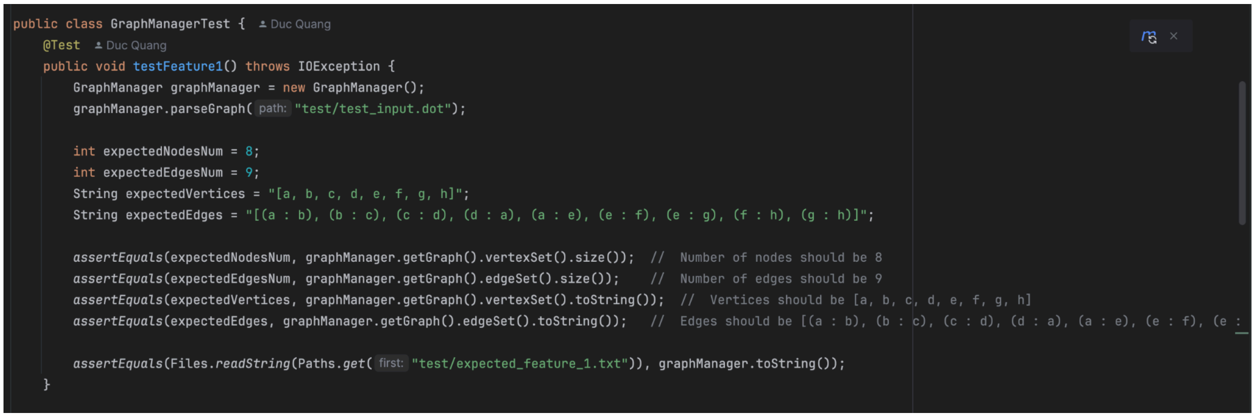
There is a test class for 4 features along with their expected values and expected output. There is also a test folder in the source code where there are example inputs.



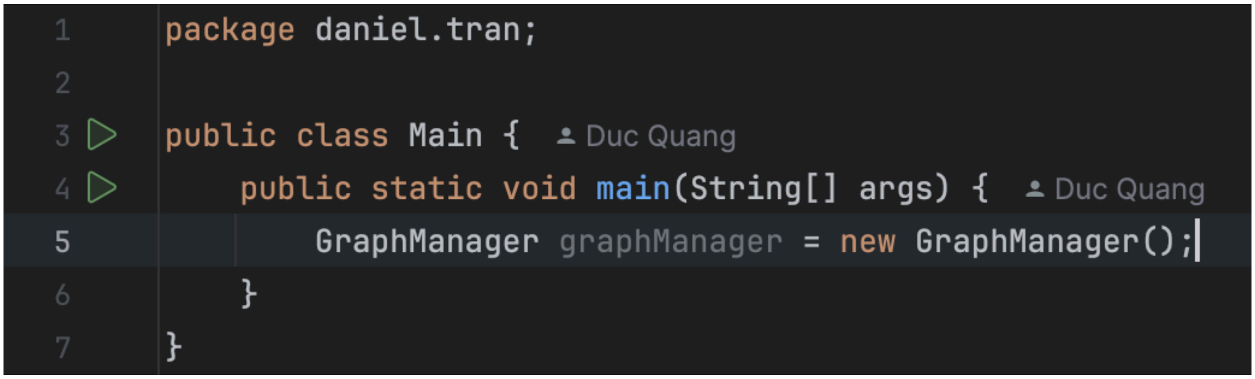
```
public class GraphManagerTest {  Duc Quang
    @Test  Duc Quang
    public void testFeature1() throws IOException {
        GraphManager graphManager = new GraphManager();
        graphManager.parseGraph( path: "test/test_input.dot");

        int expectedNodesNum = 8;
        int expectedEdgesNum = 9;
        String expectedVertices = "[a, b, c, d, e, f, g, h]";
        String expectedEdges = "[(a : b), (b : c), (c : d), (d : a), (a : e), (e : f), (e : g), (f : h), (g : h)]";

        assertEquals(expectedNodesNum, graphManager.getGraph().vertexSet().size());  //  Number of nodes should be 8
        assertEquals(expectedEdgesNum, graphManager.getGraph().edgeSet().size());    //  Number of edges should be 9
        assertEquals(expectedVertices, graphManager.getGraph().vertexSet().toString());  //  Vertices should be [a, b, c, d, e, f, g, h]
        assertEquals(expectedEdges, graphManager.getGraph().edgeSet().toString());   //  Edges should be [(a : b), (b : c), (c : d), (d : a), (a : e), (e : f), (e :

        assertEquals(Files.readString(Paths.get( first: "test/expected_feature_1.txt")), graphManager.toString());
    }
```

If you would like to test alternative features by choice, you can test on the main function/main class.



```
package daniel.tran;

public class Main {  Duc Quang
    public static void main(String[] args) {  Duc Quang
        GraphManager graphManager = new GraphManager();
    }
}
```