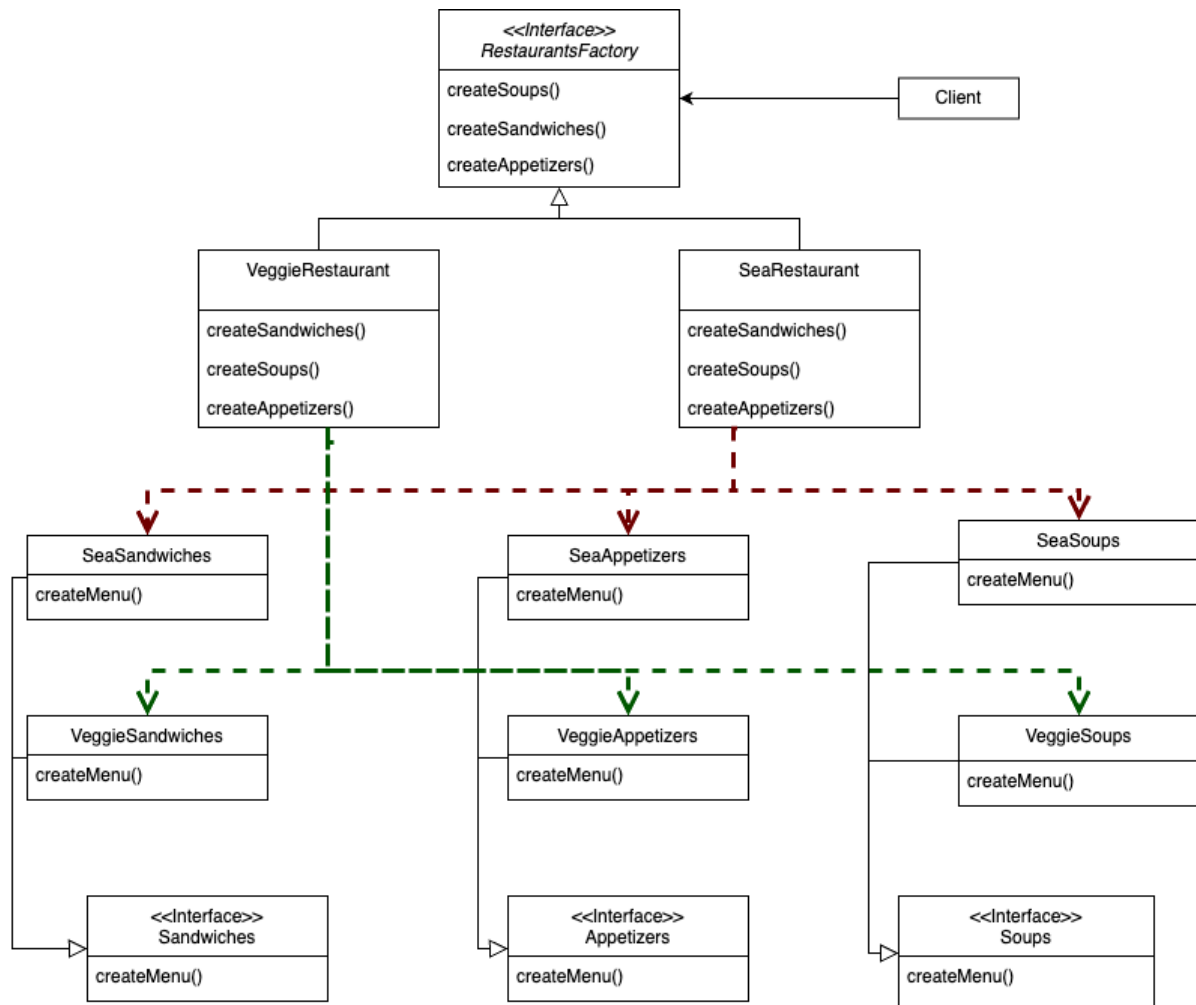


## Restaurants Factory - ADS

By: Cesar Mamani



## Restaurants Factory - ADS

### CODE:

```
public class FactoryPatternLab {
    public static void main(String[] args) {
        RestaurantFactory seaRestaurant = getSeaRestaurant(); // P2I
        System.out.println("\n----Sea Restaurant---");
        Soups soup = seaRestaurant.createSoups(); // POLY
        Sandwiches sandwich = seaRestaurant.createSandwiches(); // POLY
        System.out.println(soup);
        System.out.println("\t"+soup.createMenu()); // POLY
        System.out.println(sandwich);
        System.out.println("\t"+sandwich.createMenu()); // POLY
        System.out.println("\n----Veggie Restaurant---");
        RestaurantFactory veggieRestaurant = getVeggieRestaurant(); // P2I
        soup = veggieRestaurant.createSoups(); // POLY
        sandwich = veggieRestaurant.createSandwiches(); // POLY
        System.out.println(soup);
        System.out.println("\t"+soup.createMenu()); // POLY
        System.out.println(sandwich);
        System.out.println("\t"+sandwich.createMenu()); // POLY
    }
    static RestaurantFactory getSeaRestaurant() { // Simple Factory
        return new SeaRestaurant(); // P2C
    }
    static RestaurantFactory getVeggieRestaurant() { // Simple Factory
        return new VeggieRestaurant(); // P2C
    }
}
```

### Interfaces Factory

```
public interface RestaurantFactory {
    Sandwiches createSandwiches();
    Soups createSoups();
    Appetizers createAppetizers();
}
```

### Concretes Factories

```
public class SeaRestaurant implements RestaurantFactory {
    @Override
    public Sandwiches createSandwiches() {
        return new SeaSandwiches();
    }
    @Override
    public Soups createSoups() {
        return new SeaSoups();
    }
    @Override
    public Appetizers createAppetizers() {
        return new SeaAppetizers();
    }
}
```

## Restaurants Factory - ADS

```
public class VeggieRestaurant implements RestaurantFactory {
    @Override
    public Sandwiches createSandwiches() {
        return new VeggieSandwiches();
    }
    @Override
    public Soups createSoups() {
        return new VeggieSoups();
    }
    @Override
    public Appetizers createAppetizers() {
        return new VeggieAppetizers();
    }
}
```

### Interfaces Products

```
public interface Sandwiches {
    List<Food> createMenu();
