**IDSS Project Documentation**

**Breadth-First Strategy & Depth-First Strategy**

**Supervised by:**

**DR. Abeer Amer**

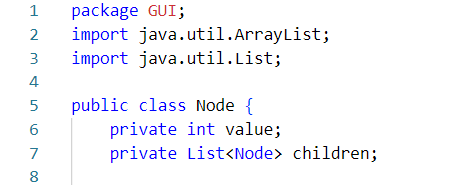
**MS. Nourhan Nasr**

**Team members:**

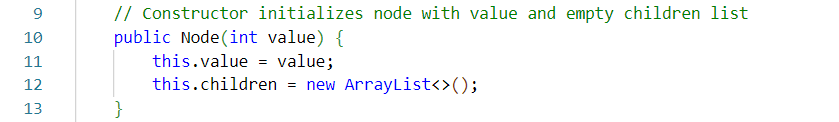
|  |  |  |
| --- | --- | --- |
| Name | Sec. | Seat n. |
| Saad Abd El-Samei Mohamed | **G4** | **64221** |
| Mahmoud Mohsen Ali El-Saftawy | **G5** | **64202** |
| Mohamed Gamal Mohamed | **G5** | **64227** |
| Mohamed Abd El-Raouf El-Fawi | **G3** | **64283** |
| Yousef Ahmed El-sayed | **G5** | **74508** |
| Ro'ya Ahmed Mohamed anwar | **G4** | **64220** |
| Salma Ebrahim El-sayed Ebrahim | **G4** | **64296** |
| Christina Sobhy Abdo Rezek allah George | **G5** | **64260** |
| Esraa Ali Ahmed sleem | **G3** | **64300** |
| Yara Khaled Ahmed Mohamed Okasha | **G5** | **64265** |

**Breadth-First Strategy & Depth-First Strategy**

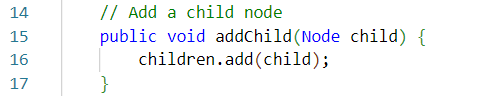
1. **Node.java :**

****

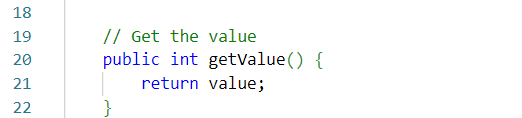
* private int value: Stores the integer value of the node.
* private List<Node> children: Stores the list of child nodes.



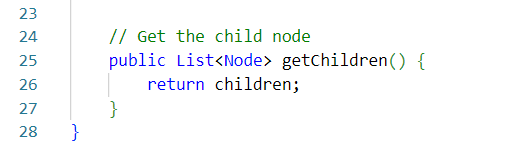
* Initializes a node with a given value and an empty list of children.



* Adds a child node to the current node.

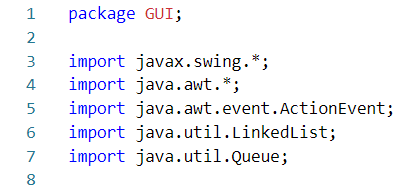


* Returns the value of the node.

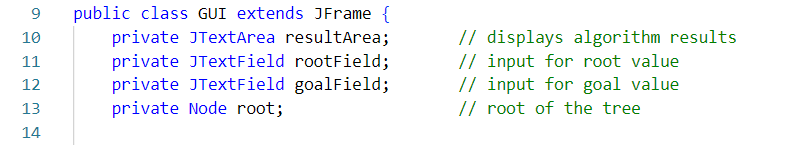


* Returns the list of child nodes.

