Name – Nethmi Bandara

Index – 16195

Mini Project - Simple Notepad App

Introduction

The Simple Notepad App is a basic text editor developed in Java Swing. It allows users to create, edit, open, and save text files. The project demonstrates knowledge of Java GUI components, event handling, and file operations. The interface is simple and user-friendly, like Windows Notepad.

Objectives

The main objectives of this project are:

- To implement a functional GUI-based text editor using Java Swing.
- To understand file handling (reading and writing) in Java.
- To practice creating menus, menu items, and event listeners in Java applications.
- To provide a simple and easy-to-use application for text editing.

Tools and Technologies

- Programming Language: Java
- IDE: IntelliJ IDEA
- GUI Components: JFrame, JTextArea, JScrollPane, JMenuBar, JMenu, JMenuItem
- File Handling: JFileChooser, BufferedReader, BufferedWriter
- Version Control: GitHub

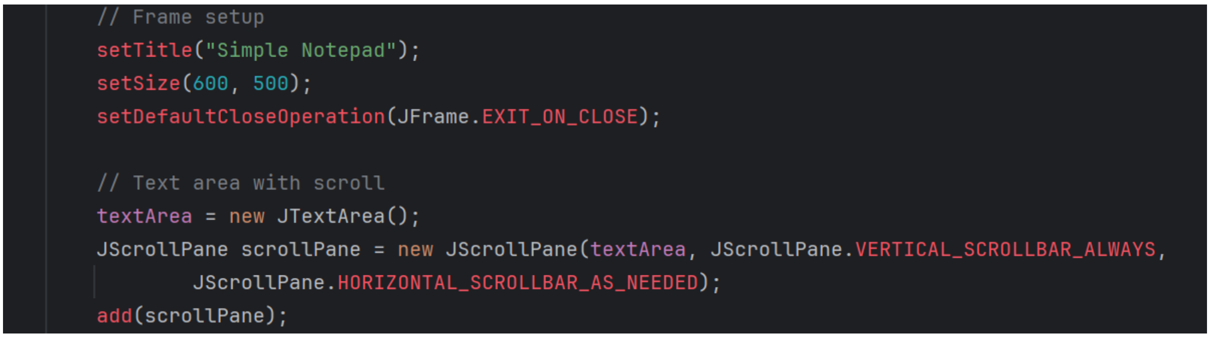
Features

Text editing area with scrollbars

- File menu with:
  - Open: Open existing .txt files
  - Save: Save content to a .txt file
  - Exit: Close the application
- Basic error handling for file operations

