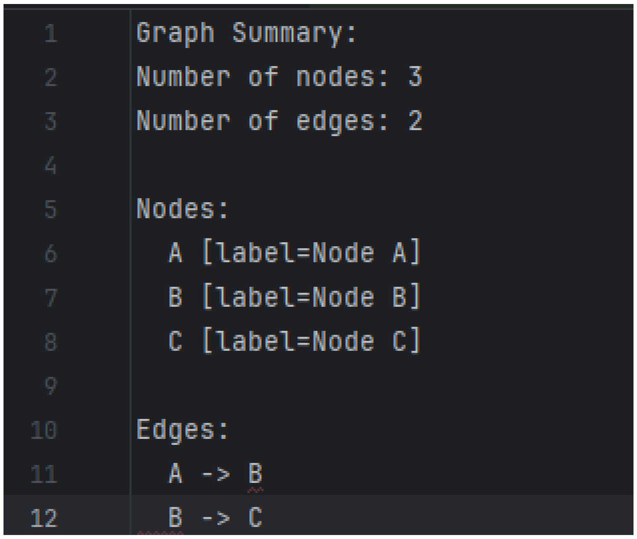
Feature 1 and JUnit 1:
https://github.com/Msajimi/CSE466PRJ/commit/c2c9b4e99fca076cd2e23aa3f937d597931f869d
https://github.com/Msajimi/CSE466PRJ/commit/ab5b3450937326c178a27a4c408f31fc548551cc

Tests:

1. Parse Graph Test 1 Expected Output:

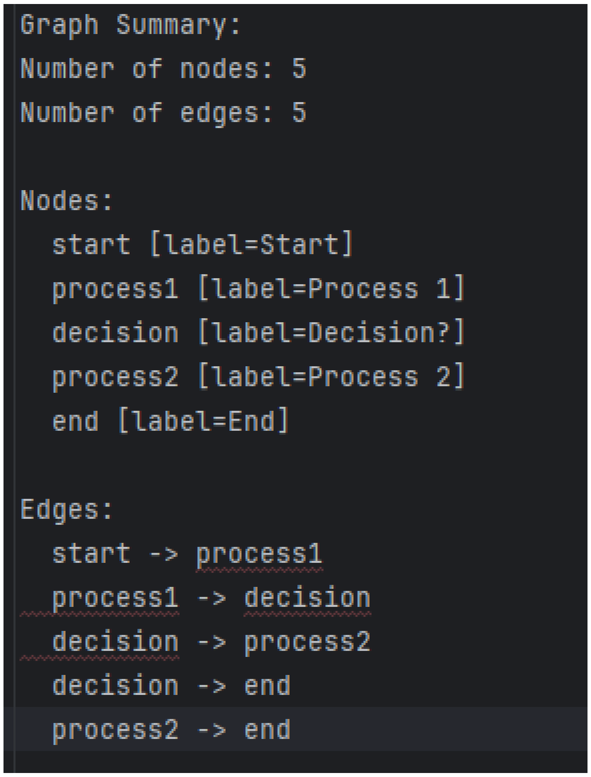


```
Graph Summary:
Number of nodes: 3
Number of edges: 2

Nodes:
  A [label=Node A]
  B [label=Node B]
  C [label=Node C]

Edges:
  A -> B
  B -> C
```

2. Parse Graph Test 2 Expected Output:

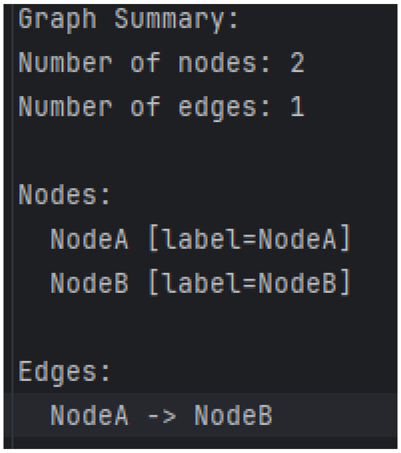


```
Graph Summary:
Number of nodes: 5
Number of edges: 5

Nodes:
  start [label=Start]
  process1 [label=Process 1]
  decision [label=Decision?]
  process2 [label=Process 2]
  end [label=End]

Edges:
  start -> process1
  process1 -> decision
  decision -> process2
  decision -> end
  process2 -> end
```

