

# **EXPERIMENT 3**

Name: Muhammed Said

Surname: Kaya

**Student Number**: 21627428

Section: 03

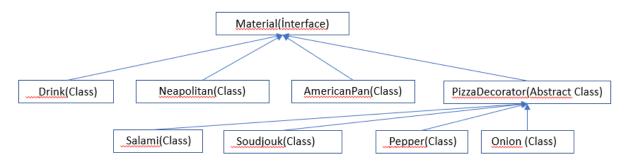
**Programing Language**: JAVA

**Subject**: Inheritance, Polymorphism, OOP

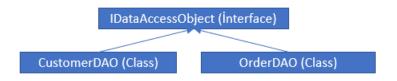
#### **Explanation of Classes**

There is exactly sixteen classes with main class . Also there is two interface in this code.

For Decorator Pattern part: (İgnore shape of the arrows)



For Data Access Object part: ( ignore shape of the arrows )



### **Explanation of Code**

In the main class , some methods is works step by step with ReadWrite class.I mean, in the main class ReadWrite.inputTxt() is example for this methods. Also in ReadWrite class , There are private objects ( IDataAccessObject customers = new CustomerDAO() , IDataAccessObject orders = new OrderDAO() ) . I use them in order to change databases , when we compare according to input.txt . If you ask me , When databases is created and where ? The Databases is created when objects is created which I mentioned above.

An object, which from CustomerDAO, is created. It's reference is IDataAccessObject .Also , every object , which is produced from CustomerDAO , has an ArrayList<Object> as private. It's name is customers. We can reach this ArrayList with getCustomers() and setCustomers() . After that again, an object, which from OrderDAO, is created. It's reference is IDataAccessObject. Also , every object , which is produced from OrderDAO , has an ArrayList<Object> as private.It's name is orderArray. We can reach this ArrayList with getOrderArray() and setOrderArray() . Also int OrderDAO and CustomerDAO implements IDataAccesObject so OrderDAO and CustomerDAO includes the methods which in IDataAccesObject . Right now we have databases, which is empty.

In the main class, ReadWrite.inputTxt() works and according to commands, which in input.txt, databases is changed.Right now If I have to mention working system of decorator pattern, Every order has an basket of materials, customer. addPizza and

addDrink methods. İn addPizza method, An array, which includes string, comes as argument. in for loop, I have reach classes, whose name is same with strings that in array, and their objects is created after that assigned to Material pizza. Lastly pizza is created (AmericanPan Salami ......).

If you aks me, How can we reach databases with using the objects (customers and orders which in ReadWrite class), When we say customers.getAll(), we can reach. Right now we changed the databases. Also When Changes are done, summary of that changes is adding to outputArray which an ArrayList includes String.

In the main class ,ReadWrite.outputTxt() works and the informations ,which in outputArray , is printed to output.txt. ReadWrite.customerTxtUpdate() works and the customer database is printed to customer.txt . With ReadWrite.orderTxtUpdate() woks and the order database is printed. Finally the code is terminated.

## **Explanation of Algorithms**

- \* Create empty databases(arrays)
- \* Make comparisons and fill the databases
- \* The changes are suppressed.
- \* The databases are suppressed.

#### **UML DIAGRAM** <<Class>> Main <<Interface>> <<Class>> IDataAccessObject ReadWrite + main ( args : String[] ) : void customers: IDataAccessObject orders : IDataAccessObject outputArray : ArrayList<String> + getBvID (ID: int): Object + inputTxt (filename : String ) : void + deleteByID ( ID : int ) : String + outputTxt (filename : String) : void + add ( object : Object ) : boolean + customerTxtUpdate ( filename : String ) : void + getAll (): ArrayList<Object> + orderTxtUpdate (filename: String): void + update (): void <<Class>> <<Class>> <<Class>> Order CustomerDAO OrderDAO <<Class>> orderID: int customers : ArrayList<Object> Customer orderArray: ArrayList<Object> orderPrice : int id:int customer: Customer name : String Basket: ArrayList<Material> + getByID ( ID : int ) : Object + getByID ( ID : int ) : Object surname : String pizza: Material + deleteByID (ID : int ) : String + deleteByID (ID: int): String phone : String drink : Material + add ( object : Object ) : boolean address : String + add ( object : Object ) : boolean constructor: Constructor + getAll (): ArrayList<Object> + getAll (): ArrayList<Object> class1: Class<?> + getld () : int + update (): void + update (): void + getName () : String + update ( a : int) : void + getOrderID (): int + getOrderArray (): ArrayList<Object> + getSurname () : String + getCustomers(): ArrayList<Object> + getOrderPrice (): int + setOrderArray ( orderArray : ArrayList<Object> ) : void + getPhone () : String + setCustomers( customerArray : ArrayList<Object> ) : void + getCustomer () : Customer + getAddress () : String + getBasket () : ArrayList<Material> + setId (id : int) : void + getAddress () : String + setName (name : String) : void + setOrderID (orderID : int) : void + setSurname (surname : String) : void + setOrderPrice (orderPrice : int) : void + setPhone (phone : String) : void + setCustomer (customer : Customer) : void + setAddress(address : String) : void + setBasket (basket : ArrayList<Material>) : void + toString (): String + addPizza ( array : String[] , count : int ) : Material + addDrink (): void + toString (): String <<Class>> Salami <<Class>> + getCost () : int <<Class>> <<Interface>> getName() : String Material PizzaDecorator Soudjouk pizza:Material + getCost () : int + getCost () : int + getCost () : int + getName() : String + getName() : String + getName() : String <<Class>> <<Class>> <<Class>> <<Class>> <<Class>> Drink Neapolitan AmericanPan Onion Pepper + getCost (): int + getCost (): int + getCost (): int + getCost (): int + getCost (): int + getName() : String + getName() : String getName() : String + getName() : String + getName() : String www.creately.com . Online Diagramming