Documentation Report for Pass2AssemblerGUI

Overview:

The Pass2AssemblerGUI class implements a graphical user interface (GUI) for a Pass 2 assembler using Java Swing. This tool processes assembly code and produces intermediate files, symbol tables, and final output.

Class Structure

Pass2AssemblerGUI

Fields:

JFrame frame: The main window of the application.

JTextArea optabField: Text area for user to input the opcode table (OPTAB).

JTextArea inputField: Text area for user to input the assembly code.

JTextArea outputFileField: Displays the final object code.

JTextArea intermediateFileField: Displays the intermediate code generated from Pass 1.

JTextArea symtabFileField: Displays the symbol table generated from Pass 1.

JTextArea finalOutputField: Displays the final output generated from Pass 2.

JButton assembleButton: Button to trigger the assembly process.

JButton clearButton: Button to clear all input and output fields.

Constructor: Initializes the GUI components and sets up event listeners for buttons.

Methods:

createLabeledOutputPanel(String labelText, JTextArea textArea)

Parameters:

labelText: The label for the output panel.

textArea: The JTextArea to be displayed in the panel.

Returns: A JPanel containing the label and text area.

Description: Creates a labeled output panel for displaying results.

pass1(String input, String optab)

Parameters:

input: Assembly code input as a String.

optab: Opcode table input as a String.

Returns: An array of two Strings, the first containing the intermediate code, and the second containing the symbol table.

Description: Processes the assembly code and generates intermediate code and symbol table (Pass 1).

pass2(String intermediate, String symtab, String optab)

Parameters:

intermediate: The intermediate code from Pass 1.

symtab: The symbol table from Pass 1.

optab: The opcode table.

Returns: An array of two Strings, the first containing the final output, and the second containing the object code.

Description: Processes the intermediate code and symbol table to generate the final output (Pass 2).

pass2Local(List<String[]> optabArr, List<String[]> intermediateArr, List<String[]> symtabArr)

Parameters:

optabArr: List of opcode entries.

intermediateArr: List of intermediate code entries.

symtabArr: List of symbol table entries.

Returns: An array of two Strings, the first being the formatted output and the second containing the object code.

Description: Handles the core logic of Pass 2, generating the object code based on the intermediate code and symbol table.

Main Method

public static void main(String[] args):

Launches the GUI by invoking the Pass2AssemblerGUI constructor.

User Interaction

Input: Users can enter the opcode table and assembly code in the respective text areas.

Assemble: On clicking the "Assemble" button, the input data is processed through Pass 1 and Pass 2.

Output: The results are displayed in the designated output text areas, including the intermediate code, symbol table, and final object code.

Clear: The "Clear" button resets all text areas to allow for new input.

Error Handling

The current implementation does not include explicit error handling. In a production version, checks should be implemented for invalid input formats, numeric conversions, and empty fields.

Dependencies

Java Development Kit (JDK) for compiling and running the application.

Java Swing library for GUI components.

Conclusion

The Pass2AssemblerGUI class provides a user-friendly interface for users to assemble code through a two-pass process, enhancing usability for students and developers learning assembly language. Further improvements could include enhanced error handling and input validation to ensure robustness.