3. Add Edge Test 1 Expected Output

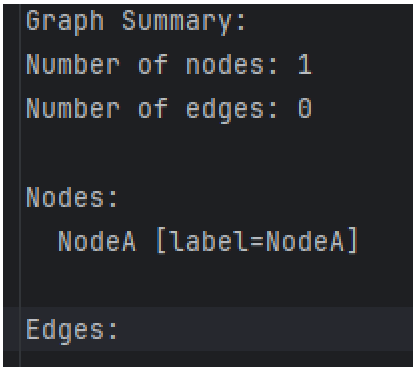


```
Graph Summary:
Number of nodes: 2
Number of edges: 1

Nodes:
  NodeA [label=NodeA]
  NodeB [label=NodeB]

Edges:
  NodeA -> NodeB
```

4. Add Edge Test 2 Expected Output

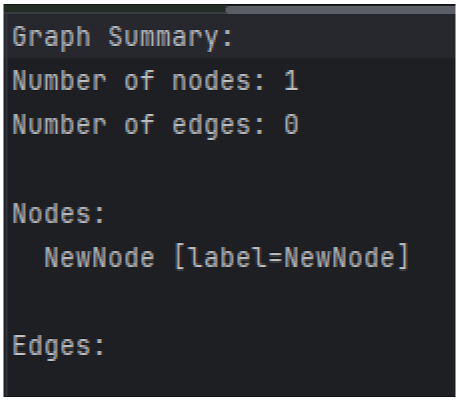


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  NodeA [label=NodeA]

Edges:
```

5. Add Node Test 1 Expected Output

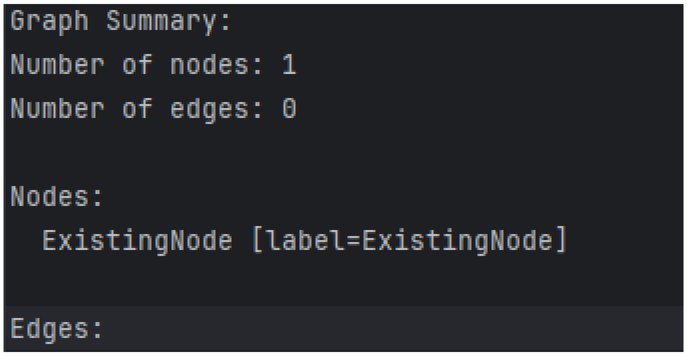


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  NewNode [label=NewNode]

Edges:
```

6. Add Node Test 2 Expected Output



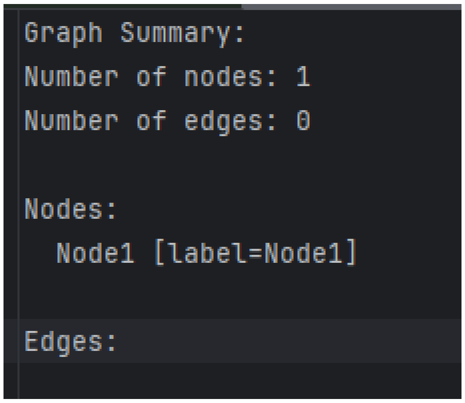
```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  ExistingNode [label=ExistingNode]

Edges:
```

Feature 2 and JUnit Test 2:
https://github.com/Msajimi/CSE466PRJ/commit/1273408182f7a617185e1135c2d5cf176bc6e8ba
https://github.com/Msajimi/CSE466PRJ/commit/8c3c2d2bdd867f70fd35a0a970e2312fad3e000f

1. Add Single Node Test Expected Output

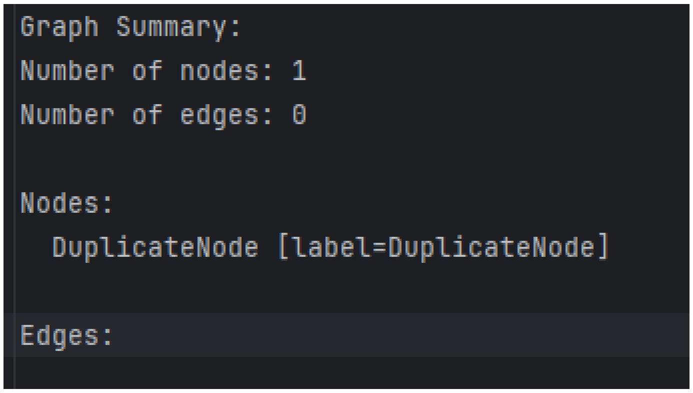


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  Node1 [label=Node1]

