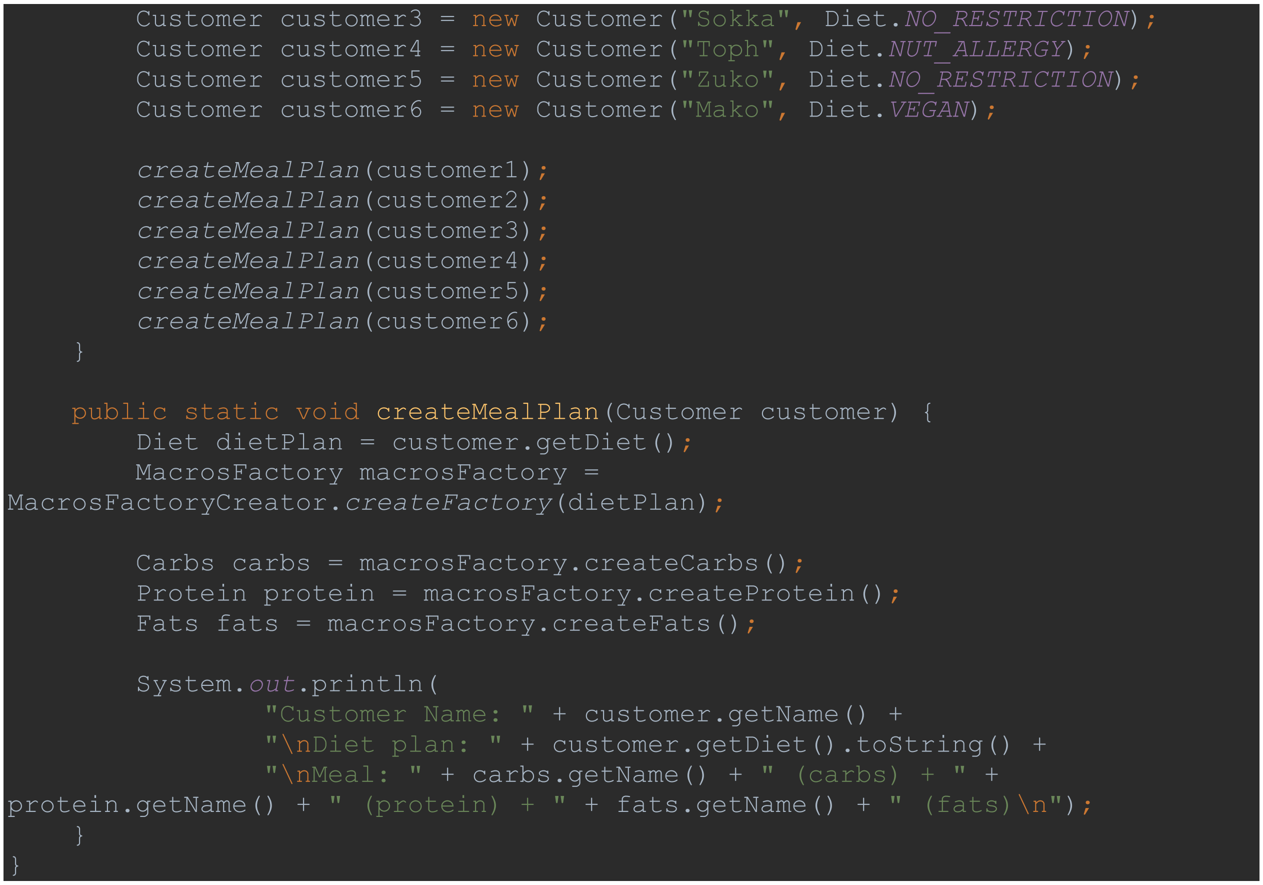
```java
                return proteinFactory.createProtein("beef");
            case 3:
                return proteinFactory.createProtein("tofu");
            default:
                return null;
        }
    }

    @Override
    public Fats createFats() {
        int fatsCount = 3;
        switch (randomInt(fatsCount)) {
            case 0:
                return fatsFactory.createFats("avocado");
            case 1:
                return fatsFactory.createFats("sour cream");
            case 2:
                return fatsFactory.createFats("tuna");
            default:
                return null;
        }
    }
}
```

MacrosFactoryCreator.java

```java
package macronutrients;

public class MacrosFactoryCreator {

    public static MacrosFactory createFactory(Diet diet) {
        switch (diet) {
            case NO_RESTRICTION:
                return NoRestrictionMacrosFactory.getInstance();
            case PALEO:
                return PaleoMacrosFactory.getInstance();
            case VEGAN:
                return VeganMacrosFactory.getInstance();
            case NUT_ALLERGY:
                return NoNutMacrosFactory.getInstance();
            default:
                System.out.println("Specify a valid diet plan");
                return null;
        }
    }
}
```

