

Criterion C: Development

Java Swing

The product I am making utilizes Java Swing, a set of GUI (graphical user interface) tools to create windows for the program to be run in. Java Swing is “considered one of the best Java GUI frameworks for building various desktop applications”(Tagline) due to the “many GUI components and platform-independent user interfaces”(Tagline) that the program is able to provide.

The GUI provides an interface for the user to interact with, which is crucial for the timer to function as a desktop application. Using Swing for this will make the process more efficient and streamlined and will simplify the coding process immensely.

```
1 import javax.swing.*;  
2 import java.awt.*;  
3 import java.awt.event.ActionEvent;  
4 import java.awt.event.ActionListener;
```

ActionListener is a tool that checks for the occurrence of ‘events’ such as JButton presses. This will be used in conjunction with JButton to execute chains of code when certain buttons are pressed, such as the planned StartButton and PauseButton.

ActionEvent works with ActionListener to execute the code when activated by ActionListener. When using ActionEvent, I implement code along with the method to be executed. For example, when ActionListener detects a button press of the JButton PauseButton, it calls the method I instantiated called pauseSession. As expected, after being prompted by ActionListener, ActionEvent calls the function and gives the JButton functionality.

```

public void pauseSession() { //simple pause method using booleans
    timer.stop();
    startButton.setEnabled(true);
    pauseButton.setEnabled(false);
}

```

Inheritance

```

public class StudyTimer extends JFrame {

    //instance variables moved
    private int studyTime;
    private int breakTime;
    private int totalStudyTime;
    private int totalBreakTime;
    private int remainingTime;
    private boolean isStudying;
    private boolean isBreak;
    private Timer timer;
    private JLabel timerLabel;
    private JButton startButton;
    private JButton pauseButton;
    private JLabel studyTimeLabel;
    private JLabel breakTimeLabel;
    private JLabel totalStudyLabel;
    private JLabel totalBreakLabel;
    private JPanel controlPanel;
}

```

A JFrame is a Java.Swing tool that handles the variables of the window display such as its size and layout. The StudyTimer class extends JFrame in order to create a window for the application. In doing so, it inherits the properties and conventions of a JFrame, which streamlines the process of setting up the GUI via Java Swing. After making StudyTimer inherit the properties of JFrame, we create instance variables for the necessary components of the timer program and the required studyTime and breakTime variables. Buttons and panels are also created here for later implementation in the program.

```

setTitle("Study Timer");
setSize(500, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(5, 2));

```

The variables of the Frame were set, being the title, size, default close operation (what the program does upon closing)

JPanel

```
controlPanel = new JPanel();

studyTimeLabel = new JLabel("    Study Time: " + studyTime + " minute(s)");
add(studyTimeLabel);

breakTimeLabel = new JLabel("    Break Time: " + breakTime + " minute(s)");
add(breakTimeLabel);

totalStudyLabel = new JLabel("    Total Study Time: " + totalStudyTime + " seconds");
add(totalStudyLabel);

totalBreakLabel = new JLabel("    Total Break Time: " + totalBreakTime + " seconds");
add(totalBreakLabel);
```

JLabel is a Swing tool that displays text/image in the GUI. In this case, it will be used because of its text display. Several JLabels are created for the stats that will be displayed on screen: Study time, Break time, Total studied time, and total break time. Each one of these are instantiated as instance variables and added to the StudyTimer constructor.

JButton

```
startButton = new JButton("Start");
startButton.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        startSession();
    }
});
```

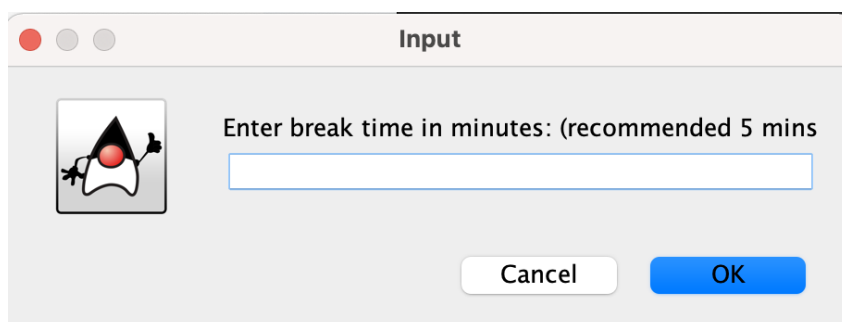
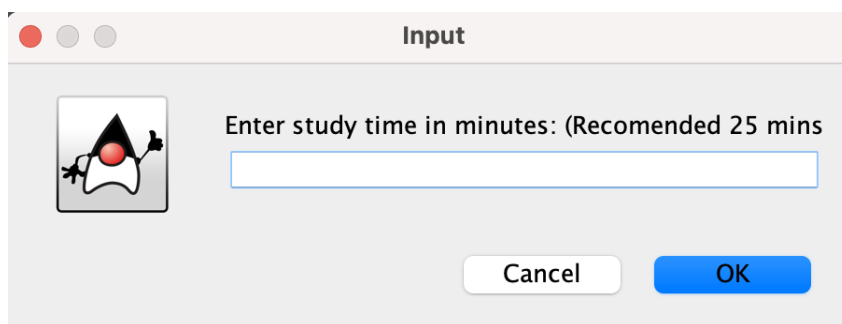
A new button is created using JButton with a built-in ActionListener to ensure that when the button press is made, that the code can be called for the button to start the timer. The same is done for the pause button as well.

JOptionPane

```
public static void main(String[] args) { //JOptionPane window for entering study times
    int studyTime = Integer.parseInt(JOptionPane.showInputDialog("Enter study time in minutes: (Recommended 25 mins)"));
    int breakTime = Integer.parseInt(JOptionPane.showInputDialog("Enter break time in minutes: (recommended 5 mins)"));

    StudyTimer timer = new StudyTimer(studyTime, breakTime);
    timer.setVisible(true);
}
```

JOptionPane is used to display a screen for the user to input the number of minutes for the work time and break times respectively. It takes an integer input for both and is used to create popup windows with options. The windows are made to take inputs from the user and run code accordingly. In this case, JOptionPane receives the integer values for StudyTime and BreakTime and sets the timer for both to take in those input values.





Study Timer

Study Time: 1 minute(s)

Break Time: 1 minute(s)

Total Study Time: 0 seconds

Total Break Time: 0 seconds

Remaining Time: 1:00

Start

Pause

Works Cited

Tagline. “Best Java Gui Frameworks.” *Tagline Infotech*, 17 Jan. 2024,

taglineinfotech.com/java-gui-frameworks/#:~:text=Nowadays%2C%20Swing%20is%20considered%20one,and%20platform%2Dindependent%20user%20interfaces.

“Java GUI: Full Course  (Free).” *YouTube*, YouTube, 14 Sept. 2020,

www.youtube.com/watch?v=Kmgo00avvEw&t=11104s&ab_channel=BroCode.

Lee, Alex. “Java GUI Tutorial - Make a GUI in 13 Minutes #99.” *YouTube*, YouTube, 6 Feb.

2020, www.youtube.com/watch?v=5o3fMLPY7qY&t=27s&ab_channel=AlexLee.

GfG. “Introduction to Java Swing.” *GeeksforGeeks*, GeeksforGeeks, 9 Jan. 2024,

www.geeksforgeeks.org/introduction-to-java-swing/.

“Java Swing Tutorial - Javatpoint.” *Www.Javatpoint.Com*, www.javatpoint.com/java-swing.

Accessed 11 Mar. 2024.