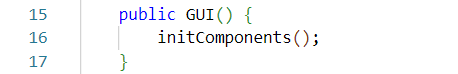
1. **GUI.java :**

**  
Imports**:

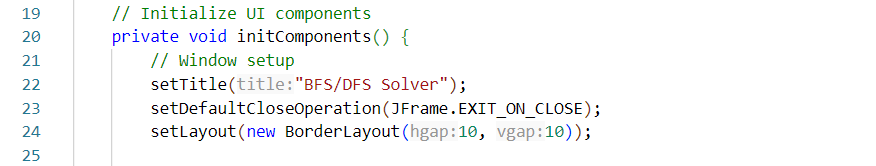
* javax.swing.\*: Swing components for GUI
* java.awt.\*: AWT classes for layout management
* ActionEvent: Handles button click events
* LinkedList & Queue: For tree construction

****

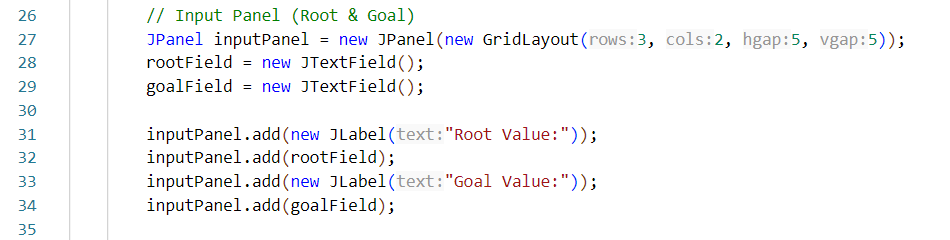
* **Extends JFrame**: Creates main application window
* **Components**:
* resultArea: Displays algorithm outputs
* rootField/goalField: Text inputs for values
* root: Stores tree structure



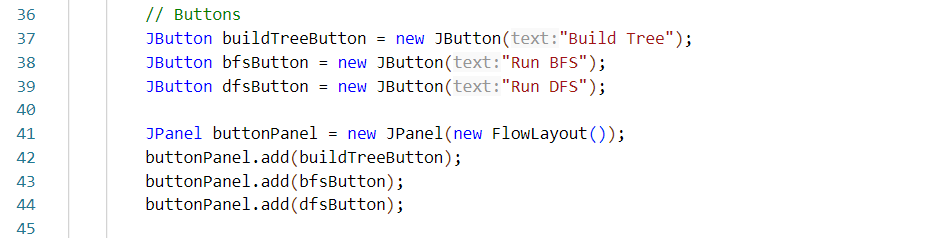
* Initializes UI components when GUI is created



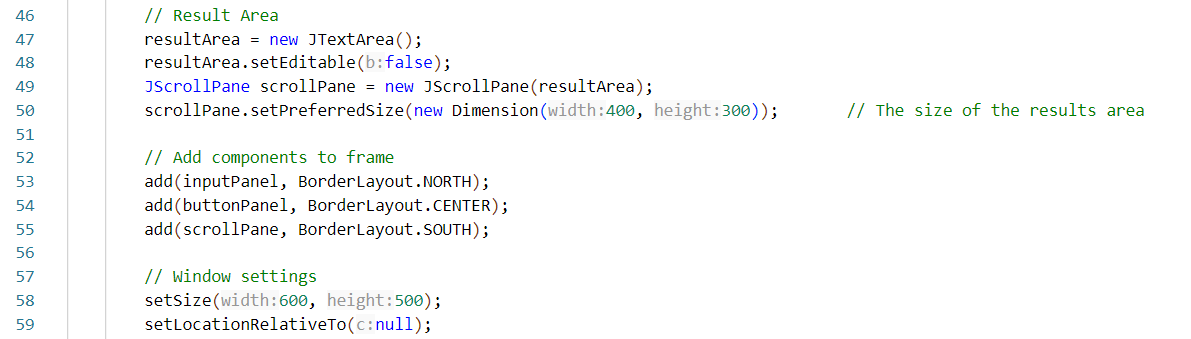
* **Window Configuration**:
* Sets title and close operation
* Uses BorderLayout with 10px gaps



