HACETTEPE UNIVERSİTY COMPUTER ENGİNEERİNG

BBM 104 PROGRAMMİNG ASSİGNMENT-3

REPORT

EXPLANATİON OF MY CODE AND MY DETAİLED DESİGN

What is our aim in this assignment : In this experiment we working on a pizza shop and its order system.We need to keep orders for each customer according to command which are taken from input.txt file.
Each order may contain two types of pizza which are called Napolitan and AmericanPan. sausages, salami, onions and soft drink can be added these pizzas but we need to add each of these additions cost to plain pizzas cost. Right here we need to use decorative design pattern. We also need to keep this order information and customer information in a database and right here we need to use data access object design pattern. To sum up according to information which is taken from input file we need do necessary operations and make the output file as well.
In this experiment we will learn the structure of a class, how classes interact,usage of inheritance and polymorphism. At the same time we will learn usage of interface and abstract class.

Explanation of my code and design:I used data Access object design pattern for managing my data.
I need to keep two types of data in this experiment.Customer and order.So I created a customer class which is contain proporties of customers.

```
public class Customer {

    protected int id;
    protected String name;
    protected String surname;
    protected String phoneNumber;
    protected String adress;
    protected int orderId;
```

I create an interface for customers which is contains necassery methods.After that ı override and used these abstract method ,according to input files ,in my concrete class.

```
import java.util.List;


public interface CustomerDao {
    public void add(Customer customer);

    public void update(int id, Customer customer);

    public void delete(int id);

    public Customer[] getAll();
    public int findCustomerIndex(int id);

}
```

Then ı created an concrete class and implements this interface.

```
public class CustomerDaoImpl implements CustomerDao{
```

İn this class ı created an Arraylist which is type of Customer and keep all customer object in this.then ı override every method which are taken from CustomerDao interface.Then According to input file, ı used these methods.

I'v done the same operations for orders.Because orders also data type which is ı need to keep.

So I created a customer class which is contain proporties of customers.
additionally the order class holds an arraylist of pizza type.this arraylist hold every order for a customer .we can think it like order basket.

```
public class Order {
    protected int orderId;
    protected int customerId;
    protected int drink;
    ArrayList<Pizza> basket;
```

I create an interface for orders which is contains necassery methods.

```
public interface OrderDao {

    public void createOrder(Order order);
    public void addPizza(int orderId,Pizza newPizza);
    public void addDrink();
    public double PayCheck();

}
```

Then ı created an concrete class and implements this interface.

```
public class OrderDaoImpl implements OrderDao {
```

İn this class ı created an Arraylist which is type of Orders and keep all order object in this. Then ı override every method which are taken from CustomerDao interface.Then According to input file, ı used these methods.

İn order part we can add additional ingredient to plain pizza.Every additional ingredient has its own price.After every addition to plain pizza we have to add this additional ingredients cost top lain pizzas cost.ı used decorative design pattern right here.so ı created a Pizza interface which is include name and cost methods.

```
public interface Pizza {

    double cost();
    String name();

}
```

Then ı created Npolitan and AmericanPan class which are implements this interface.Plain Napolitan pizzas price is 10 $ ı override the cost method and it is returns 10 and ı also

override name method it returns “Napolitan”. Plain AmericanPan pizzas price is 5 $ ı override the cost method and it is returns 5 and ı also override name method it returns “AmericanPan”.

```
public class Napoliten implements Pizza {
    int cost=10;
    public double cost() {

        return 10;
    }

    public String name() {

        return "napolitan";
    }


}
```

```
public class Americano implements Pizza {

    public double cost() {

        return 5;
    }

    public String name() {

        return "americano";
    }


}
```

Then I created abstract decorator class implementing the *Pizaa* interface.

