GIT Department of Computer Engineering
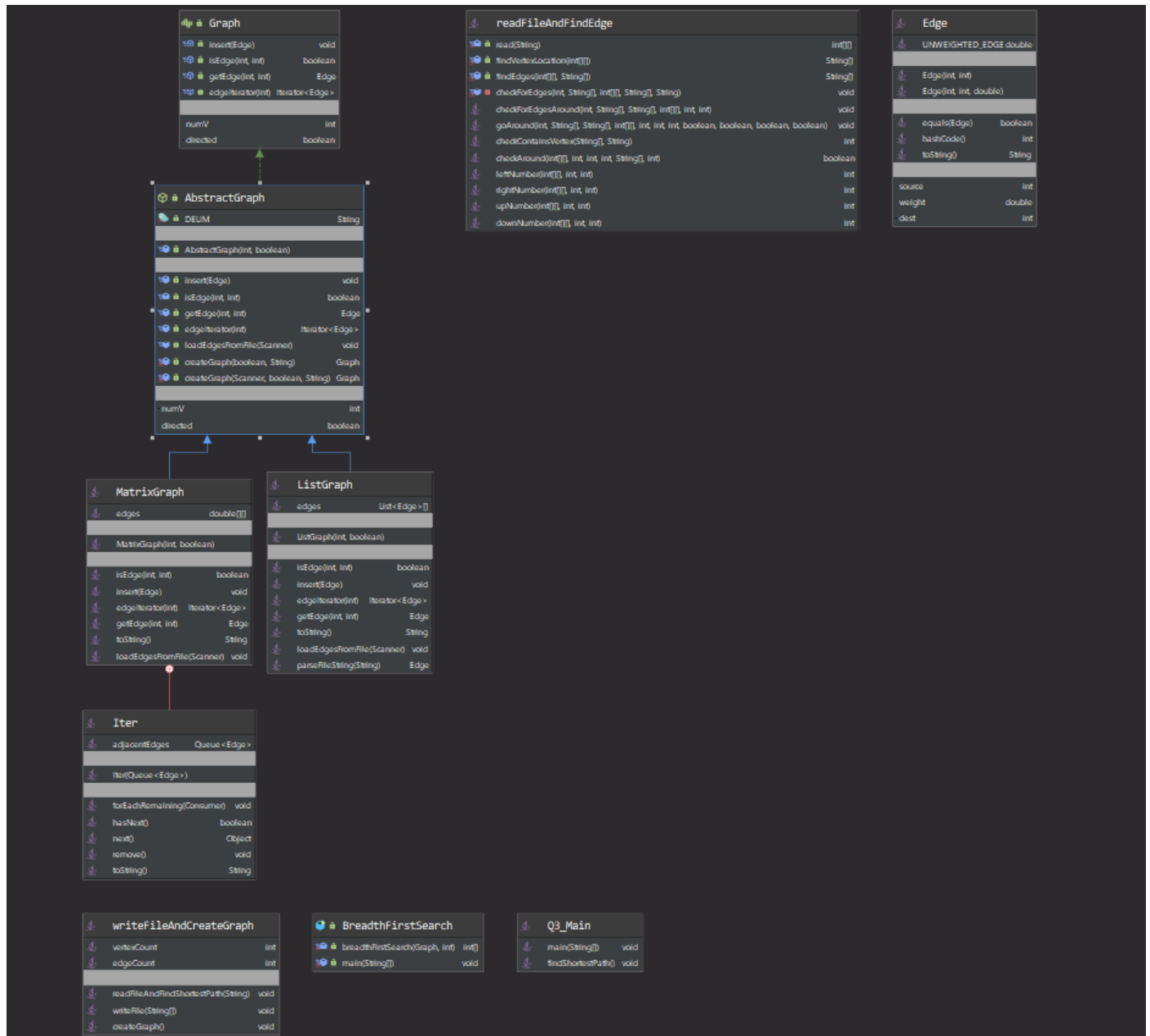# CSE 222/505 - Spring 2020
Homework #8 Report

Harun ALBAYRAK
171044014

1 – Class diagrams

Graph
- insert(Edge) : void
- isEdge(int, int) : boolean
- getEdge(int, int) : Edge
- edgeIterator(int) : Iterator<Edge>

- numV : int
- directed : boolean

AbstractGraph
- DELIM : String

- AbstractGraph(int, boolean)

- insert(Edge) : void
- isEdge(int, int) : boolean
- getEdge(int, int) : Edge
- edgeIterator(int) : Iterator<Edge>
- loadEdgesFromFile(Scanner) : void
- createGraph(boolean, String) : Graph
- createGraph(Scanner, boolean, String) : Graph

- numV : int
- directed : boolean

MatrixGraph
- edges : double[][]

- MatrixGraph(int, boolean)

- isEdge(int, int) : boolean
- insert(Edge) : void
- edgeIterator(int) : Iterator<Edge>
- getEdge(int, int) : Edge
- toString() : String
- loadEdgesFromFile(Scanner) : void

ListGraph
- edges : List<Edge>[]

- ListGraph(int, boolean)

- isEdge(int, int) : boolean
- insert(Edge) : void
- edgeIterator(int) : Iterator<Edge>
- getEdge(int, int) : Edge
- toString() : String
- loadEdgesFromFile(Scanner) : void
- parseFileString(String) : Edge

Iter
- adjacentEdges : Queue<Edge>

- Iter(Queue<Edge>)

- forEachRemaining(Consumer) : void
- hasNext() : boolean
- next() : Object
- remove() : void
- toString() : String

readFileAndFindEdge
- read(String) : int[][]
- findVertexLocation(int[][]) : String[]
- findEdges(int[][], String[]) : String[]
- checkForEdges(int, String[], int[][], String[], String) : void
- checkForEdgesAround(int, String[], String[], int[][], int, int) : void
- goAround(int, String[], String[], int[][], int, int, int, boolean, boolean, boolean, boolean) : void
- checkContainsVertex(String[], String) : int
- checkAround(int[][], int, int, int, String[], int) : boolean
- leftNumber(int[][], int, int) : int
- rightNumber(int[][], int, int) : int
- upNumber(int[][], int, int) : int
- downNumber(int[][], int, int) : int

Edge
- UNWEIGHTED_EDGE : double

- Edge(int, int)
- Edge(int, int, double)

- equals(Edge) : boolean
- hashCode() : int
- toString() : String

- source : int
- weight : double
- dest : int

writeFileAndCreateGraph
- vertexCount : int
- edgeCount : int

- readFileAndFindShortestPath(String) : void
- writeFile(String[]) : void
- createGraph() : void

BreadthFirstSearch
- breadthFirstSearch(Graph, int) : int[]
- main(String[]) : void

Q3_Main
- main(String[]) : void
- findShortestPath() : void

2 – Problem solution approach

Q3) First, I read the table from the file. And then i found vertices and edges. And i wrote this edges and theirs weight onto the file that its name output.txt. Finally Thanks to the book implementation(breadth-first search and Listgraph), i created ListGraph and i found the shortest path.

3 – Test cases

Q3)

Test Scenario	Expected Results	Actual Results	Pass/Fail
Find the shortest path	Finds correctly	It has found correctly	Pass

4 – Running command and results

```
Please enter the name of the file you want to read the graph:
asd.txt
The shortest path is:
1
2
4
12
```