

**Gebze Technical University  
Computer Engineering**

**CSE 222 - 2018 Spring**

**HOMEWORK X REPORT**

**EDA BAHRIOGLU  
131044055**

Course Assistant:

# INTRODUCTION

## 1.1 Problem Definition

The aim of this study is to create a program with linked list structure by using OOP principles. This program contains the applications of a machine learning experiment. The connections between the experiments are provided with days. There are more than one day and experiment. These experiments can be grouped so that there are more than one experiment in the same day. The connection of the experiments using various data, such as start and end time, is provided with nodes. This structure provides great convenience unlike arraylist in the event of any insertion event..

## 1.2 System Requirements

The system consists of a simple linked list structure that controls the insertion and subtraction nodes of the connection nodes between the experiments. We have 5 data to be used for the connections first..

Setup: string is a type. Try to tell us the setup status.

Day: is a int type . The links between the dates should be determined according to the days.

Time : is is a string type. show the start time.

Complete : is a boolean type. give information about the completion status of the experiment.

Accuracy: is a type float It make control valid of experiments.

There must be two classes as Experiment and ExperimentaList which we have to have for system structure.Experiment is used as the type of data and the list of data is stored in the above .ExperimentList is an iterable classt from the iterable interface where all the operations between the tests are done. It should contain the iterator method.It must be perform the following tasks.

1. addExp: should be able to add the test node to the end of the day.
2. removeExp: should deleted an experiment from the given index.
3. removeDay: delete all experiments from the given day.
- 4.getExp: You should take an experiment from the given position and index

5.setExp: change an experiment taken from parameter

6.listExp: List all experiments and display them on the screen

7.orderDay: Sort the experiments by their accuracy, but the list should not change

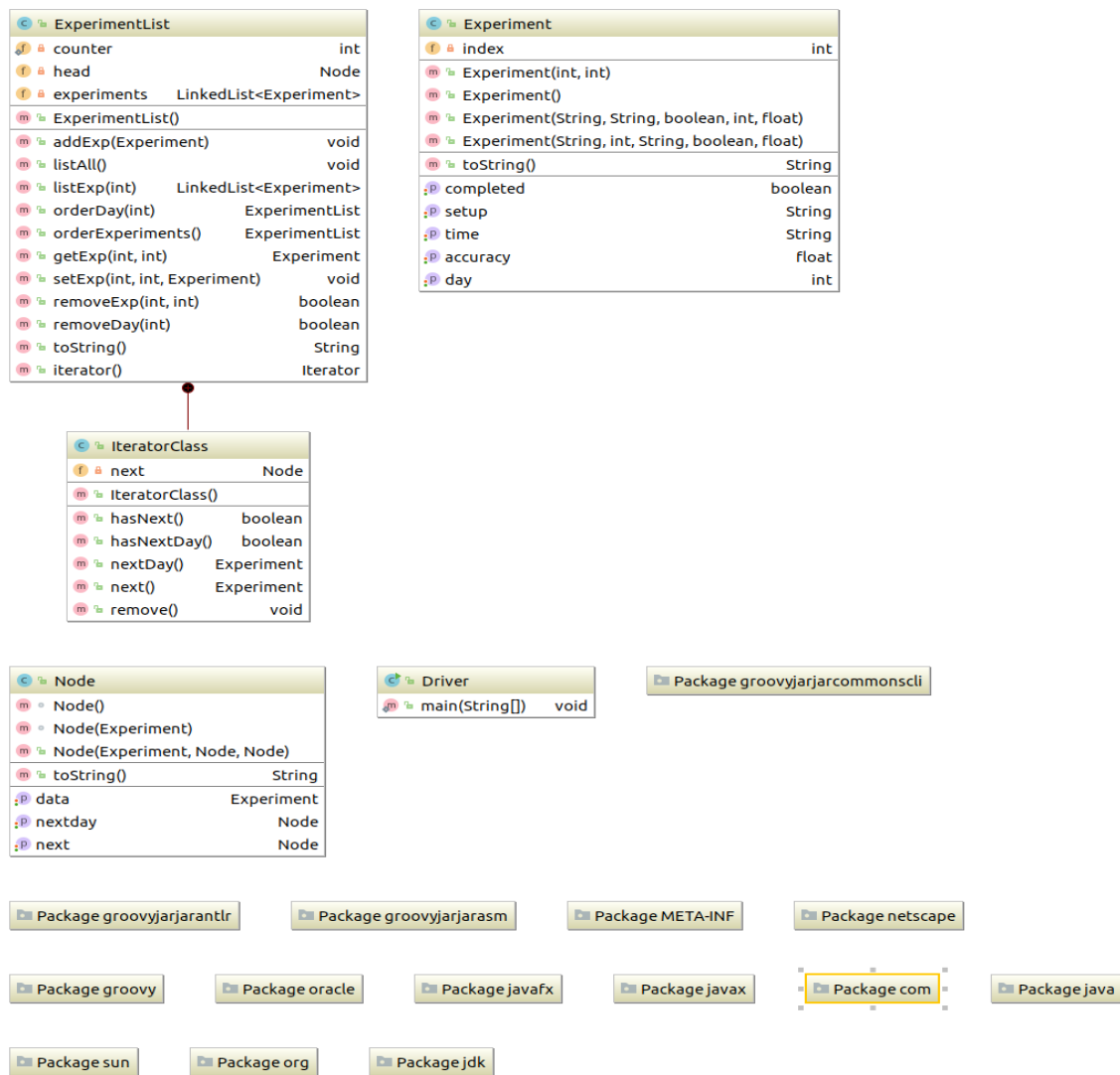
8.orderExperimets: Sort the tests according to their accuracy, but the list should change.