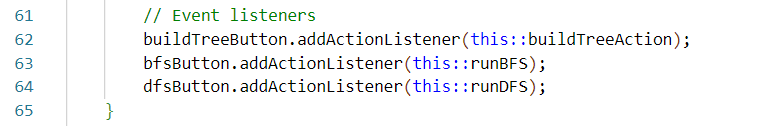
* **Input Panel**:
  + Grid layout (3 rows, 2 columns)
  + Contains labeled text fields for root/goal values



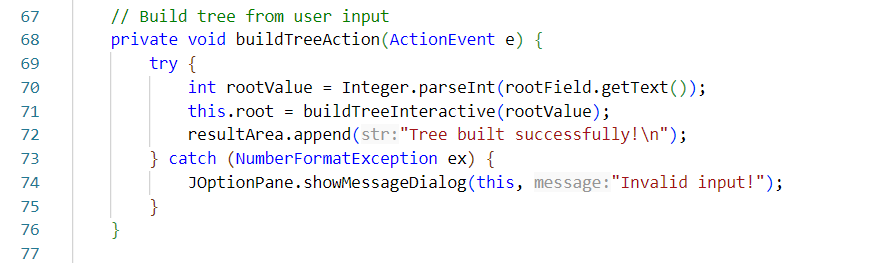
* **Button Panel**:
  + Uses FlowLayout for horizontal button arrangement
  + Contains tree/build/algorithm buttons



* **Result Area**:
  + JTextArea with scroll pane
  + Fixed preferred size for consistent UI



* **Event Handling**:
  + Connects buttons to action methods



* **Purpose**: Entry point for tree construction
* **Workflow**:
* Parses root value from text field
* Calls interactive builder
* Handles invalid inputs with error dialog



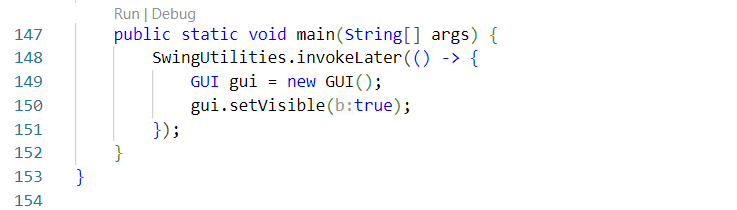
* **Algorithm**: Level-order tree construction
* **Features**:
* Uses queue for BFS-like node processing
* Interactive dialogs for child nodes
* Input validation and error handling



* **Workflow**:
* Checks for existing tree
* Parses goal value
* Calls BFS algorithm
* Displays results with auto-scroll

