Java Programming II 2018: Exercise 02

Submission Guide:
- Due Date: Before the next exercise class (April 20, 2018).
- Create a directory ~/java2/ex02/ and give the "chmod 705 ~/java2/ex02/" command, and save your answers to the directory.

In this exercise, you will exercise Java Swing programming.

Preparation

"cd" to your work directory and copy the following files to your work directory.

```
/home/course/java2/code/ByTopics/ExGUISwing/*
```

[Step 1] A Simple Swing Frame (50)

The details of JFrame

View the answer of this problem by "java ExGUISwing_01_ans".

Create a simple window with title "ExGUISwing-01" but showing nothing on its contents area .

Requirements:

- class name: ExGUISwing_01 (file name: ExGUISwing_01.java)

- super class: JFrame

Comments:

- Import javax.swing.JFrame class at the beginning of your file.

- In the constructor:
  (1) Call the constructor of JFrame with a title string.
  (2) Set the initial size of the window (by setSize() of JFrame) .
  (3) Call the setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) for termination of the program when the window is closed.

- Implement the main() method:
  (1) It must be the following format
  "**public static void main(String argv[ ])**".
  (2) Create a ExGUISwing_01 object.
  (3) Call ExGUISwing_01's setVisible() with parameter "true".

[Step 2] A Car Audio Control Panel Simulator (50)

The details of Container, JPanel, JLabel, JButton, BevelBorder and Font

View the answer of this problem by "java ExGUISwing_02_ans".

This is an extension of ExGUISwing_01, a window showing a central display and 6 buttons. The display has a border.

Requirements:

- class name: ExGUISwing_02 (file name: ExGUISwing_02.java)

- super class: ExGUISwing_01 (If you have not finished Step 1, please use ExGUISwing_01_ans instead of ExGUISwing_01, as the super class)

Comments:

- Use a JButton object for each button

- Use a JPanel object and a BevelBorder object for the central display

- For showing text on the central display, use JLabel object

- Create above JButton, JLabel and JPanel objects as attributes for future use.

- In the constructor:
  (1) Call the constructor of its super class.
  (2) Get the Container object of the JFrame (by getContentPane() of JFrame)
  (3) Remove the layout manager of the Container and JPanel objects by calling their setLayout(null).
  (This is to control the location and the size of the buttons and text by ourselves)
  (4) For each created object:
  Set the size (use setSize() method).
  Set the location (use setLocation() method).
  (5) For the JPanel object:
  Create a BevelBorder object and set it to the JPanel by its setBorder() method.
  (6) For the JLabel object:
  Create a Font object and set it to the JLabel (use setFont()).
  (7) Add the JButton and JPanel objects to the Container object (use add() method).
  (8) Change the JLabel object's foreground color to dark green.
  (9) Add the JLabel object to the JPanel object.

- Implement the main() method:
  (1) Create a ExGUISwing_02 object.
  (2) Call ExGUISwing_02's setVisible() with parameter "true".

[Step 3] Give Live to the Simulator (Bonus: 20)

The details of ActionListener

View the answer of this problem by "java ExGUISwing_03_ans".

This is an extension of ExGUISwing_02. Clicking the buttons on the window can change the text on the central display and on-off other buttons.

Requirements:

- class name: ExGUISwing_03 (file name: ExGUISwing_03.java)

- super class: ExGUISwing_02 (If you have not finished Problem 2, please use ExGUISwing_02_ans instead of ExGUISwing_02, as the super class)

- implement interface: ActionListener to catch button events.

Comments:

- Use another JLabel object as an attribute to show "AM" or "FM" on the central display.

- Use attributes to save necessary information.

- Implement the constructor:
  (1) locate the new JLabel object on the central display (JPanel object);
  (2) add listeners (this) to all the buttons.
  * If you use ExGUISwing_02_ans as the super class, use the information below for accessing its attributes:

```
protected JButton btnToCD = new JButton("CD");
protected JButton btnToAM = new JButton("AM");
protected JButton btnToFM = new JButton("FM");
protected JButton btnPW = new JButton("PW");
protected JButton btnUp = new JButton("Up");
protected JButton btnDown = new JButton("Down");
protected JPanel pnlDisplay = new JPanel();
protected JLabel lblInfo = new JLabel("Power off");
```

- Implement actionPerformed() for ActionListener:

- To on-off a button use setVisible() method.

- To change the text on a JLabel, use setText() method.