Edges:
```

2. Add Duplicate Node Test Expected Output

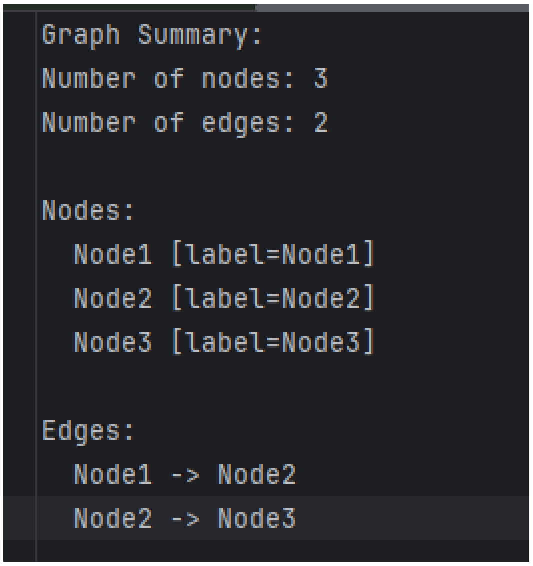


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  DuplicateNode [label=DuplicateNode]

Edges:
```

3. Add Multiple Node Test Expected Output

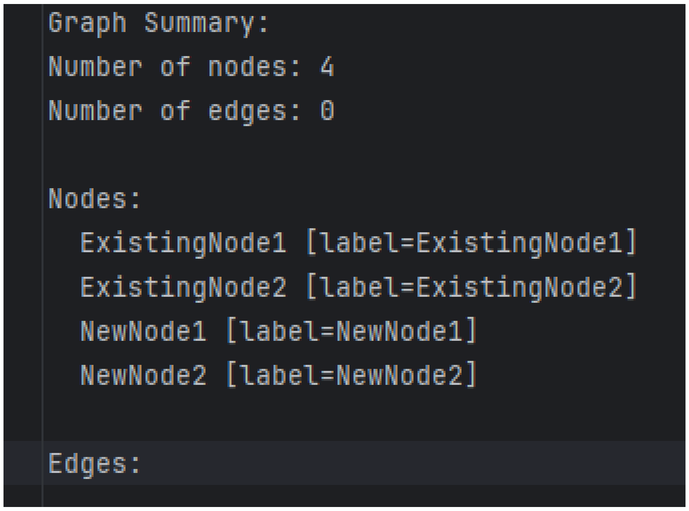


```
Graph Summary:
Number of nodes: 3
Number of edges: 2

Nodes:
  Node1 [label=Node1]
  Node2 [label=Node2]
  Node3 [label=Node3]

Edges:
  Node1 -> Node2
  Node2 -> Node3
```

4. Add Mixed Nodes Expected Output



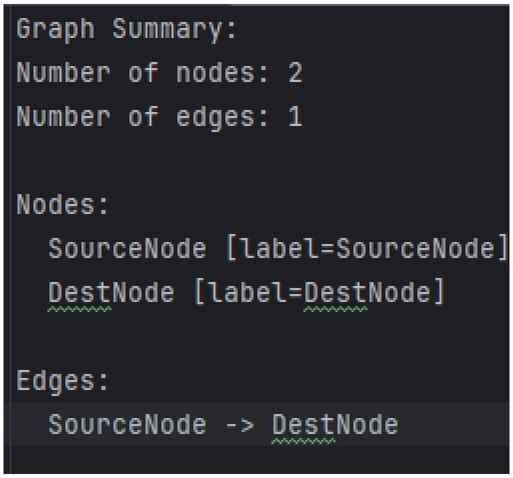
```
Graph Summary:
Number of nodes: 4
Number of edges: 0

Nodes:
  ExistingNode1 [label=ExistingNode1]
  ExistingNode2 [label=ExistingNode2]
  NewNode1 [label=NewNode1]
  NewNode2 [label=NewNode2]

Edges:
```

Feature 3 and JUnit 3
https://github.com/Msajimi/CSE466PRJ/commit/60ee8fc3f2f5f94df03b68320e00ea8b2df6df7f

1. Add Valid Test Expected Output

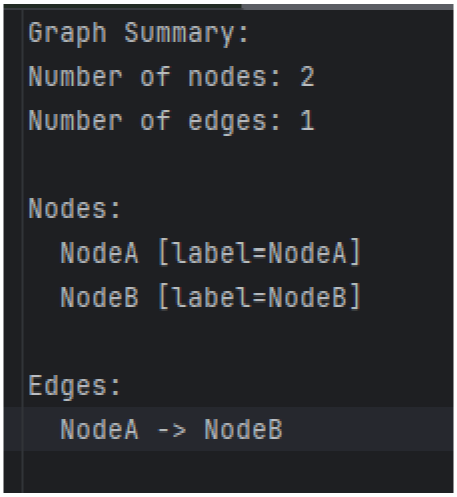


```
Graph Summary:
Number of nodes: 2
Number of edges: 1

Nodes:
  SourceNode [label=SourceNode]
  DestNode [label=DestNode]

Edges:
  SourceNode -> DestNode
```