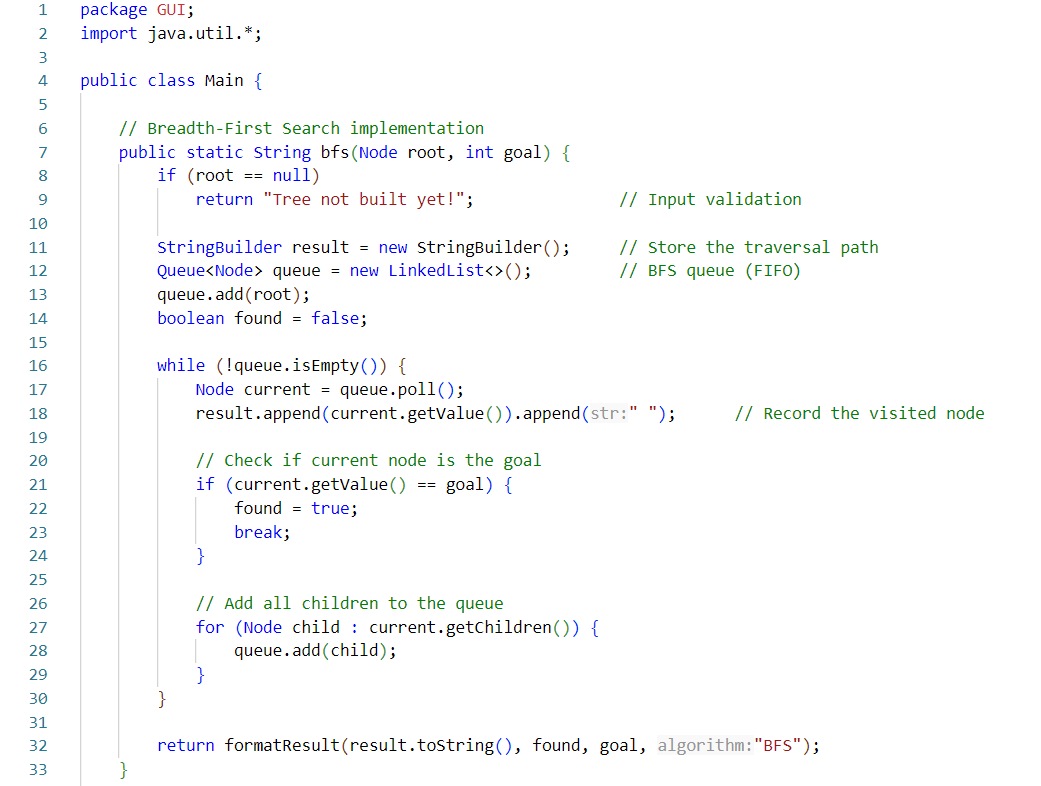


* **Workflow**:
* Checks for existing tree
* Parses goal value
* Calls DFS algorithm
* Displays results with auto-scroll

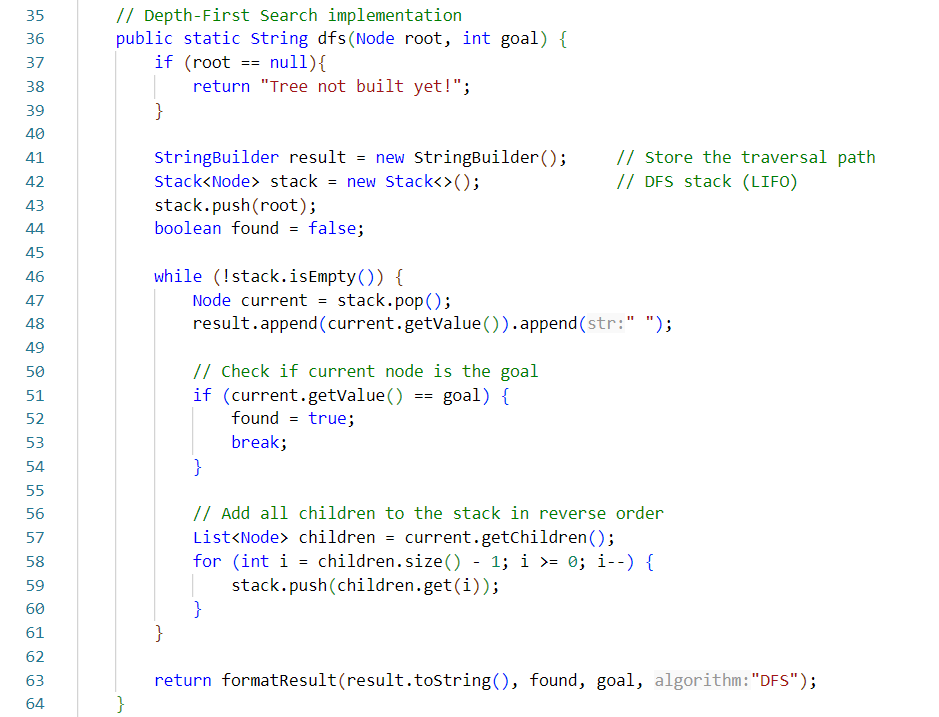


* **SwingUtilities.invokeLater()**: Ensures thread-safe GUI creation
* **GUI Lifecycle**: Creates and displays the application window