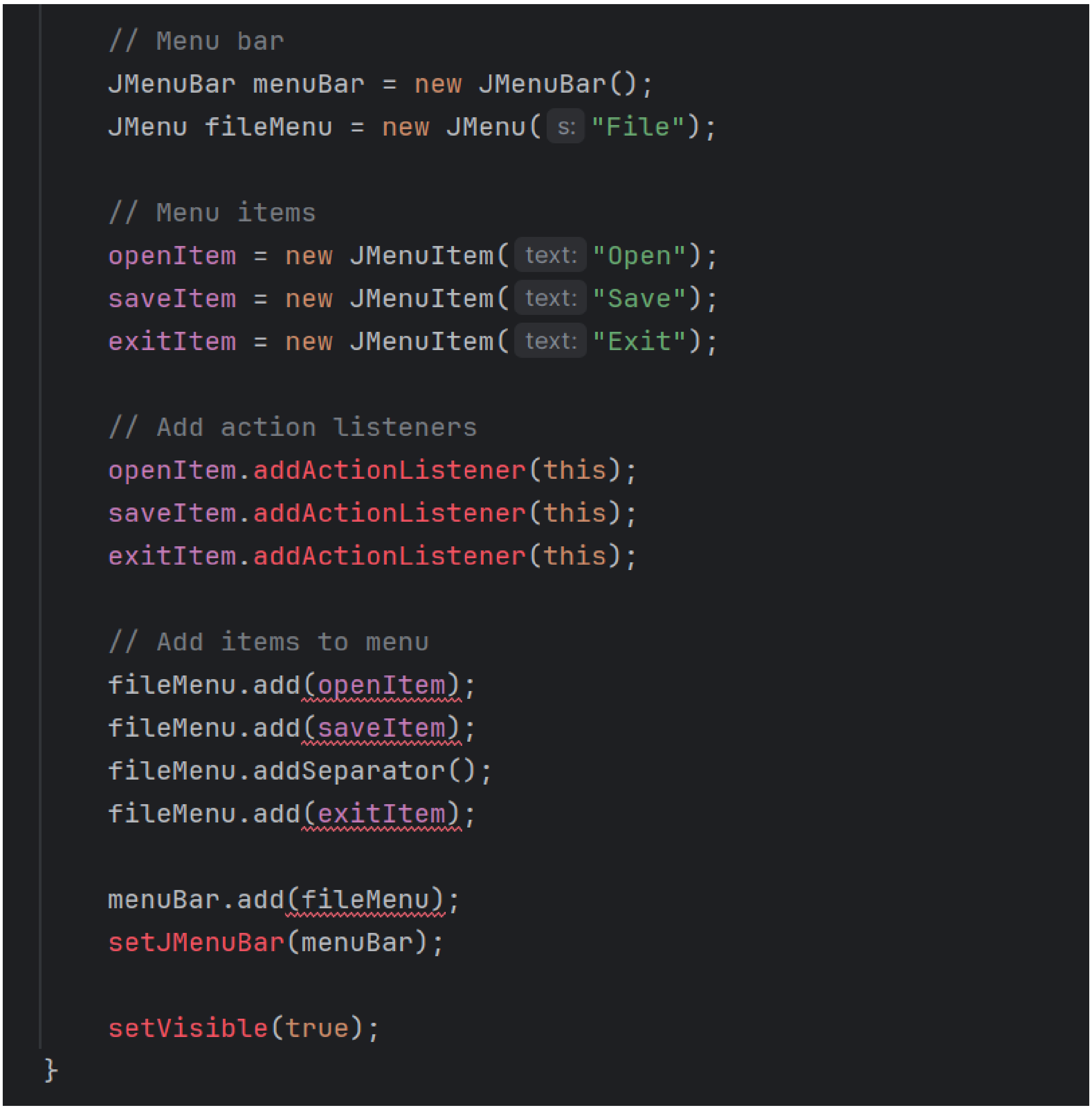
Code Snippets

JFrame and Text Area



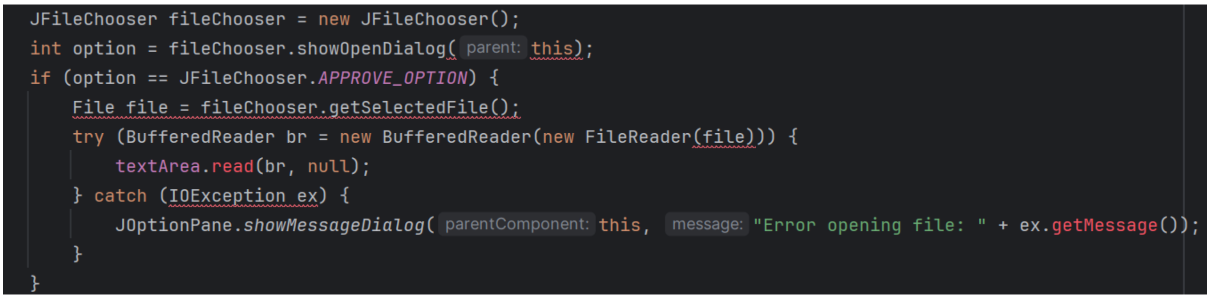
```
    // Frame setup
    setTitle("Simple Notepad");
    setSize(600, 500);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Text area with scroll
    textArea = new JTextArea();
    JScrollPane scrollPane = new JScrollPane(textArea, JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
            JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
    add(scrollPane);
```

Menu Setup



```
    // Menu bar
    JMenuBar menuBar = new JMenuBar();
    JMenu fileMenu = new JMenu( s: "File");

    // Menu items
    openItem = new JMenuItem( text: "Open");
    saveItem = new JMenuItem( text: "Save");
    exitItem = new JMenuItem( text: "Exit");

    // Add action listeners
    openItem.addActionListener(this);
    saveItem.addActionListener(this);
    exitItem.addActionListener(this);

    // Add items to menu
    fileMenu.add(openItem);
    fileMenu.add(saveItem);
    fileMenu.addSeparator();
    fileMenu.add(exitItem);

    menuBar.add(fileMenu);
    setJMenuBar(menuBar);

    setVisible(true);
}
```

