Binary Expression Tree Traversal

Matthew Hunt
huntm2@go.stockton.edu

Overview

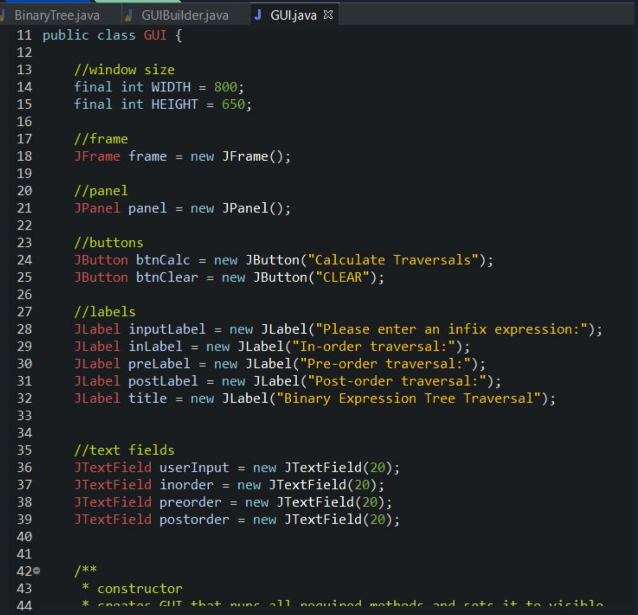
The Binary Expression Tree Traversal Program allows the user to input infix algebraic expressions, and receive the in-order, pre-order, and post-order traversal of the binary tree created with the given expression, as output.

BinaryTree

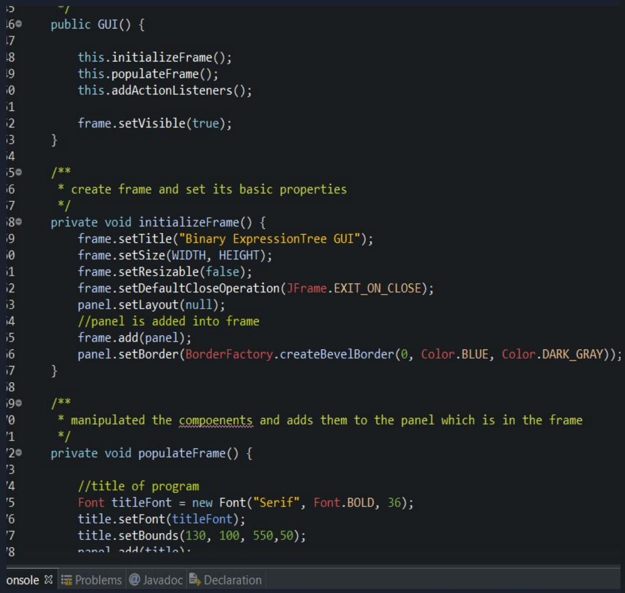
- BinaryTree class creates a modified binary tree that uses nodes and allows the input of infix expressions
- makeExpressionTreeMethod converts the infix input into postfix, then takes the postfix expression ands them to nodes using a stack in the order of normal java order of operations
- Three traversal methods (in,pre,post), traverse the tree according to each algorithm
- The traversal methods return a string of the traversal

GUI

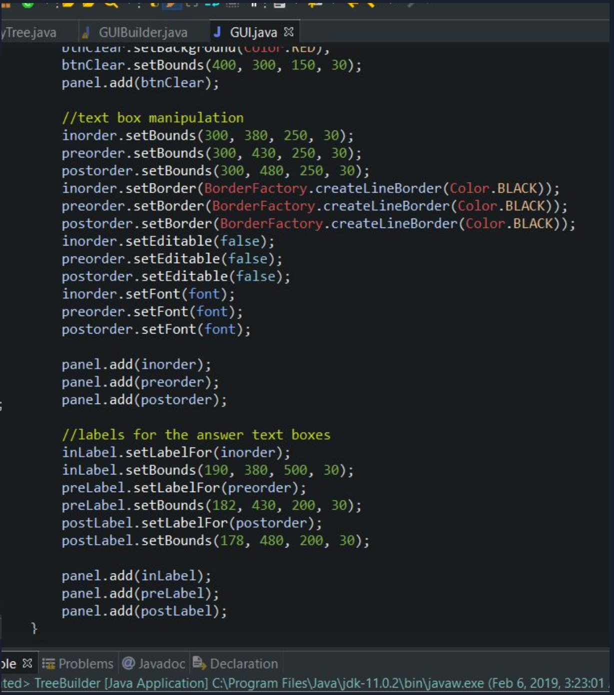
- Graphical interface created
- GUI object creation calls initializeFrame, populateFrame, addActionListeners
- InitializeFrame creates the window and sets the standard settings (size visibility, shape, etc.)
- PopulateFrame create panel, customizes all components, puts components onto screen, adds them to panel, adds panel to frame
- Method sets the look of the GUI
- addActionListeners connect the components like button to action listeners
- Action listeners tie the buttons to creating the tree and displaying the output



```java
public class GUI {

    //window size
    final int WIDTH = 800;
    final int HEIGHT = 650;

    //frame
    JFrame frame = new JFrame();

    //panel
    JPanel panel = new JPanel();

    //buttons
    JButton btnCalc = new JButton("Calculate Traversals");
    JButton btnClear = new JButton("CLEAR");

    //labels
    JLabel inputLabel = new JLabel("Please enter an infix expression:");
    JLabel inLabel = new JLabel("In-order traversal:");
    JLabel preLabel = new JLabel("Pre-order traversal:");
    JLabel postLabel = new JLabel("Post-order traversal:");
    JLabel title = new JLabel("Binary Expression Tree Traversal");


    //text fields
    JTextField userInput = new JTextField(20);
    JTextField inorder = new JTextField(20);
    JTextField preorder = new JTextField(20);
    JTextField postorder = new JTextField(20);


    /**
     * constructor
     * creates GUI that runs all required methods and sets it to visible
```



```java
     */
    public GUI() {

        this.initializeFrame();
        this.populateFrame();
        this.addActionListeners();

        frame.setVisible(true);
    }

    /**
     * create frame and set its basic properties
     */
    private void initializeFrame() {
        frame.setTitle("Binary ExpressionTree GUI");
        frame.setSize(WIDTH, HEIGHT);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel.setLayout(null);
        //panel is added into frame
        frame.add(panel);
        panel.setBorder(BorderFactory.createBevelBorder(0, Color.BLUE, Color.DARK_GRAY));
    }

    /**
     * manipulated the compoenents and adds them to the panel which is in the frame
     */
    private void populateFrame() {

        //title of program
        Font titleFont = new Font("Serif", Font.BOLD, 36);
        title.setFont(titleFont);
        title.setBounds(130, 100, 550,50);
        panel.add(title);
```



