Project #4 (170 points)

Due Date

Thursday, April 4, by 11:59pm.

Submission

(1) Zip your project folder and submit it to Canvas. The zipped file MUST include the following grading items.
- Source folder **src**, containing the folders below. [140 points]
  - ✓ Java package folder, including all the Java **source files** (the "Model" and the "Controller")
  - ✓ Resource folder, containing all the **fxml files** (the "View") and image files.
- Class Diagram. [10 points]
- A **JUnit test class** for Sandwich class. [15 points]
- Javadoc folder. [5 points]

(2) The submission button on Canvas will disappear after **April 4, 11:59pm**. It is your responsibility to ensure your Internet connection is good for the submission. **You get 0 points** if you do not have a submission on Canvas. **Projects sent through the emails will not be accepted**.

Project Description

RU Café provides a few menu items, including sandwiches, donuts, and coffee, which are available between 9am and 4pm daily. Your team will develop a software with JavaFX to help the cafe manages the orders. The staff will be the users who use the software to take the orders, place the orders and cancel the orders.

There are 3 types of donuts available in the café: yeast donuts, cake donuts and donut holes. Each type of donuts includes a variety of flavors. Regardless of the flavors, a yeast donut is $1.79, a cake donut is $1.89, and a donut hole is $0.39.

The café provides brewed coffee only. Customers have the choices of add-ins, including sweet cream, french vanilla, Irish cream, caramel, and mocha. Each add-in costs $0.30. There are different cup sizes for the coffee: Short, Tall, Grande and Venti. The base price for a Short black coffee is $1.99. The price increases $0.50 for the next cup size. For example, a Tall black coffee would be $2.49, and a Grande black coffee would be $2.99. A Short coffee with 2 add-ins would be $2.59.

The sandwich options are beef, chicken, and fish. Customers can choose from 3 different breads for the sandwich, including bagel, wheat bread, and sour dough. A beef sandwich is $10.99, a chicken sandwich is $8.99, and a fish sandwich is $9.99. Customers can choose the add-ons, including cheese, lettuce, tomatoes, and onions. However, each veggie add-on costs $0.30 extra, and the cheese costs $1 extra.

Project Requirement

1. You MUST follow the Coding Standard posted on Canvas under Week #1 in the "Modules". **You will lose points** if you are not following the rules.
2. You are required to follow the Academic Integrity Policy. See the **Additional Note #14** in the syllabus. If your team uses a repository hosted on a public website, you MUST set the repository to private. Setting it to public is considered as violating the academic integrity policy. The consequences of violation of Academic Integrity Policy are: **(i) your group receives 0 (zero) on the project, (ii) the violation is reported, (iii) a record on your file of this violation.**
3. You are **NOT ALLOWED to use System.out** (write to console) or **System.in** (read from console) ANYWHERE in ALL CLASSES, **or you will lose 3 points for each violation, with a maximum of losing 15 points**.

4. **Navigation design**. Your software must provide at least 3 different “Views” for ordering sandwiches, coffee, and donuts **-5 points** each violation. The user shall be able to view/add/remove multiple menu items of a current order. **-10 points** if this is not done properly. The users shall be able to view the details of all the orders that have been placed and cancel an order, **-10 points** if this is not done properly.
5. The functional requirements are listed below. **-2 points** for each requirement not met.
   (1) **Ordering Donuts**. User can add or remove the selected donuts to/from the current order. The GUI shall include the following user interface design.
   - The options of 3 donut types: yeast donuts, cake donuts and donut holes.
   - For each donut type selected by the user, an image associated with the donut type must be displayed.
   - For each donut type selected, display the associated donut flavors for selection, with a minimum of 6 flavors for yeast donuts, 3 flavors for cake donuts and 3 flavors for donut holes.
   - For each donut flavor selected, user can enter or select the quantity.
   - While adding or removing the donuts, the GUI shall display the running sub-total of the donuts selected so far, with 2 decimal places.
   - User can add the selected donuts to the current order.

   (2) **Ordering Coffee**. The GUI shall include the following user interface design.
   - Cup size selection of Short, Tall, Grande and Venti.
   - Multiple selection of the add-ins: sweet cream, French vanilla, Irish cream, caramel, and mocha.
   - Quantity selection of the number of cups, up to 5 cups of the chosen size and chosen add-ins.
   - While adding or removing the add-ins or changing the cup sizes, the GUI shall display the running sub-total of the coffee, with 2 decimal places.
   - User can add the coffee to the current order.