2. Add duplicate edge test

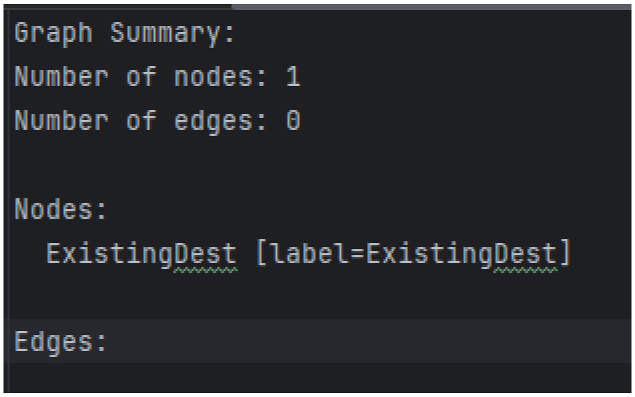


```
Graph Summary:
Number of nodes: 2
Number of edges: 1

Nodes:
  NodeA [label=NodeA]
  NodeB [label=NodeB]

Edges:
  NodeA -> NodeB
```

3. Add non existent edge (source)

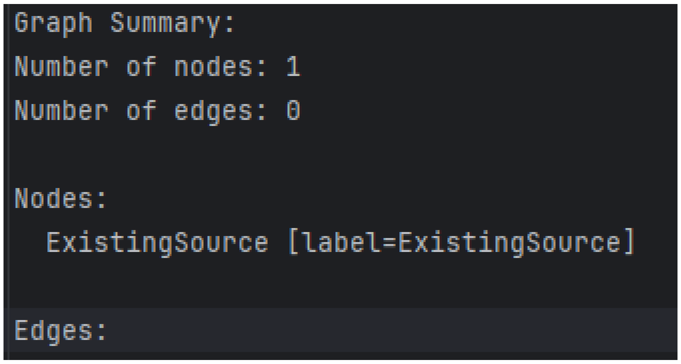


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  ExistingDest [label=ExistingDest]

Edges:
```

4. Add non existent edge (dest)

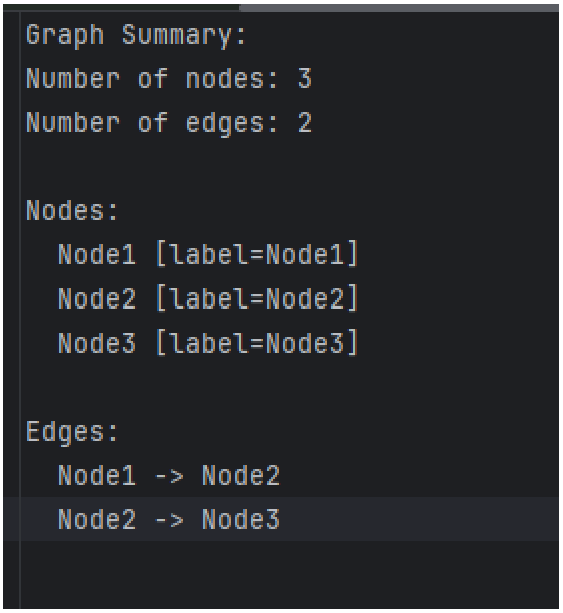


```
Graph Summary:
Number of nodes: 1
Number of edges: 0

Nodes:
  ExistingSource [label=ExistingSource]

Edges:
```

5. Add multiple edge



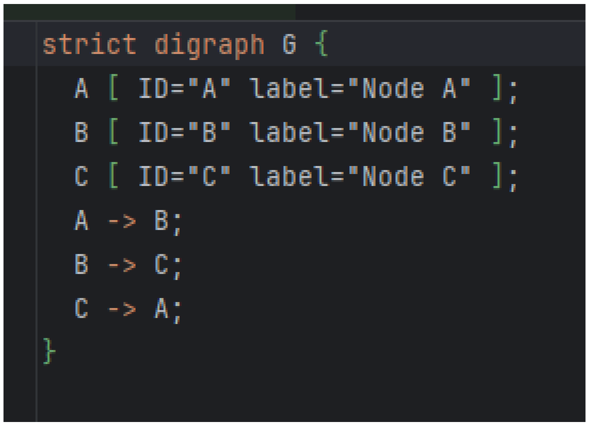
```
Graph Summary:
Number of nodes: 3
Number of edges: 2

