

Description:

The second workshop lets the student practice about designing a bit more complicated JavaFx screen. Create a **Pizza Ordering Application** which helps you to practice different design layouts and model classes.

Note: For designing the front end you can see example in Week 2 lectures like grouping and container.

Task:

You are being hired by a company which design and deliver the ultimate software solutions to their clients. Your job is to design and deliver to one of their client who runs a pizza shop a desktop application which will help the client not only to manage the orders also to store them in a data structure.

Requirements:

- Following the MVC design pattern you are required to create model classes in this project which will deal with multiple model classes like Customer and Order at the minimum.
 - a. Pizza Size (Fix the price for each Pizza size)
(Idea: This can be a ChoiceBox or Radio Buttons)
 - i Small - \$7.00
 - ii Medium - \$10.00
 - iii Large - \$13.00
 - iv Extra Large - \$15.00
 - b. Crust Type
(Idea: This can be a ChoiceBox or Radio Buttons)
 - i Normal
 - ii Thin
 - iii Deep Dish
 - c. Toppings (Each topping is going to cost extra \$1.10)
(Idea: This can be a Check Boxes)
 - i Pineapple
 - ii Extra Cheese
 - iii Dried Shrimps
 - iv Mushrooms
 - v Anchovies
 - vi Sun Dried Tomatoes

- vii Dacon
- viii Spinach
- ix Roasted Garlic
- x Jalapeno
- (Meat type toppings cost \$2.15 Each)
- xi Ground Beef
- xii Shredded Chicken
- xiii Grilled Chicken
- xiv Pepperoni
- xv Ham
- xvi Bacon

- Text Field for customer name.
- Text Field for customer Phone number.
- Text Field for pizza quantities.
- You are allowed to discuss the design of the window during the lab times and discuss better options. (Design can be changed on the basis of ease of use of the application from clients perspective).
- You are required to choose the data structure of your choice to store the information of a client and order details. (ArrayList, Map, LinkedList etc...)
- Discuss during the lab days about the data structure you want to choose.
- You are required to follow the java naming conventions for classes and member variables. Your solution should be designed to follow the OO-design concepts using encapsulation, abstraction etc.

Event handlers:

- **Clear** button should clear all the fields and should not save the data.
- **Place Order** button should properly create the objects of class(es) and store them in the data structure(s).
- **Place Order** button should also display the summary of the order after placing the order.
- **Order confirmation** button (keep the work simple this our first GUI app)
- **Order summary** can be a text area. (For the simplicity of the work consider that the summary will be shown once Order Confirmation button is clicked.
- **Order Summary** should display the information as follows

Customer Name:

Customer Phone:

Pizza Type:

Pizza Size:

Quantity:

Total before tax:

Total to be paid: (should include tax if applicable)