NORTH SOUTH UNIVERSITY

SPRING 2024

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

PROJECT REPORT

Course Code: CSE215
Course Title: Programming Language II
Course Instructor: Dr. Mohammad Shifat-E-Rabbi [MSRb]
Section: 08
Project Topic:

TEXT EDITOR WITH SYNTAX HIGHLIGHTING FOR DIFFERENT PROGRAMMING LANGUAGES

Submitted by Group Number: 09 [Nine]
Group Members:

Name	ID
Md.Tanjeelur Rahman Labib	2013677642
Md.Rejwan	2311238042
Jubayer Ahsan	2231404642

This application is built using the Swing framework and demonstrates the integration of advanced features such as syntax highlighting, dynamic font management, file handling, and a responsive graphical user interface (GUI).

1. Features Overview

1. **Text Editing**:
    - The editor uses a JTextPane to provide rich text formatting capabilities.
    - Syntax highlighting is implemented for programming constructs, string literals, and comments.
2. **Font Management**:
    - Users can change the font family and size dynamically using a JComboBox and a JSpinner.
    - Font color can be customized via a color picker (JColorChooser).
3. **File Operations**:
    - **Open File**: Supports reading and displaying .txt, .java, .py, and .html files.
    - **Save File**: Allows users to save their work to a specified file location.
    - File handling is executed on a separate thread to ensure UI responsiveness.
4. **Syntax Highlighting**:
    - Keywords, string literals, and comments are highlighted using regular expressions and StyledDocument.
    - The highlighting process is triggered dynamically on text changes (DocumentListener).
5. **User-Friendly Interface**:
    - Scrollable text area for enhanced usability.
    - Menu bar for accessing file operations.
    - Compact layout with intuitive controls.

---

2. Implementation Details

2.1 Class Overview

1. TextEditor Class:

- The main class responsible for constructing the GUI and managing interactions.
- Implements ActionListener to handle user actions on buttons, menus, and controls.

2. SyntaxHighlighter Class:

- Responsible for applying syntax highlighting using regular expressions.
- Implements DocumentListener to react to changes in the text document.

2.2 Design Highlights

1. **Event Handling:**
    - Event-driven programming is used for button clicks, menu selections, and document updates.
    - Threads ensure that file handling does not block the main UI thread.
2. **Modularization:**
    - Syntax highlighting logic is separated into a dedicated class, improving code maintainability.
    - File operations (loadFile and saveFile) are encapsulated within methods.
3. **Syntax Highlighting:**
    - Keywords: Highlighted in blue with bold styling.
    - Strings: Highlighted in orange for distinction.
    - Comments: Displayed in gray with italicized styling.
    - Regular expressions are used for pattern matching, ensuring efficiency and flexibility.

2.3 Libraries and Tools Used

- **Swing Components**: JTextPane, JScrollPane, JMenuBar, etc., for GUI development.
- **Regular Expressions**: Used for pattern matching in syntax highlighting.
- **Threads**: Ensures responsive file handling operations.

---

3. Strengths

1. **Responsiveness**: File operations are executed on separate threads, ensuring smooth UI performance.
2. **Flexibility**: Supports multiple file types and allows customization of font and color settings.

3. **Maintainability**: The separation of concerns (GUI, file handling, and syntax highlighting) enhances readability and future extensibility.

---

4. Limitations and Suggestions

1. **Error Handling**:
    - Current error handling is minimal. For example, exceptions in file operations are only printed to the console.
    - **Suggestion**: Implement user-friendly error messages using JOptionPane.
2. **Syntax Highlighting Scope**:
    - Limited to a predefined set of keywords and patterns.
    - **Suggestion**: Allow users to define custom keywords or load syntax rules for different programming languages.
3. **Multithreading**:
    - File handling operations, although threaded, lack feedback (e.g., a loading spinner).
    - **Suggestion**: Display a progress indicator or status bar during file operations.
4. **GUI Design**:
    - The layout, while functional, can be further refined for better aesthetics.
    - **Suggestion**: Use layout managers like BorderLayout or GridBagLayout for a professional look.

---

5. Conclusion

The TextEditor program demonstrates an excellent application of Swing in building a functional text editor with advanced features. While the implementation is robust and well-structured, incorporating additional features and refinements will elevate its usability and professional appeal.

This program is a commendable example of combining GUI development with core Java features such as threading, file handling, and regular expressions. It provides a strong foundation for building more sophisticated IDE-like tools.