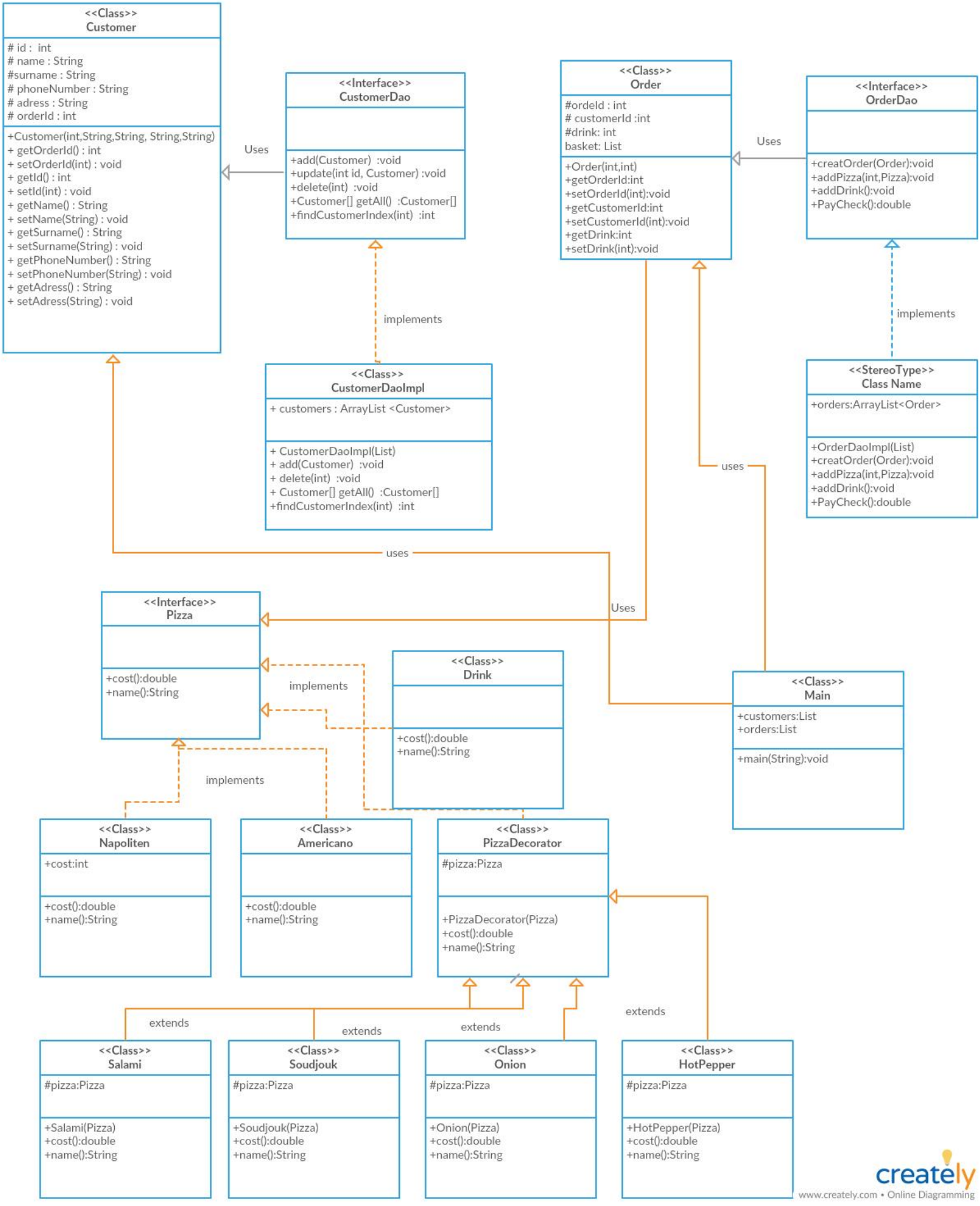
This class has a variable which is type of pizza and override name and cost methods. Then ı create concrete class which is extends this abstract class for every additional ingredient.Each class override cost and name methods.when overriding these methods every additional ingredient return cost+its own price and name+its own name.So when we call these class object in others object ve can add cost and name on plain pizzas name and cost.

After designing and coding all of these we have to read input and and fulfill the necessary commands.ı open the input file in main class.ı also open the empty customer,order and output files with writing mode.when ı reading the input files assign the each line in a string variable and create a while loop.This loop Works if the string variable is not equal to null.

Finally according to each line in the input file, I did the necessary operations and printed the these empty files

.

UML DİAGRAM OF MY DESİGN İS BELOW



Tolga Furkan GÜLER

21692874