INTPROG
Introduction to Programming
Java Coursework: "Pizza Ordering System"

Introduction

This coursework assignment is designed to allow you to demonstrate your understanding of Java. The assignment will be marked out of 40 and is worth 20% of the unit marks. You need to submit your program via the unit's Moodle site by the deadline of **11.00pm, Friday 3rd May 2019**, and are required to demonstrate your submitted program in your practical class during the **week beginning 6th May 2019**. Study this handout thoroughly in order to understand exactly what is expected for the coursework.

Scenario

A company, Nel Mezzo, has a small take-away restaurant in Portsmouth and currently takes pizza orders by phone and in person. When a customer places an order by phone, Nel Mezzo takes their order by writing it down on paper. A single order can be made up of a number of pizzas. Typically Nel Mezzo staff have to note down the combinations and costs of ingredients to calculate the cost of a pizza and then the cost of the whole order. See **cost sheet of ingredients** at end of document. A customer can order as many pizzas as they wish (including more than one pizza of the same type). Each pizza is made up of a size, crust type, sauce and two types of topping (selected from same list). Both toppings are optional. The customer can only select two toppings. The first topping comes in tablespoonfuls of 5. The second topping comes in tablespoonfuls of 4. The base cost is size cost + crust cost.

Details of the **costs** of **ingredients** and **sample data** are at the end of the document**.**

Tasks

You have to design and implement a desktop Pizza Ordering System using the Java programming language. The submitted version of your application must run in NetBeans 8.1. if you implemented your program in another IDE you will need to convert it to this platform and thoroughly test it before submission. See the submission notes on page 5 for further details.

The pizza ordering system should allow a user to begin a new order by resetting the application to its default state. They create a new pizza by selecting a base (size and crust), sauce and two (optional) toppings. Once the base and ingredients are selected the pizza can be added to the Order and more individual pizzas can be added. During the process the application should display up-to-date costs of the pizzas being added and the cost of the whole order. All testing must be written in a separate Test class with a main() method.

Task Breakdown

You should complete all core tasks before attempting the challenge task. To complete Tasks 1 & 2 a fully functioning test script should be built for your `Pizza` and `Order` classes. See **sample order data** at end of document for test script layout. You will lose marks if you attempt the challenge without completing Tasks 1 & 2 and their testing.

Tasks 1 & 2: Core	
Task	**Marks**
1	12
2	20
Task 3: Challenge	
Task	**Marks**
3	8

You can create additional classes, use **Enums**. add any further methods to support the behaviour of classes in your core tasks if you feel they are necessary but you will be asked to explain the design rationale for those additional objects and methods.

Task 1 Build both a Pizza class and a corresponding test script for the Pizza class. The Pizza class is responsible for representing an individual pizza and its values in the ordering process. A Pizza class should include variables for the following aspects of a pizza:

size, crust, sauce, topping1 and topping2

It should have get/set methods and constructor(s). Additionally, it should have the following functionality to allow it to return information to the user:

1. A method to return a formatted String of the pizza information. See **sample pizza** at end of document for output layout.
2. A method to calculate and return the cost of the pizza. See **cost sheet for ingredients** at end of document for cost calculations.

The test script should be laid out clearly and demonstrate all the functionality of the class.

Task 2 Build both an Order class and a corresponding test script for the Order class. The Order class is responsible for managing an array list of Pizza objects. It should have get/set methods and constructor(s). Additionally, it should have the following functionality:

1. A method to return a formatted String for the whole order. See **sample order data** at end of document for output layout.
2. A method to calculate and return the cost of the whole order. See **cost sheet for ingredients** at end of document for cost calculations.
3. A method to return the number of pizzas in the order.
4. A method to add pizzas to the order.
5. A method to select and delete a specific pizza from the order.
6. A method to select and update a specific pizza from the order.

The test script should be laid out clearly and demonstrate all the functionality of the class and, where necessary, relevant functionality of the Pizza class.

Challenge:

Task 3 Build a graphical user interface (GUI) for classes and enums you have built and tested in Tasks 1 & 2. It is recommended you use a drag-and-drop technology to build the user interface components (i.e. Java Swing or an equivalent). If you use any other technology check with your tutors that it is compatible with NetBeans 8.1. All data stored and updated is held in objects made from the `Order` and `Pizza` classes.

The design, layout and number of forms is for guidance and illustration purposes only.

