Project 01 Part 1 README

Description

This project implements a Java application for parsing, manipulating, and visualizing DOT format graphs.

Prerequisites:

Java17+
Maven 3.6+
IntelliJ IDEA/CE

Building Project:

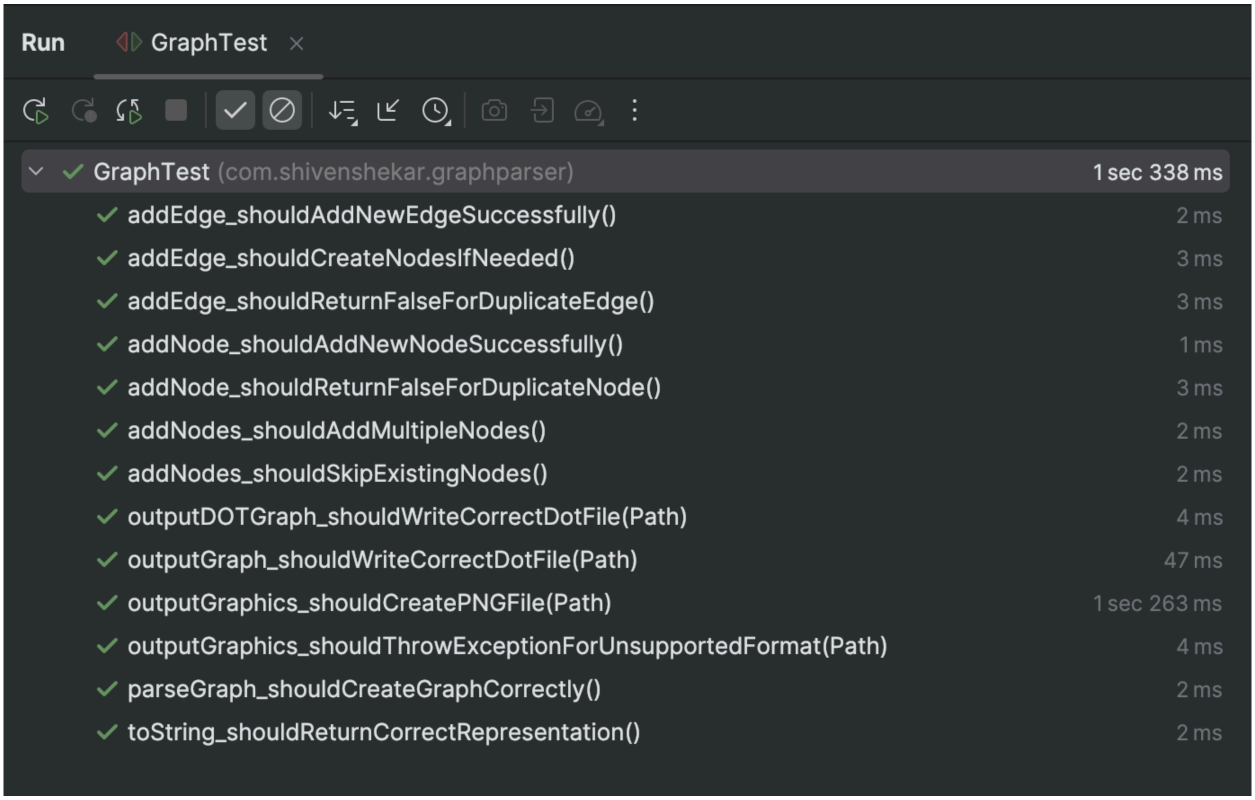
Run: mvn package

Complete Unit Testing

Run: mvn test

OR

1. Open in IntelliJ
2. Go to GraphTest.java
3. Click on run in class GraphTest {
4. Expected Output:



```
✓ GraphTest (com.shivenshekar.graphparser)                          1 sec 338 ms
    ✓ addEdge_shouldAddNewEdgeSuccessfully()                                2 ms
    ✓ addEdge_shouldCreateNodesIfNeeded()                                   3 ms
    ✓ addEdge_shouldReturnFalseForDuplicateEdge()                           3 ms
    ✓ addNode_shouldAddNewNodeSuccessfully()                                1 ms
    ✓ addNode_shouldReturnFalseForDuplicateNode()                           3 ms
    ✓ addNodes_shouldAddMultipleNodes()                                     2 ms
    ✓ addNodes_shouldSkipExistingNodes()                                    2 ms
    ✓ outputDOTGraph_shouldWriteCorrectDotFile(Path)                        4 ms
    ✓ outputGraph_shouldWriteCorrectDotFile(Path)                          47 ms
    ✓ outputGraphics_shouldCreatePNGFile(Path)                    1 sec 263 ms
    ✓ outputGraphics_shouldThrowExceptionForUnsupportedFormat(Path)         4 ms
    ✓ parseGraph_shouldCreateGraphCorrectly()                               2 ms
    ✓ toString_shouldReturnCorrectRepresentation()                          2 ms
```

Features:

1. Parsing DOT File

Commit Link:
https://github.com/shiven01/CSE-464-2025-sshekar9/commit/901b5989750f3d0262a40c079a2058e872991765

Expected Output:
After running mvn test, look for the passing test cases:

- parseGraph_shouldCreateGraphCorrectly() - Tests the core parsing functionality
- toString_shouldReturnCorrectRepresentation() - Tests the string representation
- outputGraph_shouldWriteCorrectDotFile(Path) - Tests writing the graph to a DOT file

2. Adding Nodes to Graph

Commit Link:
https://github.com/shiven01/CSE-464-2025-sshekar9/commit/10198d6e2b3b40aa8eb2b7fb02c0d104c73df152

Expected Output:
After running mvn test, look for:

- addNode_shouldAddNewNodeSuccessfully() - Tests adding a single new node
- addNode_shouldReturnFalseForDuplicateNode() - Tests duplicate node handling
- addNodes_shouldAddMultipleNodes() - Tests adding multiple nodes at once
- addNodes_shouldSkipExistingNodes() - Tests adding a mix of new and existing nodes

3. Adding Edges to Graph

Commit Link:
https://github.com/shiven01/CSE-464-2025-sshekar9/commit/10c0f66cabbdc5112bc3387b6e0045b774ab9866

Expected Output:
After running mvn test, look for:

- addEdge_shouldAddNewEdgeSuccessfully() - Tests adding a new edge between nodes
- addEdge_shouldReturnFalseForDuplicateEdge() - Tests duplicate edge handling
- addEdge_shouldCreateNodesIfNeeded() - Tests creating missing nodes automatically when adding an edge

4. Output to Graph

Commit Link:
https://github.com/shiven01/CSE-464-2025-sshekar9/commit/1f86be4e4bd971880e5b9ad6a3ad34d485aca051

Expected Output:
After running mvn test, look for:

- outputDOTGraph_shouldWriteCorrectDotFile(Path) - Tests DOT format output
- outputGraphics_shouldCreatePNGFile(Path) - Tests PNG image generation
- outputGraphics_shouldThrowExceptionForUnsupportedFormat(Path) - Tests error handling for unsupported formats

Project Structure

```
src/
  main/
  java/
    com/
      shivenshekar/
        graphparser/
          Graph.java
          Main.java
  resources/
    sample.dot         // Sample DOT file for testing
test/
  java/
    com/
      shivenshekar/
        graphparser/
          GraphTest.java // Unit tests
```