In addition, the program for our own Node class I and Iterator class is available. Iterable class is called with the iterator method.

These methods should be testable for the entire list in Driver class

## METHOD

### 2.1 Class Diagrams



## 2.2 Use Case Diagrams

Add use case diagrams if required.

## 2.3 Other Diagrams (optional)

I have not another diagrams.

## 2.4 Problem Solution Approach

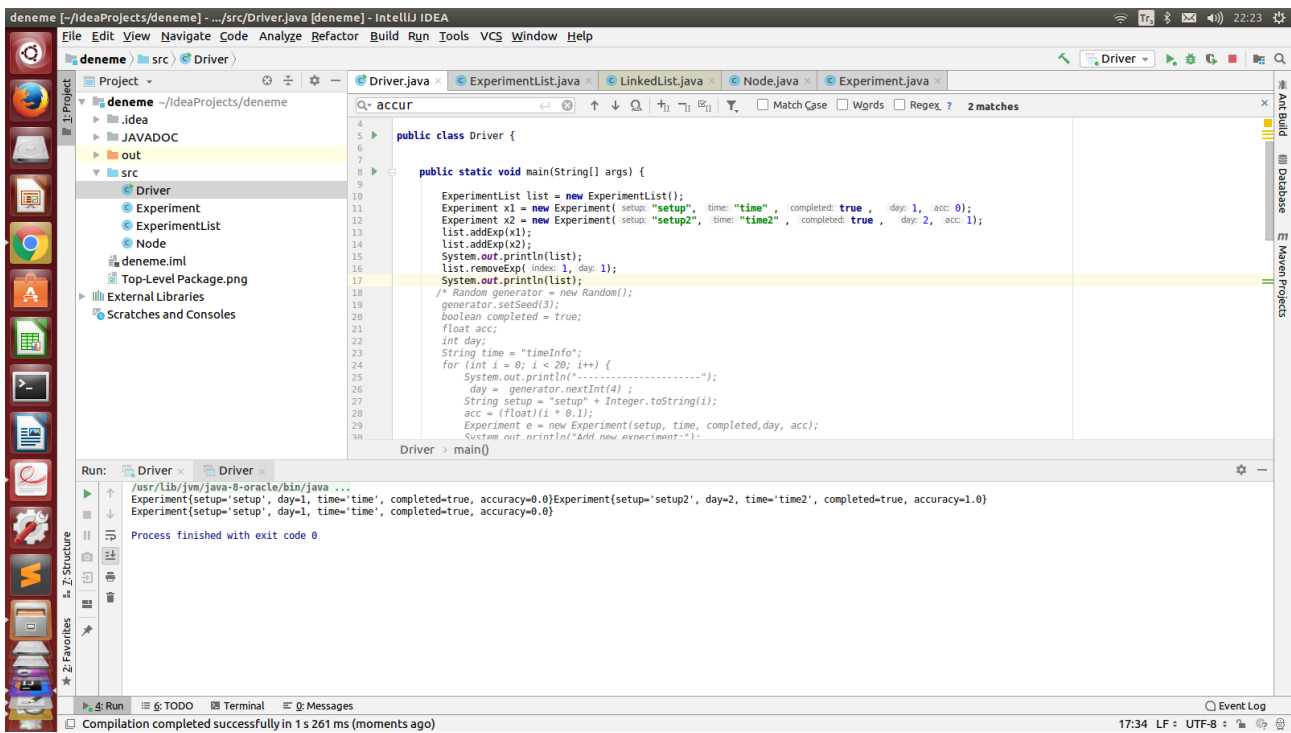
I tried not to go beyond the object oriented principles in problem solving. In order to prevent complexity, I applied all the necessary classes and tried not to use inner class.