Example Home/Order Form

(sample)

The example Home/Order Form should have buttons (or equivalent mechanisms) to add or edit pizzas. It should also display the current Order on the form as a listing (see image).

It should display a separate Total Order Cost.

It should include a Close or Exit button.

This form should use an instance of the Order Class to store and manage data.

Example New Pizza Form

(sample)

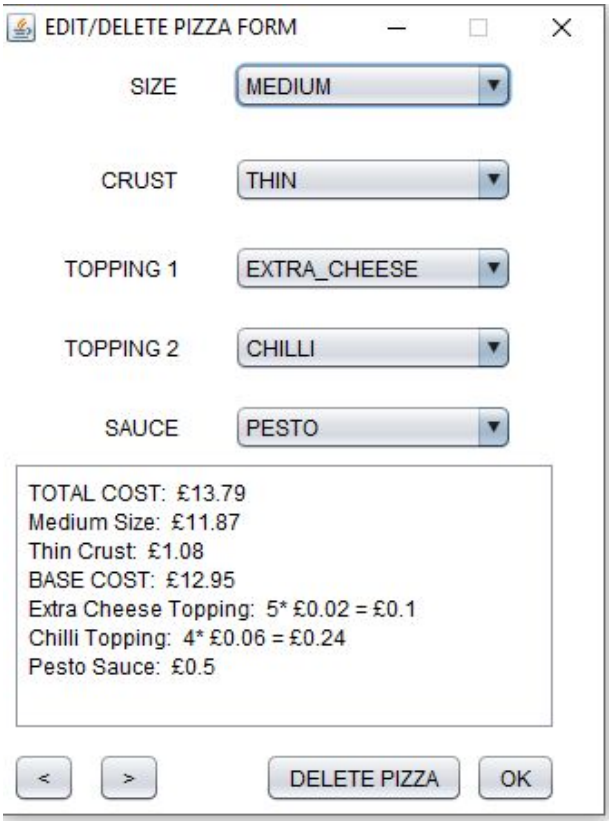
The example New Pizza Form should allow the user to select the ingredients and base for a Pizza. Combo Boxes or equivalent are recommended for the ingredients and these must be based on the **cost sheet for ingredients**. These are directly linked to the member variables of the Pizza Class.

This form should also display the current total of the Pizza as it is being built.

It should include an Add to Order button (or equivalent mechanism) which will add the Pizza to the Order and return focus to the Home/Order Form.

This form should create an instance of the Pizza Class to store and manage data.

Example Edit Form



(sample)

The example Edit Form should have simple toggle buttons or equivalent mechanism) to allow you to display a particular Pizza:

This example form should allow you to change any of the elements of the base or any ingredient and it should dynamically recalculate the cost

The OK Button (or equivalent mechanism) should update the Pizza in the Orders and return to the Home/Order Form.

This form should also allow you to delete the Pizza being displayed and return you to the Home/Order Form.

This form should use both the Order class (to toggle through the Pizzas in the Order) and the Pizza Class to store and manage data.

Moodle Submission & Demonstration

Project/Package name: **JavaCourseworkUPXXXXXX**

You must upload your Java Project as a NetBeans 8.1 project in a zipped form to Moodle by **Friday 11 pm 3rd May 2019 GMT**:

File Name: **INTPROG_UPXXXXXX.zip**

If you have used another development environment it is your responsibility to ensure that it executes in NetBeans 8.1. **Please Note:** The University system only runs NetBeans 8.1. If you build your project in any other IDE then you must test it on Netbeans 8.1 on a university machine so it can run during demonstration week.

If the project does not run on NetBeans 8.1 during demonstration week it will receive 0 for functionality and will only be judged on code quality.

You will have a demonstration in the **week beginning 6th May 2019** in which you will be required to demonstrate your test script and class code for task 1 and 2 and, if attempted, user interface.

