JAVA Project

Building of

JNotepad



**By:**

Anudeep Ravulapally Sharma

12111223, B57

Avijeet Singh Bhati

12111692, A32

Ravindra Vishnoi

12111865, A03

**Submitted to:** Mr, Arvind Kumar, 16921

**INTRODUCTION**

Graphical User Interfaces (GUIs) are widely used in software development, providing users with an interactive interface to perform various tasks. The Swing library is one of the several libraries offered by Java for the creation and administration of graphical user interfaces. Developers may create applications with a polished appearance and feel by extending Abstract Window Toolkit (AWT) with the help of the Swing library of graphical user interface components. We shall examine the Java Swing code for a Word Counter programme in this report.

The Word Counter program is a simple GUI-based application that allows users to count the number of words in a given text and perform text search and replacement operations. The program has various features like finding all the occurrences of a given word, highlighting them and showing the number of times it occurs. The application provides a simple and easy-to-use interface with features that are useful for anyone who needs to manipulate text files.

In the following sections, we will provide a detailed analysis of the Java Swing code used in the Word Counter program. We will discuss the different components of the GUI, their functionalities, and how they are implemented in the program. Additionally, we will examine the code for the main method and other methods used in the program, as well as the different event listeners and the actions they perform when triggered. This analysis will provide a better understanding of how Java Swing can be used to develop GUI applications.

**CODE ANALYSIS**

The code consists of a single class named "WordCounter" that extends the JFrame class and implements the ActionListener interface. The class contains a constructor that initializes the GUI components, such as JTextArea, JLabel, and JButtons, and adds them to a panel that is added to the JFrame. The ActionListener interface is implemented to handle events from the buttons.

The GUI components are:

* JTextArea: a text area where the user can enter and edit text.
* JLabel: a label that displays the number of words in the text.
* JButtons: several buttons for performing different actions such as counting words, finding text, replacing text, clearing the text area, opening a file, saving a file, and changing the font size.

The constructor sets the default close operation, preferred size, and initializes the components. The buttons and the label are added to a panel, and the panel and the text area are added to the frame. Action listeners are added to the buttons, and the frame is made visible. The main method creates an instance of the WordCounter class to execute the program.

The class also contains four methods:

* findAllIndexes: a method that finds all indexes of a substring in a string and returns a List of integers representing the indexes.
* actionPerformed: a method that handles events from the buttons. It implements different actions depending on which button is pressed. For example, it counts the number of words, finds and replaces text, clears the text area, opens a file, saves a file, and changes the font size.
* countWords: a method that counts the number of words in a string and returns the count.
* main: the entry point of the program that creates an instance of the WordCounter class.

Overall, the code provides a basic implementation of a text editor with some useful functionalities. However, there is room for improvement, such as adding error handling, improving the UI, and adding more features like undo/redo, syntax highlighting, auto-indentation and font changing.

**CONCLUSION**

Swing makes it easier to create intuitive and user-friendly applications in Java. Additionally, Swing is platform-independent, meaning that it can run on any operating system that supports Java, making it an ideal choice for developing cross-platform applications. Overall, the use of Swing in the code demonstrates the versatility and power of Java as a programming language for creating complex and interactive applications with ease. The main purpose of this project for us was to understand the implementation of swing to further our Java skills. By integrating the implementation of swing in our respective skillsets we believe we have become better coders.