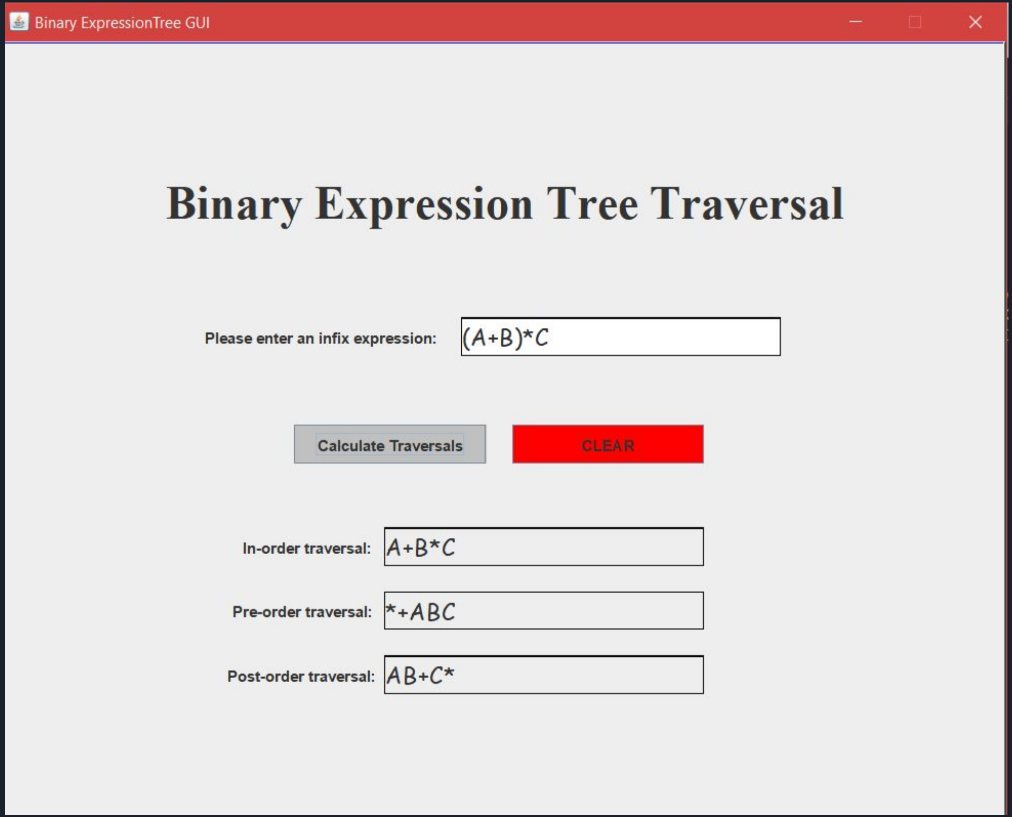
```java
        btnClear.setBackground(Color.RED);
        btnClear.setBounds(400, 300, 150, 30);
        panel.add(btnClear);

        //text box manipulation
        inorder.setBounds(300, 380, 250, 30);
        preorder.setBounds(300, 430, 250, 30);
        postorder.setBounds(300, 480, 250, 30);
        inorder.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        preorder.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        postorder.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        inorder.setEditable(false);
        preorder.setEditable(false);
        postorder.setEditable(false);
        inorder.setFont(font);
        preorder.setFont(font);
        postorder.setFont(font);

        panel.add(inorder);
        panel.add(preorder);
        panel.add(postorder);

        //labels for the answer text boxes
        inLabel.setLabelFor(inorder);
        inLabel.setBounds(190, 380, 500, 30);
        preLabel.setLabelFor(preorder);
        preLabel.setBounds(182, 430, 200, 30);
        postLabel.setLabelFor(postorder);
        postLabel.setBounds(178, 480, 200, 30);

        panel.add(inLabel);
        panel.add(preLabel);
        panel.add(postLabel);
    }
```

```
ted> TreeBuilder [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Feb 6, 2019, 3:23:01
```

GUIBuilder

- Main Method
- All it does is create a new GUI object
- The creation of the GUI then calls all the other necessary methods

interface



Binary Expression Tree Traversal

Please enter an infix expression: (A+B)*C

In-order traversal: A+B*C

Pre-order traversal: *+ABC

Post-order traversal: AB+C*