// Zakery Cumbie

// CSE 464 Fall 2023

The Main.java file hosts the methods that implement the four features, those being parseGraph, addNode & addNodes, addEdge, and outputDOTGraph. The main method is a driver that will manually put some inputs into the methods so that the outputs of the methods can be seen, some are errors and some are standard outputs.

For example,

```
parseGraph("C:/Users/ldf08/IdeaProjects/CSE464Project/input.dot");
```

Will return the parsed information from the input.dot file, which returns

```
Number of Nodes: 4
Number of Edges: 5
Node Labels: [A, B, C, D]
Edge Directions:
A -> B
B -> C
C -> A
A -> C
D -> B
```

The parseGraph method uses a file path in string form to find input. The input should follow the format "X -> X" or the information will not be stored.

So long as the given input is formatted correctly, it will sort for duplicate nodes and edges, and give numbers for both.

addNode(String s)

This method will add a Node labeled by the given input string. It will first check that the node does not already exist in the graph object.

addNodes(String[] labels)

This method performs the same function as addNode but accross multiple strings for various labels. Checks for repeat labels are done at each instance of a label being processed.

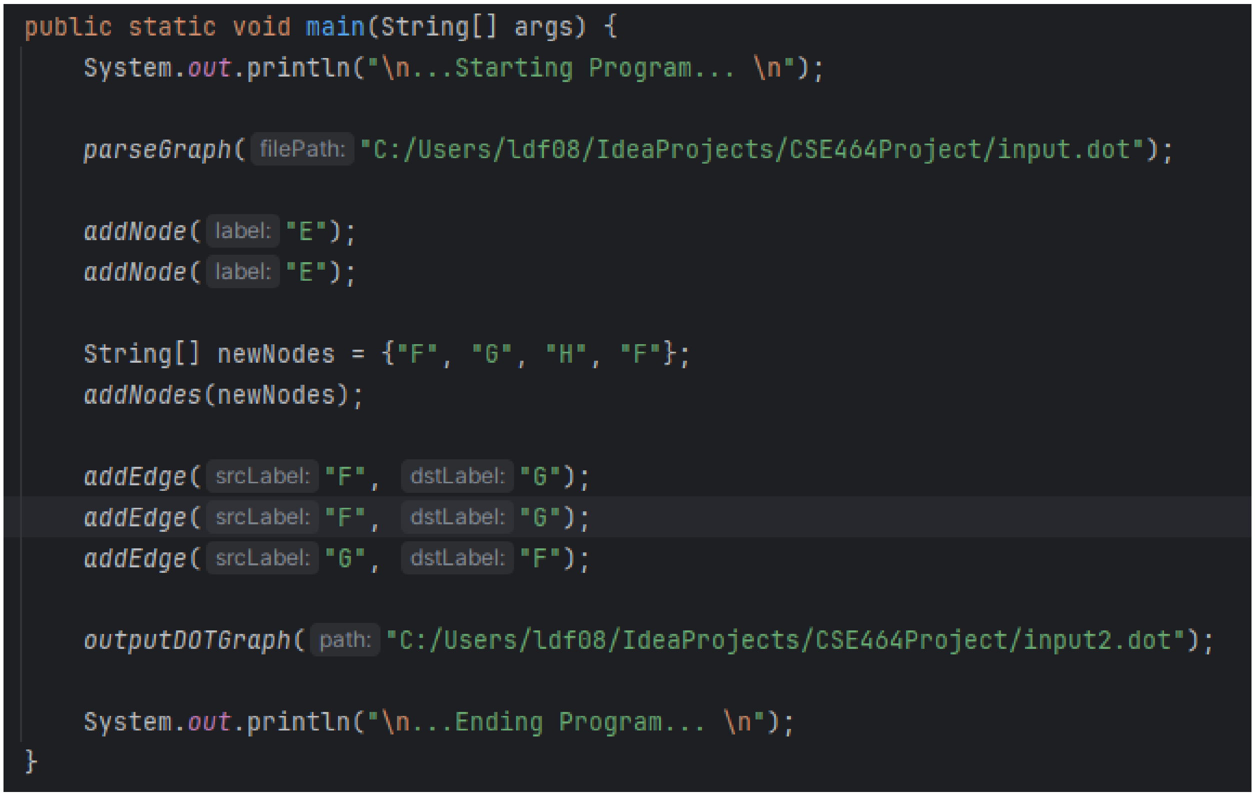
addEdge(source, destination)

This method adds an edge, starting from source and pointing towards destination. Checks for edges in the SAME direction will catch duplicates, but in the case of A -> B, B -> A, these edges will be added as they are valid.