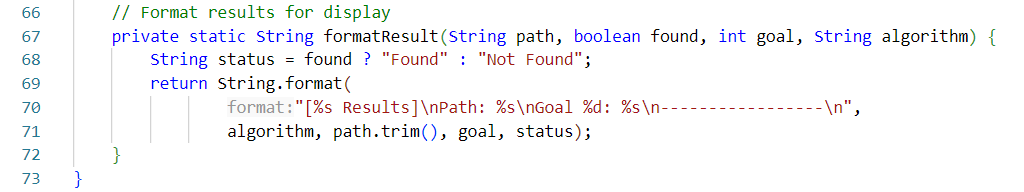
1. **Main.java :**

****

* **Imports**:
* java.util.\*: For collections (Queue, Stack, LinkedList)
* **Key Features**:
* **Queue Behavior**: Processes nodes level-by-level (FIFO)
* **Termination**: Stops early if goal is found
* **Path Recording**: Appends visited nodes to StringBuilder

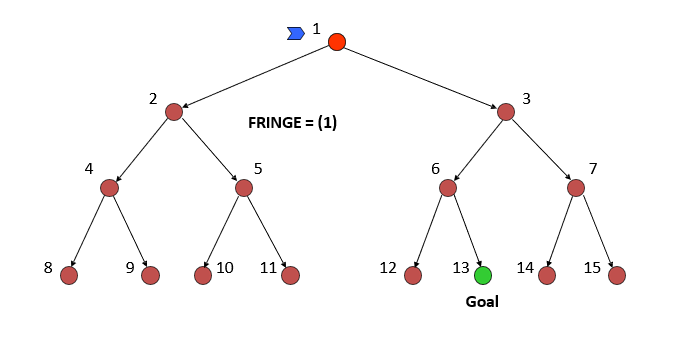


* **Key Features**:
* **Stack Behavior**: Processes deepest nodes first (LIFO)
* **Child Ordering**: Reversed to maintain left-first traversal
* **Path Recording**: Similar to BFS but with depth-first logic

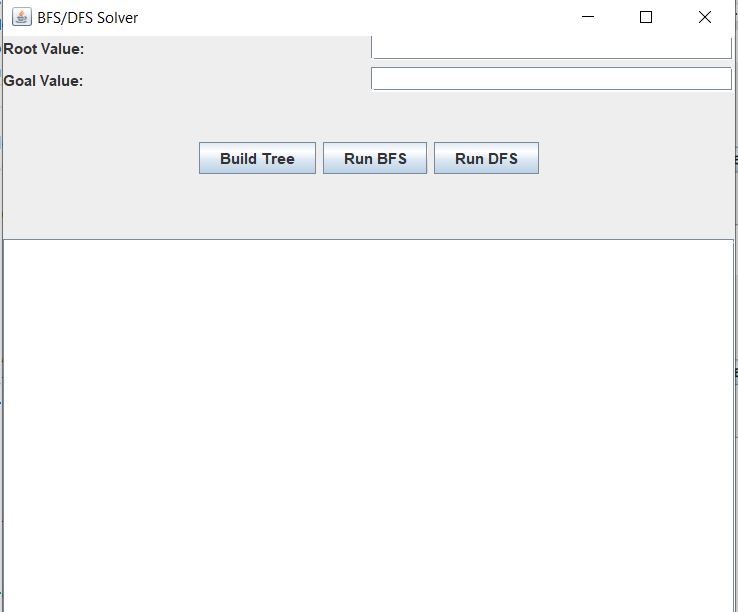
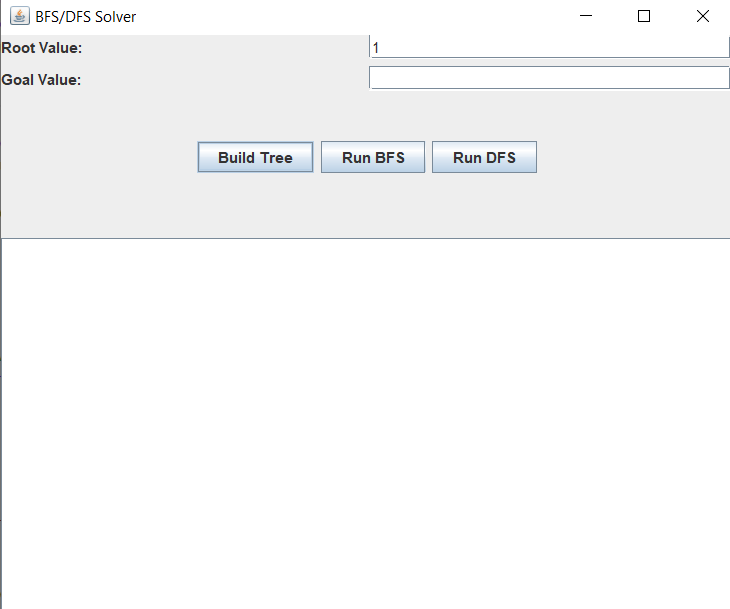