}
```

```
public interface Soups {
    List<Food> createMenu();
}
```

```
public interface Appetizers {
    List<Food> createMenu();
}
```

### Concretes Sandwiches Products

```
public class VeggieSandwiches implements Sandwiches {
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Cucumber Sandwich"));
    }
    @Override
    public String toString() {
        return "Veggie Sandwich";
    }
}
public class SeaSandwiches implements Sandwiches{
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Shrimp Sandwich"));
    }
    @Override
    public String toString() {
        return "Sea Sandwich";
    }
}
```

## Restaurants Factory - ADS

### Concretes Soups Products

```
public class SeaSoups implements Soups{
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Parihuela Peruvian Soup"));
    }
    @Override
    public String toString() { return "Sea Soups"; }
}

public class VeggieSoups implements Soups{
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Quinoa Soup"));
    }
    @Override
    public String toString() {
        return "Veggie Soups";
    }
}
```

### Concretes Appetizers Products

```
public class VeggieAppetizers implements Appetizers {
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Solterito Peruvian"));
    }
    @Override
    public String toString() {
        return "Veggie Appetizers";
    }
}

public class SeaAppetizers implements Appetizers{
    @Override
    public List<Food> createMenu() {
        return List.of(Food.getFood("Shrimp Spice"));
    }
    @Override
    public String toString() {
        return "Sea Appetizers";
    }
}
```

## Restaurants Factory - ADS

Entity Additional;

```
public class Food {
    List<String> getIngredients;
    String description;
    List<String> photosUrl;
    Double price;
    String name;

    protected Food(List<String> getIngredients, String description,
List<String> photosUrl, Double price) {
        this.getIngredients = getIngredients;
        this.description = description;
        this.photosUrl = photosUrl;
        this.price = price;
    }

    protected Food(String name) {
        this.name = name;
        this.price = 0.0;
    }

    static Food getFood(String name){ //Simple Factory
        return new Food(name);
    }

    @Override
    public String toString() {
        return String.format("Food: %s, price: %s", name,price);
    }
}
```