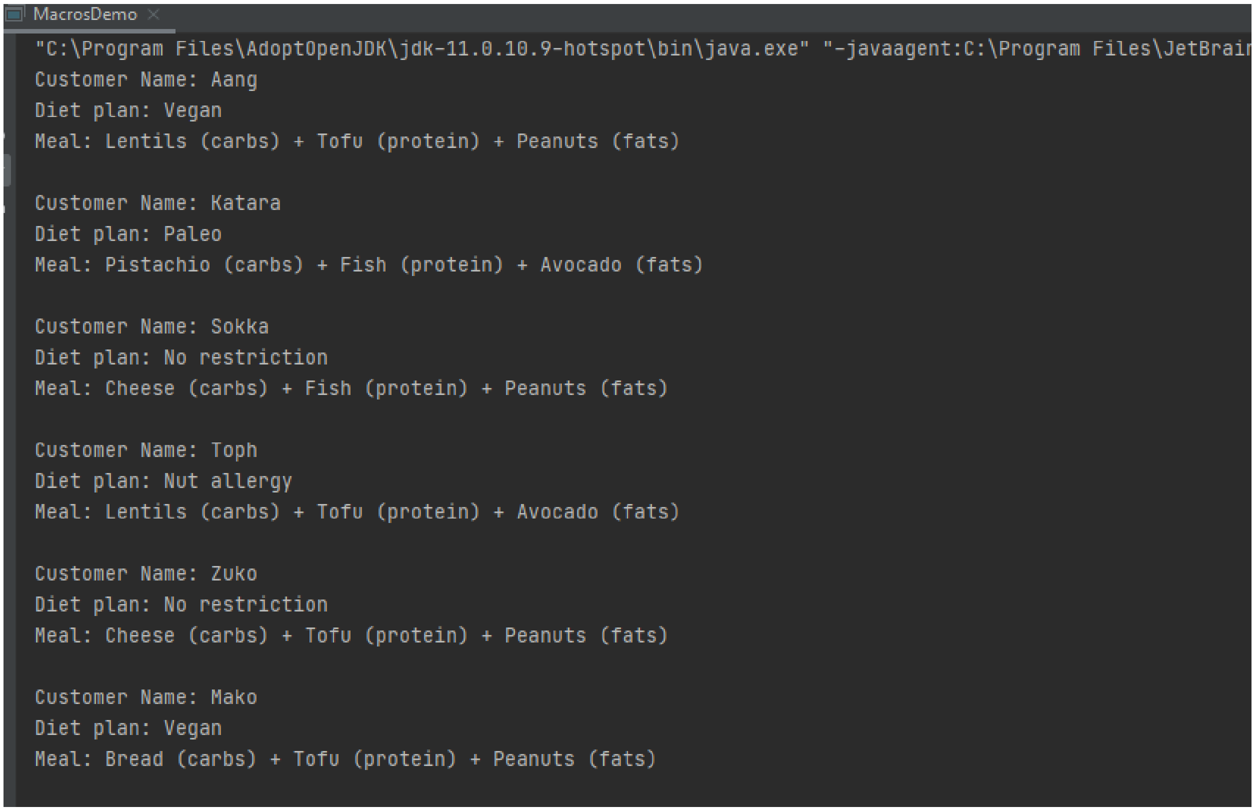
MacrosDemo.java

```java
package macronutrients;

public class MacrosDemo {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Aang", Diet.VEGAN);
        Customer customer2 = new Customer("Katara", Diet.PALEO);
```



```
        Customer customer3 = new Customer("Sokka", Diet.NO_RESTRICTION);
        Customer customer4 = new Customer("Toph", Diet.NUT_ALLERGY);
        Customer customer5 = new Customer("Zuko", Diet.NO_RESTRICTION);
        Customer customer6 = new Customer("Mako", Diet.VEGAN);

        createMealPlan(customer1);
        createMealPlan(customer2);
        createMealPlan(customer3);
        createMealPlan(customer4);
        createMealPlan(customer5);
        createMealPlan(customer6);
    }

    public static void createMealPlan(Customer customer) {
        Diet dietPlan = customer.getDiet();
        MacrosFactory macrosFactory =
MacrosFactoryCreator.createFactory(dietPlan);

        Carbs carbs = macrosFactory.createCarbs();
        Protein protein = macrosFactory.createProtein();
        Fats fats = macrosFactory.createFats();

        System.out.println(
                "Customer Name: " + customer.getName() +
                "\nDiet plan: " + customer.getDiet().toString() +
                "\nMeal: " + carbs.getName() + " (carbs) + " +
protein.getName() + " (protein) + " + fats.getName() + " (fats)\n");
    }
}
```

Output:



```
"C:\Program Files\AdoptOpenJDK\jdk-11.0.10.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrair
Customer Name: Aang
Diet plan: Vegan
Meal: Lentils (carbs) + Tofu (protein) + Peanuts (fats)

Customer Name: Katara
Diet plan: Paleo
Meal: Pistachio (carbs) + Fish (protein) + Avocado (fats)

Customer Name: Sokka
Diet plan: No restriction
Meal: Cheese (carbs) + Fish (protein) + Peanuts (fats)

Customer Name: Toph
Diet plan: Nut allergy
Meal: Lentils (carbs) + Tofu (protein) + Avocado (fats)

Customer Name: Zuko
Diet plan: No restriction
Meal: Cheese (carbs) + Tofu (protein) + Peanuts (fats)

Customer Name: Mako
Diet plan: Vegan
Meal: Bread (carbs) + Tofu (protein) + Peanuts (fats)
```

## Source Code & Supporting Files

The entire source code and other supporting documents/ files can be obtained from this GitHub repository:

https://github.com/fidelisprasetyo/CS5800/tree/hw4