CS 235 - Week 11 Lab Exercise - 2021-11-05

Deadline

Due by the end of lab on 2021-11-05.

How to submit

Submit your `.java` file for your resulting Java application for this lab on https://canvas.humboldt.edu

Purpose

To practice a bit with `JCheckBox` and `JRadioButton` components.

Important notes

- IF you are attending the lab via Zoom, you are expected to pair program in a breakout room (possibly trio-program if necessary based on class members' Internet and the number of class members attending via Zoom).
  - In this case, **be sure to TYPE BOTH (all) OF YOUR NAMES** in the beginning comment of EACH of your `.java` files

  But, because of the delta variant surge, if you are attending lab in person in BSS 317, you will each work on a separate computer, although discussion amongst those attending will be encouraged!

Your Task

Create a GUI Java application `PizzaOrder.java`, following the course style standards, that meets the following requirements:

- The frame should include a panel whose top/north contains a top subpanel, whose center contains a subpanel containing a left and right subpanel, and whose bottom/south contains a bottom subpanel.
- The top subpanel should contain one or more `JLabel` objects including the name of your pizza organization and your names.
- The center subpanel's left subpanel should contain a logically-grouped set of **at least 4** `JRadioButton` choices of different pizza sizes.
  - Each radio button should be associated with a listener that sets a data field `basePizzaCost` based on the pizza size radio button selected.
  - Your smallest size should be the default selected when your application starts.
  - In the bottom/south of this center left subpanel, a `JTextField` should display the pizza size chosen, updated immediately and automatically upon the selection or any change in a pizza size.
- The center subpanel's right subpanel should contain **at least 6** `JCheckBox` choices of different pizza toppings.
  - Each checkbox should be associated with a listener that keeps track of the extra cost of the toppings

selected ($1.50 for each chosen topping, unless you decide you would rather have different prices for each topping, in which case you may choose those) and saves the total toppings cost for the checkboxes selected in a data field `toppingsCost`.

  - In the south/bottom of this center right subpanel, a `JTextField` should display the names of all toppings currently selected, updated immediately and automatically upon the selection or any change in pizza toppings.

- The bottom subpanel should contain a `JTextField` that displays the current cost of the pizza based on the size and the choice of toppings (this will simply be the value of `basePizzaCost + toppingsCost`). This field should updated immediately and automatically upon the selection or any change in a pizza size and toppings.

- Along with this lab exercise, there is an example application `JScrollBarDemo.java`, that demonstrates a simple `JScrollBar` object that is sensitive to `AdjustmentEvent` events because an `AdjustmentListener` has been added to it.

  - Using this as an example of basic `JScrollBar` setup, decide on a "pizza criteria" where a range of integers would be a reasonable choice (a heat level for the pizza sauce ranging from 1 to 6, for example, or a toastiness or thickness of crust ranging from 10 to 20, etc.).

  - Add a suitable `JScrollBar` for entering this in the north, south, east, or west of one of your panels or subpanels -- or in an additional panel added to your center panel, if you prefer.

  - Either add a `JTextField` displaying its value in one of your panels, or decide which existing `JTextField` will also be reporting this scrollbar's value.

  - This `JScrollBar` should be sensitive to adjustment events such that, when the user adjusts this scrollbar, the resulting value is displayed in its corresponding `JTextField` (or in the `JTextField` you chose).

- (Colors, font sizes, borders if any, etc. are up to you. And you may add additional `JLabel` objects to these subpanels if you'd like.)

- When you are done, or before you leave lab, use Gmail to

  - MAIL a copy of your `.java` file to BOTH/ALL of you, and

  - EACH of you should SUBMIT the required files on Canvas