# RESULT

## 3.1 Test Cases

## 3.2 Running Results

Basically tests add and remove node.



The screenshot displays the IntelliJ IDEA IDE interface. The main editor window shows the `Driver.java` file with the following code:

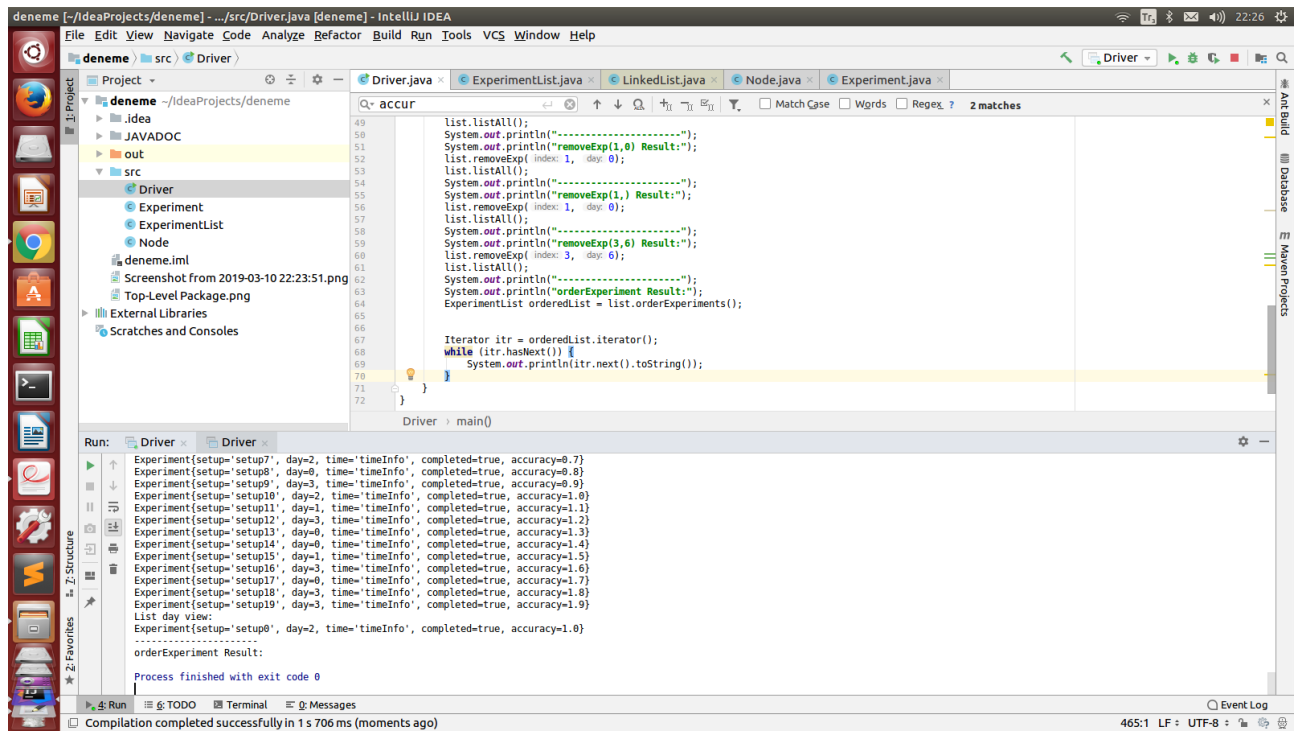
```
public class Driver {  
    public static void main(String[] args) {  
        ExperimentList list = new ExperimentList();  
        Experiment x1 = new Experiment( setup: "setup", time: "time", completed: true, day: 1, acc: 0);  
        Experiment x2 = new Experiment( setup: "setup2", time: "time2", completed: true, day: 2, acc: 1);  
        list.addExp(x1);  
        list.addExp(x2);  
        System.out.println(list);  
        list.removeExp( index: 1, day: 1);  
        System.out.println(list);  
        /* Random generator = new Random();  
        generator.setSeed(3);  
        boolean completed = true;  
        float acc;  
        int day;  
        String time = "timeInfo";  
        for (int i = 0; i < 20; i++) {  
            System.out.println(".....");  
            day = generator.nextInt(4);  
            String setup = "setup" + Integer.toString(i);  
            acc = (float)(i * 0.1);  
            Experiment e = new Experiment(setup, time, completed, day, acc);  
            System.out.println("Add new experiment:");  
        }  
    }  
}
```