Cost Sheet for Ingredients & Sample Data

Cost Sheet for Ingredients

Your software should assume these will be fixed and not subject to change.

base & ingredients	values (Type, Cost)	quantity (per Pizza)	Comments (for Nel Mezzo staff)
size (part of base)	SMALL, Cost £9.45 MEDIUM, Cost £11.87 LARGE, Cost £15.90	1	There are only 3 options of pizza size to select from: SMALL, MEDIUM and LARGE.
crust (part of base)	THIN, Cost £1.08 DEEP, Cost £1.10 STUFFED, Cost £2.14	1	There are only 3 options of pizza crust to select from: THIN, DEEP and STUFFED
base	size + crust	1	the base is calculated by adding the cost of the size and the cost of the crust.
sauce	TOMATO, Cost £0.00 PESTO, Cost £0.50	1	There are only 2 options of pizza sauce to select from:TOMATO, which is free or PESTO A customer can only select 1 sauce per pizza.
topping1	NONE, Cost £0.00 OLIVES, Cost £0.08 EXTRA_CHEESE, Cost £0.02 CHILLI, Cost £0.06 PEPPERONI, Cost £0.10 MUSHROOM, Cost £0.03 ROCKET, Cost £0.05 JALAPENOS, Cost £0.20 ONION, Cost £0.07 ANCHOVY, Cost £0.09	5 (topping1 is only given in spoonful of 5)	There are only 10 options of topping to select from: NONE, OLIVES, EXTRA_CHEESE,CHILLI, PEPPERONI, MUSHROOM, ROCKET, JALAPENOS, ONION and ANCHOVY topping1 is only given in tablespoonfuls of 5. eg. if ONION is selected as topping1 then they get 5 tablespoonfuls of onion toppings, costing 5*£0.07=£0.35
topping2	NONE, Cost £0.00 OLIVES, Cost £0.08 EXTRA_CHEESE, Cost £0.02 CHILLI, Cost £0.06 PEPPERONI, Cost £0.10 MUSHROOM, Cost £0.03 ROCKET, Cost £0.05 JALAPENOS, Cost £0.20 ONION, Cost £0.07 ANCHOVY, Cost £0.09	4 (topping2 is only given in tablespoonfuls of 4)	There are only 10 options of topping to select from: NONE, OLIVES, EXTRA_CHEESE,CHILLI, PEPPERONI, MUSHROOM, ROCKET, JALAPENOS, ONION and ANCHOVY topping2 is only given in tablespoonfuls of 4. eg. if OLIVES are selected as topping2 then they get 4 tablespoonfuls of olive toppings, costing 4*£0.08=£0.32

Sample Order Data

sample pizzas	details
TOTAL COST: £13.79 Medium Size: £11.87 Thin Crust: £1.08 BASE COST: £12.95 Extra Cheese Topping: 5* £0.02 = £0.10 Chilli Topping: 4* £0.06 = £0.24 Pesto Sauce: £0.50	This pizza is medium sized with a thin crust. The first topping is extra cheese. All first toppings come in tablespoonfuls of 5.The second topping is chilli. All second toppings come in tablespoonfuls of 4. The sauce selected is pesto. **Base = size + crust** **TOTAL COST = Base + Extra Cheese + Chilli + Pesto Sauce**
TOTAL COST: £15.19 Medium Size: £11.87 Stuffed Crust: £2.14 BASE COST: £14.01 Olive Topping: 5* £0.08 = £0.40 Onion Topping: 4* £0.07 = £0.28 Pesto Sauce: £0.50	This pizza is medium sized with a stuffed crust. The first topping is olive. All first toppings come in tablespoonfuls of 5.The second topping is onion. All second toppings come in tablespoonfuls of 4. The sauce selected is pesto. **Base = size + crust** **TOTAL COST = Base + Olive + Onion + Pesto Sauce**
TOTAL COST: £11.73 Small Size: £9.45 Thin Crust: £1.08 BASE COST: £10.53 Chilli Topping: 5* £0.06 = £0.30 Pepperoni Topping: 4* £0.10 = £0.40 Pesto Sauce: £0.50	This pizza is small sized with a thin crust. The first topping is chilli. It comes in tablespoonfuls of 5. The second topping is pepperoni and comes in tablespoonfuls of 4. The sauce selected is pesto. **Base = size + crust** **TOTAL COST = Base + Extra Cheese + Pepperoni + Pesto Sauce**
TOTAL COST: £11.75 Small Size: £9.45 Thin Crust: £1.08 BASE COST: £10.53 Olive Topping: 5* £0.08 = £0.40 Olive Topping: 4* £0.08 = £0.32 Pesto Sauce: £0.50	This pizza is small sized with a thin crust. The first topping is olive. It comes in tablespoonfuls of 5. The second topping is olive and comes in tablespoonfuls of 4. The sauce selected is pesto. **Base = size + crust** **TOTAL COST = Base + Olive + Olive + Pesto Sauce**