File Open and Save



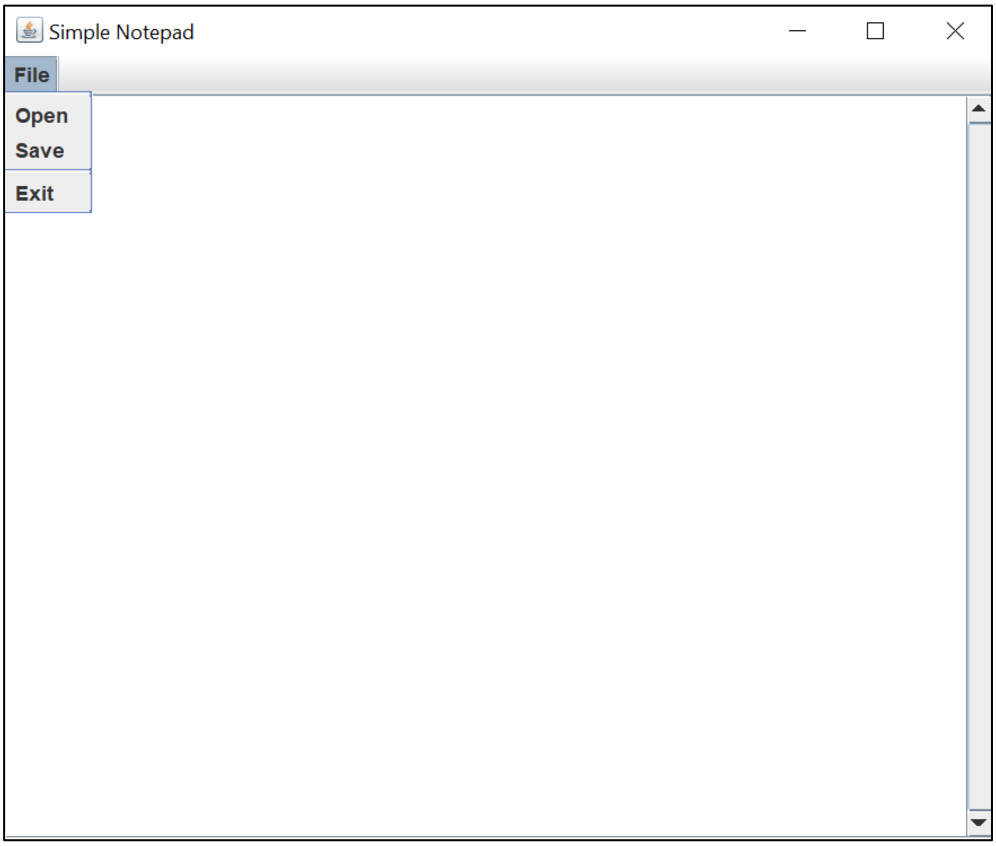
```
JFileChooser fileChooser = new JFileChooser();
int option = fileChooser.showOpenDialog( parent: this);
if (option == JFileChooser.APPROVE_OPTION) {
    File file = fileChooser.getSelectedFile();
    try (BufferedReader br = new BufferedReader(new FileReader(file))) {
        textArea.read(br, null);
    } catch (IOException ex) {
        JOptionPane.showMessageDialog( parentComponent: this,  message: "Error opening file: " + ex.getMessage());
    }
}
```

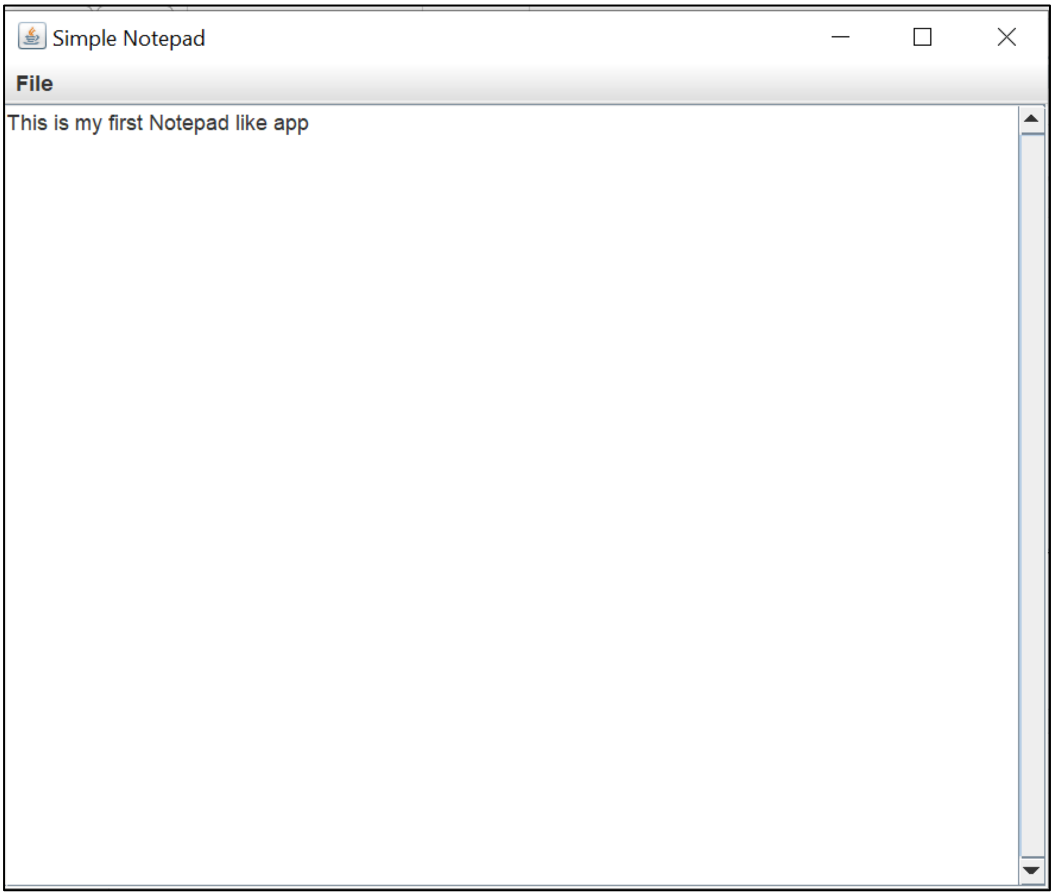
Conclusion

The Simple Notepad App provides a functional text editing solution using Java Swing. It demonstrates:

- Effective use of GUI components and layout management
- Implementation of menus and event handling
- Basic file input/output operations

This project serves as a foundation for building more advanced text editors in the future and helps to understand concepts of Java Swing and file handling.





This is my first Notepad like app