The Run window at the bottom shows the output of the program:

```
/usr/lib/jvm/java-8-oracle/bin/java ...  
Experiment[setup='setup', day=1, time='time', completed=true, accuracy=0.0]Experiment[setup='setup2', day=2, time='time2', completed=true, accuracy=1.0]  
Experiment[setup='setup', day=1, time='time', completed=true, accuracy=0.0]  
Process finished with exit code 0
```

The status bar at the bottom indicates "Compilation completed successfully in 1 s 261 ms (moments ago)" and the time is 17:34.

Your test is following:



```
deneme [~/IdeaProjects/deneme] - .../src/Driver.java [deneme] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Project - deneme
  deneme
  .idea
  JAVADOC
  out
  src
    Driver
    Experiment
    ExperimentList
    Node
  deneme.iml
  Screenshot from 2019-03-10 22:23:51.png
  Top-Level Package.png
  External Libraries
  Scratches and Consoles

Driver.java
49 list.listAll();
50 System.out.println("-----");
51 System.out.println("removeExp(1,0) Result:");
52 list.removeExp(index: 1, day: 0);
53 list.listAll();
54 System.out.println("-----");
55 System.out.println("removeExp(1,1) Result:");
56 list.removeExp(index: 1, day: 0);
57 list.listAll();
58 System.out.println("-----");
59 System.out.println("removeExp(3,6) Result:");
60 list.removeExp(index: 3, day: 6);
61 list.listAll();
62 System.out.println("-----");
63 System.out.println("orderExperiment Result:");
64 ExperimentList orderedList = list.orderExperiments();
65
66
67 Iterator itr = orderedList.iterator();
68 while (itr.hasNext()) {
69     System.out.println(itr.next().toString());
70 }
71
72 }

Run: Driver
Experiment{setup='setup7', day=2, time='timeInfo', completed=true, accuracy=0.7}
Experiment{setup='setup8', day=0, time='timeInfo', completed=true, accuracy=0.8}
Experiment{setup='setup9', day=3, time='timeInfo', completed=true, accuracy=0.9}
Experiment{setup='setup10', day=2, time='timeInfo', completed=true, accuracy=1.0}
Experiment{setup='setup11', day=1, time='timeInfo', completed=true, accuracy=1.1}
Experiment{setup='setup12', day=3, time='timeInfo', completed=true, accuracy=1.2}
Experiment{setup='setup13', day=0, time='timeInfo', completed=true, accuracy=1.3}
Experiment{setup='setup14', day=0, time='timeInfo', completed=true, accuracy=1.4}
Experiment{setup='setup15', day=1, time='timeInfo', completed=true, accuracy=1.5}
Experiment{setup='setup16', day=3, time='timeInfo', completed=true, accuracy=1.6}
Experiment{setup='setup17', day=0, time='timeInfo', completed=true, accuracy=1.7}
Experiment{setup='setup18', day=3, time='timeInfo', completed=true, accuracy=1.8}
Experiment{setup='setup19', day=3, time='timeInfo', completed=true, accuracy=1.9}
List day view:
Experiment{setup='setup0', day=2, time='timeInfo', completed=true, accuracy=1.0}
-----
orderExperiment Result:
Process finished with exit code 0

Compilation completed successfully in 1 s 706 ms (moments ago)
465:1 LF : UTF-8 : 22:26
```