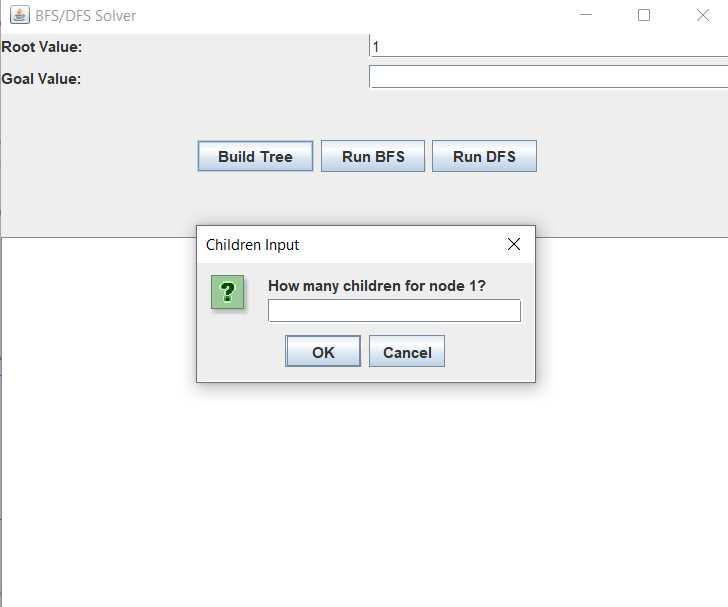
* **Formatting Logic**:
* **Header**: Identifies algorithm used
* **Path**: Shows visited nodes in order
* **Status**: Clear found/not found indication

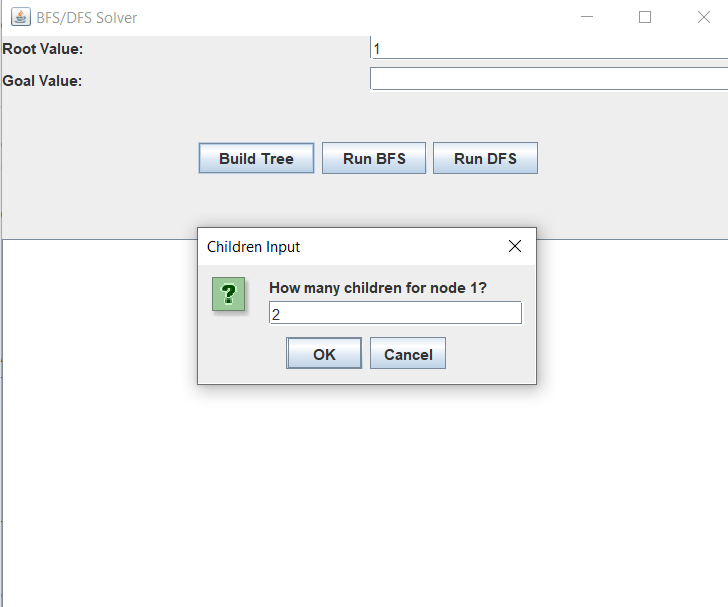
**Example 1**:



**Run The Code:**

**** ****

****

****