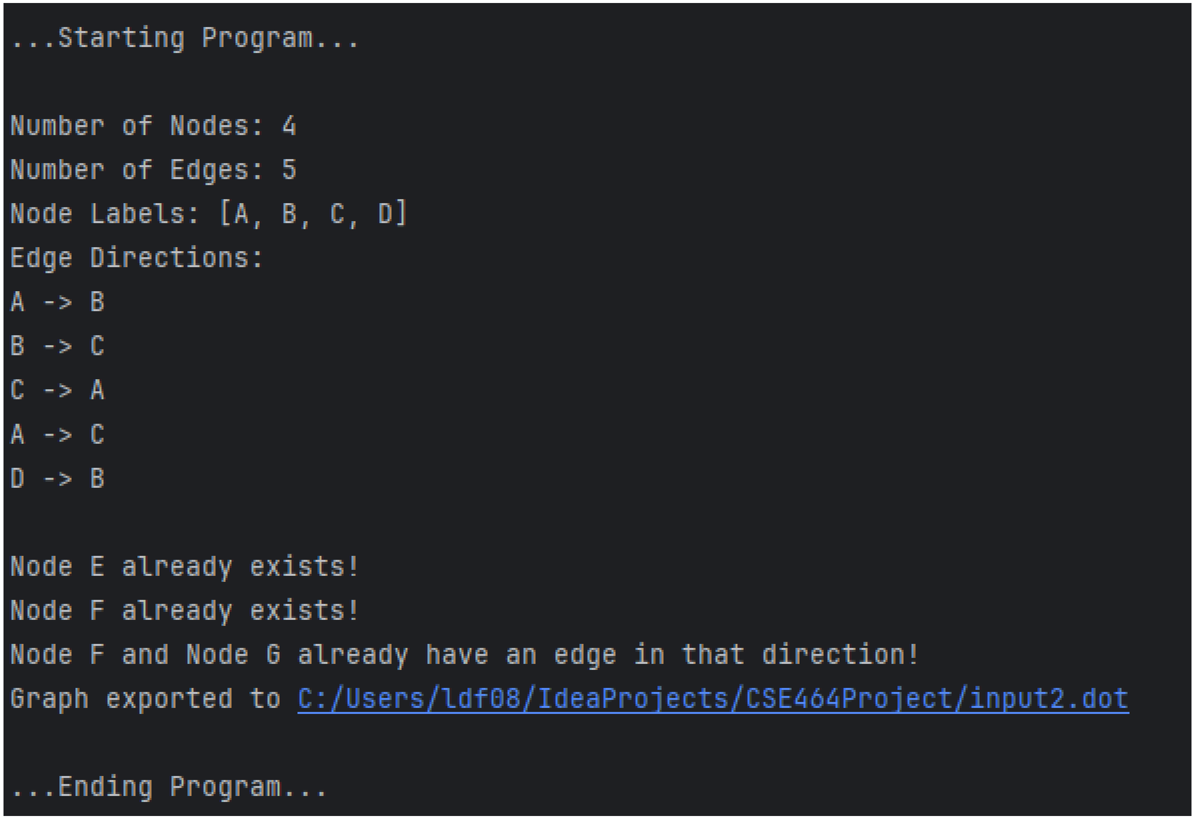
outDOTGraph(String filePath)

This method will use a filePath to take the current instance of graph object and turn it into a dot file.

Below are pictures of sample input and output that I achieved through using the main function. In order to see the console output of each function, use the “main” method as a driver.



```
public static void main(String[] args) {
    System.out.println("\n...Starting Program... \n");

    parseGraph( filePath: "C:/Users/ldf08/IdeaProjects/CSE464Project/input.dot");

    addNode( label: "E");
    addNode( label: "E");

    String[] newNodes = {"F", "G", "H", "F"};
    addNodes(newNodes);

    addEdge( srcLabel: "F",  dstLabel: "G");
    addEdge( srcLabel: "F",  dstLabel: "G");
    addEdge( srcLabel: "G",  dstLabel: "F");

    outputDOTGraph( path: "C:/Users/ldf08/IdeaProjects/CSE464Project/input2.dot");

    System.out.println("\n...Ending Program... \n");
}
```



```
...Starting Program...

Number of Nodes: 4
Number of Edges: 5
Node Labels: [A, B, C, D]
Edge Directions:
A -> B
B -> C
C -> A
A -> C
D -> B

Node E already exists!
Node F already exists!
Node F and Node G already have an edge in that direction!
Graph exported to C:/Users/ldf08/IdeaProjects/CSE464Project/input2.dot

...Ending Program...
```

The file featureTester, contains the unit tests for each feature to be used by maven. These do not output anything unless an error is caught in testing, but will test for the expected outputs from each feature.

The link below contains the full history of commits for the GitHub repository relating to the Main.java file, in which a feature was added or modified.

https://github.com/Khrone99/CSE-464-2023-zcumbie/commits/main/Main.java