   (3) **Ordering Sandwiches**. The GUI shall include the following user interface design.
   - Bread selection of bagel, wheat bread and sour dough.
   - Protein selection of beef, fish and chicken.
   - Add-ons selection of lettuce, tomatoes, onions, and cheese.
   - While adding or removing the add-ons, or choosing the protein, the GUI shall display the running sub-total of the sandwich, with 2 decimal places.
   - User can add the customized sandwich to the current order.

   (4) **Current order**. This is similar to a shopping cart where the system temporarily holds the menu items added so far, but the order hasn’t been placed yet. The GUI shall include the following user interface design.
   - User can review the current items in the order, remove a selected menu item or place the order. Each menu item in the order shall include the details of the item name, quantity, add-ins and the cup size (for coffee only), or the add-ons (for sandwich only.)
   - Display the subtotal, sales tax, and total amount (subtotal + tax) of the current order. The sales tax in New Jersey is 6.625%. All dollar amounts must be displayed with 2 decimal places.
   - User can place the order.

   (5) **All orders**. User can view all the orders that have been placed. The GUI shall include the following design.
   - User can select an order and view the details and the total amount (subtotal + tax, with 2 decimal places) of each order.
   - User can select an order and cancel the order.
   - User can save/export the orders to a text file, which shall include the details of every order, consistent with the details displayed on the GUI. You will **lose 10 points** if this is not implemented.
6. You CAN use any Java library classes. In addition, you must include the following classes. **-5 points** for each class missing OR not used OR not properly defined. All instance variables must be private, or **-2 points** for each violation. You can define additional classes if necessary.

- **MenuItem class**. This is an **abstract class** and the superclass of all menu items. Any class defined for a menu item must extend this class, or **-5 points** for each violation. You must include the abstract method below.

```
public abstract double price(); //subclasses must implement this method
```

- Create Coffee class, Donut class and Sandwich class to extend MenuItem class. You should define the necessary instance variables for each class based on good object-oriented practices.
- **Order class**. Include 2 private instance variables: (1) the order number and (2) the list of menu items. The order number is a unique number for each order. The list of menu items must use the abstract type **MenuItem**, which can hold donut, coffee, or sandwich objects. **-10 points** if this is not done properly.

7. You MUST set the titles of all the stages (titles for the windows), or **-2 points each, with a max of 4 points off.**
8. You MUST use at least one ComboBox, one ListView and one Image Button, or **-5 points each violation.**
9. All methods should not **exceed 40 lines, or -2 points** for each violation**, maximum 4 points off.**
10. You are required to **generate the Javadoc** after you properly commented your code. Your Javadoc must include the documentations for the constructors, private methods, and public methods of all Java classes (*.java files.)
    - You DO NOT need to comment the .java file that contains the main() method. You MUST **comment all other .java files.**
    - When you are generating the Javadoc, DO NOT include the *.fxml files, which are NOT java classes.
    - Generate the Javadoc in a single folder and include it in your project folder to be submitted to Canvas.
    - You are responsible to **double check** your Javadoc after you generated them. The grader will navigate the Javadoc with the "index.html". You will **lose 5 points** for not including the Javadoc, OR, the grader cannot navigate your Javadoc through the "index.html".
11. Create a **JUnit test** class for the Sandwich class, and test the **price() method**. Must include at least 5 test cases with different options of the protein and add-ons that will return difference prices. You can use **assertEqual()** to compare the expected output and actual output.
12. Create a **Class diagram** to document your design. You must include all the Java classes created for this project. DO NOT include the Java/JavaFX library classes, JUnit test class and .fxml files.
13. **System Testing**. You are responsible to thoroughly test your software and ensure your software is meeting the project requirements. You will **lose 2 points** for each incorrect implementation, incorrect amount or incorrect information shown on the GUIs. Your software must always run in a sane state and **should not crash in any circumstances**. You must catch all Java Exceptions. Your program shall continue to run until the user stops the program execution or closes the window. **You will lose 2 points** for each exception not caught.