

**GTU Department of Computer Engineering**

**CSE 222/505 – Spring 2020**

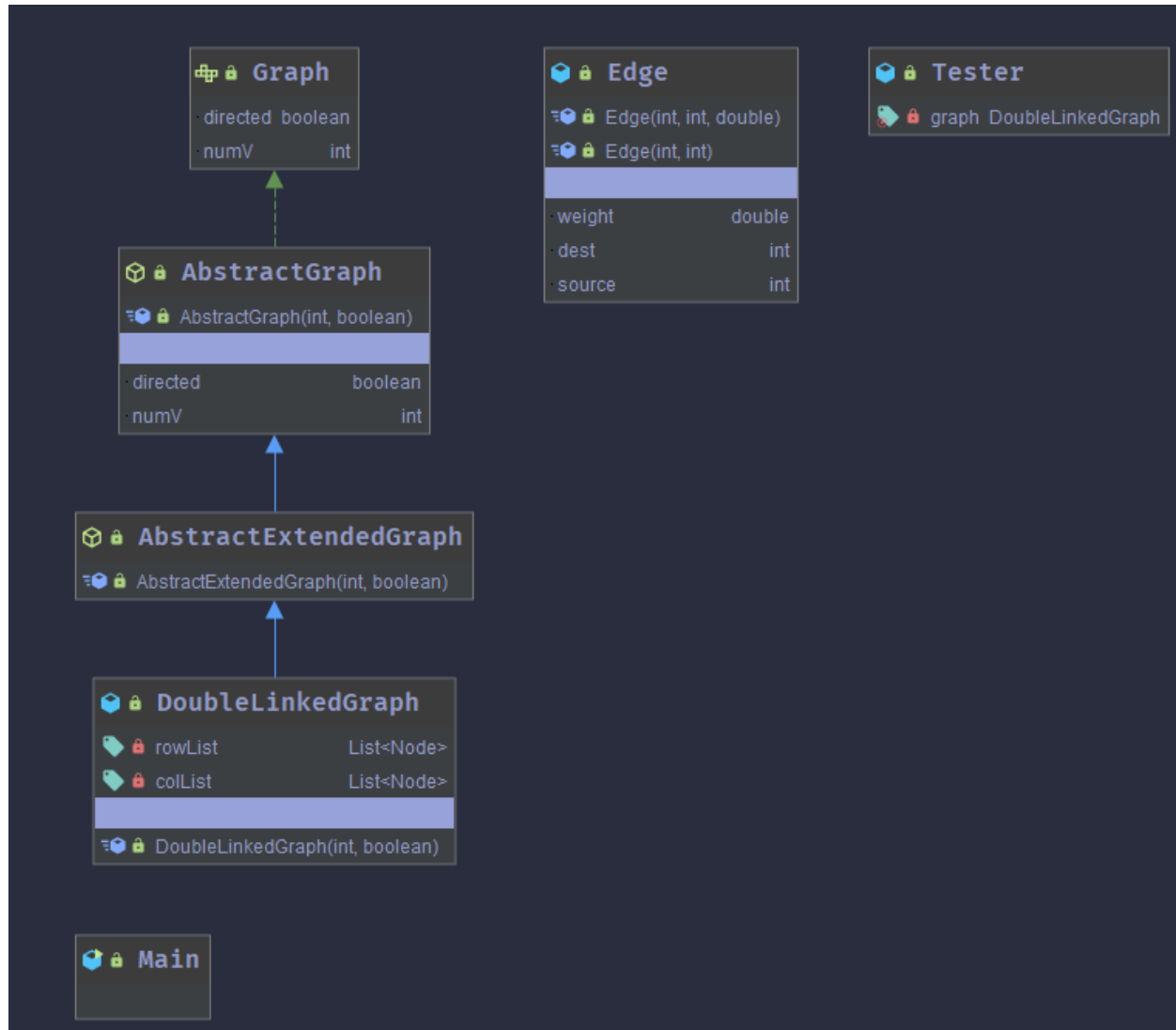
**Homework 8**

**Buğra Eren Yılmaz**

**1801042669**

**Q2**

# 1. Class Diagrams



## 2. Problem Solution Approach

The problem at hand required me to extend the functionality of graph ADT from book.

To achieve that I extended the AbstractGraph ADT from book and added those needed functionalities as methods, such as inserting and deleting vertices, deleting edges and that kind of stuff.

Sadly I couldn't achieve the deletion operations, sorry about that.

The current version of implementation can only insert vertices edges and verify they are inserted correctly and get their edge structures. The deletion of a vertex or edge is not implemented.

I created the class DoubleLinkedGraph which is a concrete implementation of AbstractExtendedGraph. It represents the graph as two linkedlists of row and column. Initially these lists has every vertex in them.

For adding a edge to this list, I traverse from the corresponding row list and find the head for row, then do the same thing for col list and find the head for col. Then from these heads I traverse the 2D Linked structure and find the corresponding positions for Nodes.

The node classes hold an edge as a data type and has 4 links for 4 directions.

To demonstrate that the graph is working fine I have included various unit tests below.

### 3. Tests

```
private static void t_insertTest() {
    System.out.println("Inserting 0,4 - 0,2 - 1,2 edges");
    graph.insert(new Edge(0, 4));
    graph.insert(new Edge(0, 2));
    graph.insert(new Edge(1, 2));
}

private static void t_isEdgeTest() {
    System.out.println("isEdge 0,4 - 0,2 - 1,2 - 5,0 edges");
    System.out.println("isEdge 0,4:" + graph.isEdge(0, 4));
    System.out.println("isEdge 0,2:" + graph.isEdge(0, 2));
    System.out.println("isEdge 1,2:" + graph.isEdge(1, 2));
    System.out.println("isEdge 5,0:" + graph.isEdge(5, 0));
    System.out.println("isEdge 5,14:" + graph.isEdge(5, 14));
}

private static void t_getEdgeTest() {
    Edge edge = graph.getEdge(0, 4);
    System.out.println("Got the edge (0,4): " + edge);
}

private static void t_addVertex() {
    graph.insertVertex(14);
    System.out.println("Added new vertex: 14");
}

private static void t_addEdgeToNewVertex() {
    graph.insert(new Edge(5, 14));
    System.out.println("Added new edge to new vertex: 5,14");
}
```

## 4. Results

Inserting 0,4 - 0,2 - 1,2 edges

isEdge 0,4 - 0,2 - 1,2 - 5,0 edges

isEdge 0,4:true

isEdge 0,2:true

isEdge 1,2:true

isEdge 5,0:false

isEdge 5,14:false

Got the edge (0,4): Edge[dest=4, source=4, weight=1.000000]

Added new vertex: 14

Added new edge to new vertex: 5,14

isEdge 0,4 - 0,2 - 1,2 - 5,0 edges

isEdge 0,4:true

isEdge 0,2:true

isEdge 1,2:true

isEdge 5,0:false

isEdge 5,14:true