**PizzaRestaurant**

This is a simple console application for a university assignment for ordering virtual pizza. A client can order different pizzas from different restaurants and put some toppings on the pizza if he wants.

**Commands**

* **chooseRestaurant()** - a client can choose the restaurant he wants to order from by typing the restaurant's number.
* **choosePizza()** - then, he will be given a menu for the chosen restaurant and he can select his preferred pizza.
* **addToppings()** - if the client wants, he can add different toppings to his pizza.

# Classes and public methods

**Client**

* **Client() –** create a object of class Client.
* **void showRestaurants()** – shows the available restaurants for the client. It doesn’t accept any arguments and returns void.
* **void makeOrder()** – represents the process of ordering pizza. It doesn’t accept any arguments and returns void.

**PizzaRestaurant**

* **PizzaRestaurant()** – default constructor for an abstract class PIzzaRestaurant.
* **virtual Pizza\* createPizza(std::string pizzaName) = 0** – pure virtual method that will be overwritten in the derivatives. It accepts a pizza name as a string as an argument and returns a pointer of type Pizza, created from the pizza name.
* **virtual Pizza\* addTopping(Pizza\* pizza, std::string topping)** – adds a topping to the pizza. It accepts a pizza of type Pizza\*, which is the pizza to which the topping will be added and a topping as a string which is the topping’s name. It returns an object of type Pizza\* which is the decorated pizza.
* **virtual void showMenu() = 0** – pure virtual method that will be overwritten in the derivatives. It shows the menu for a concrete restaurant. It doesn’t accept any arguments and returns void.
* **virtual void showToppings()** – shows the available toppings for the pizza. It doesn’t accept any arguments and returns void.

**DominosRestaurant**

* **DominosRestaurant() –** creates an object of class DominosRestaurant.
* **virtual Pizza\* createPizza(std::string pizzaName) override** – creates a pizza. It accepts a pizza name as a string as an argument and returns a pointer of type Pizza, created from the pizza name.
* **virtual void showMenu() override –** shows the menu for the restaurant. It doesn’t accept any arguments and returns void.

**MrPizzaRestaurant**

* **MrPizzaRestaurant() –** creates and object of class MrPizzaRestaurant.
* **virtual Pizza\* createPizza(std::string pizzaName) override** – creates a pizza. It accepts a pizza name as a string as an argument and returns a pointer of type Pizza, created from the pizza name.
* **virtual void showMenu() override –** shows the menu for the restaurant. It doesn’t accept any arguments and returns void.

**Pizza**

* **Pizza() = default –** default constructor for an abstract class Pizza.
* **virtual std::string getDescription() = 0** – pure virtual method that will be overwritten in the derivatives. Returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() = 0** – pure virtual method that will be overwritten in the derivatives. Returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**DominosHawai**

* **DominosHawai() –** creates an object of class DominosHawai.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**DominosMargaritta**

* **DominosMargaritta() –** creates an object of class DominosMargaritta.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**DominosPeperoni**

* **DominosPeperoni() –** creates an object of class DominosPeperoni.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**MrPizzaHawai**

* **MrPizzaHawai() –** creates an object of class MrPizzaHawai.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**MrPizzaMargaritta**

* **MrPizzaMargaritta() –** creates an object of class MrPizzamargaritta.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**MrPizzaPeperoni**

* **MrPizzaPeperoni() –** creates an object of class MrPizzaPeperoni.
* **virtual std::string getDescription() override** – returns the description of the pizza as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**ToppingsDecorator**

* **ToppingsDecorator() = default –** default constructor for an abstract class ToppingsDecorator.
* **virtual std::string getDescription() = 0** – pure virtual method that will be overwritten in the derivatives. Returns the description of the topping as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() = 0** – pure virtual method that will be overwritten in the derivatives. Returns the price of the pizza. It doesn’t accept any arguments and returns a type of double.

**DijonMustard**

* **DijonMustard() –** default constructor for the creation of the object of class DijonMustard.
* **DijonMustard(Pizza\* pizza)** – constructs a pizza with the topping. It accepts a pointer to Pizza (Pizza\*) as an argument.
* **virtual std::string getDescription() override** – returns the description of the topping which consists of pizza and a topping as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the new price of the pizza when the topping is added. It doesn’t accept any arguments and returns a type of double.

**HeinzKetchup**

* **HeinzKetchup() –** default constructor for the creation of the object of class HeinzKetchup.
* **HeinzKetchup(Pizza\* pizza)** – constructs a pizza with the topping. It accepts a pointer to Pizza (Pizza\*) as an argument.
* **virtual std::string getDescription() override** – returns the description of the topping which consists of pizza and a topping as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the new price of the pizza when the topping is added. It doesn’t accept any arguments and returns a type of double.

**GarlicSauce**

* **GarlicSauce() –** default constructor for the creation of the object of class GarlicSauce.
* **GarlicSauce(Pizza\* pizza)** - constructs a pizza with the topping. It accepts a pointer to Pizza (Pizza\*) as an argument.
* **virtual std::string getDescription() override** – returns the description of the topping which consists of pizza and a topping as a string. It doesn’t accept any arguments and returns a type of string.
* **virtual double getPrice() override** – returns the new price of the pizza when the topping is added. It doesn’t accept any arguments and returns a type of double.