Nodes:
  Node1 [label=Node1]
  Node2 [label=Node2]
  Node3 [label=Node3]

Edges:
  Node1 -> Node2
  Node2 -> Node3
```

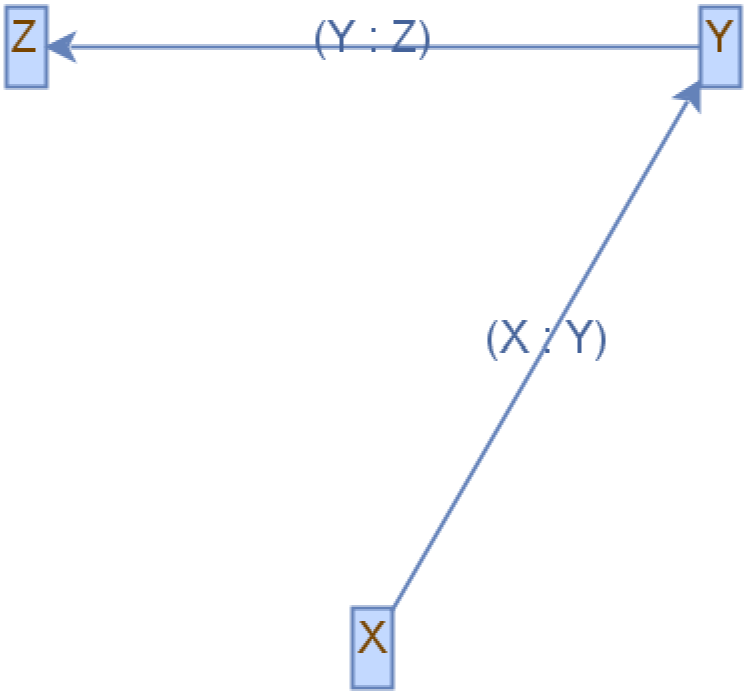
Feature 4 and JUnit 4
https://github.com/Msajimi/CSE466PRJ/commit/7e152d13b20c0e2570ee592232dbcbdc3b08e0cb

1. outputDotGraph Expected output

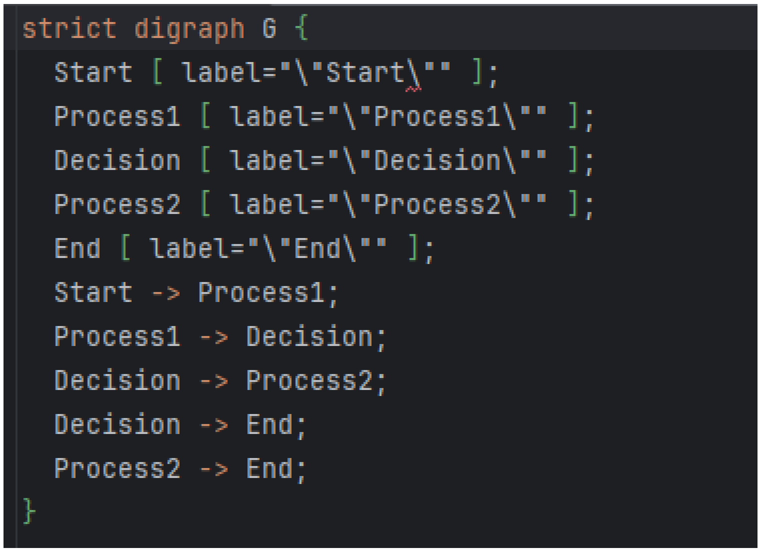


```
strict digraph G {
  A [ ID="A" label="Node A" ];
  B [ ID="B" label="Node B" ];
  C [ ID="C" label="Node C" ];
  A -> B;
  B -> C;
  C -> A;
}
```

2. Test Output Dot Graph PNG



3. Test complex graph



```
strict digraph G {
  Start [ label="\"Start\"" ];
  Process1 [ label="\"Process1\"" ];
  Decision [ label="\"Decision\"" ];
  Process2 [ label="\"Process2\"" ];
  End [ label="\"End\"" ];
  Start -> Process1;
  Process1 -> Decision;
  Decision -> Process2;
  Decision -> End;
  Process2 -> End;
}
```