Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 2 REPORT

ÖMER ÇEVİK
161044004

Course Assistant: Ayşe Şerbetçi Turan

1 INTRODUCTION

1.1 Problem Definition

This project is about saving machine learning experiments and experiments's results per a day.

- In that work we use single linked list data structure and the problems are:

- Adding nodes in linked list causes to following the links to maintain.
- Removing nodes in linked list causes to following the links to maintain.
- Sorting nodes in linked list causes to following the links to maintain.

It is going to be explained in detailed in other parts.

1.2 System Requirements

In system requirements, firstly I used single linked list data structure. That linked list has two way link that one of them keeps nextDay, other one keeps next.

In that project, with using linked list structure I created two classes: Experiment and ExperimentList. ExperimentList classes has a head of linked list structure of Experiment.

That head of linked list has given eight functionality. These functionalities are implemented as methods as **addExp**, **getExp**, **setExp**, **removeExp**, **listExp**, **removeDay**, **orderDay** and **orderExperiments**. This methods use instance variables that declared in Experiment class setup, day, time, completed and accuracy. In Experiment class for to moving to other link I declared instance variable named **next** that keeps next of experiment information.

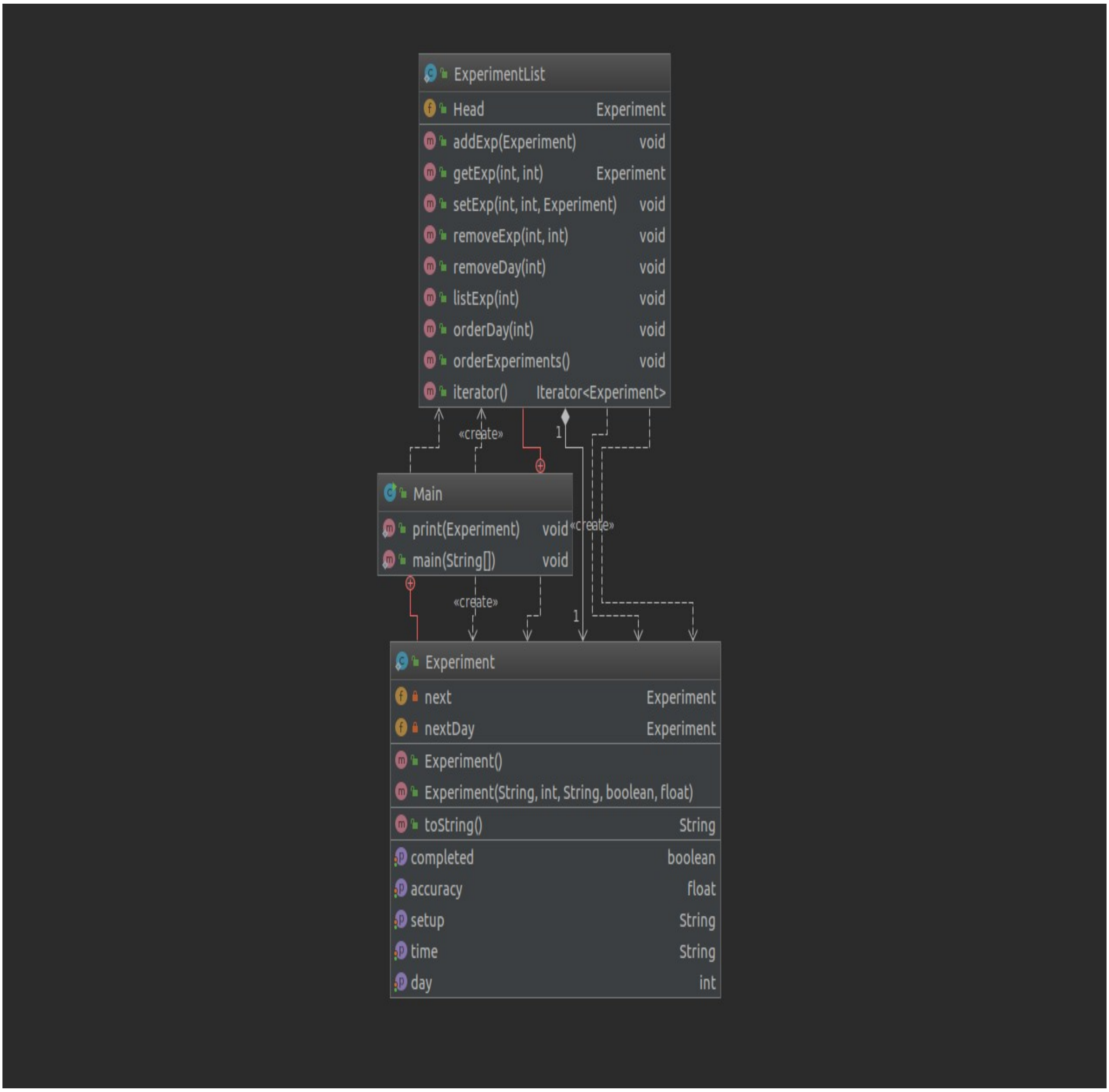
In Experiment class for moving to other day link I declared instance variable named **nextDay** that keeps next day of experiment information. In ExperimentList class **addExp**, **getExp**, **setExp**, **removeExp**, **listExp**, **removeDay**, **orderDay** and **orderExperiments** methods have been implemented. Also ExperimentList class implements **Iterable** interface to override the **iterator** and its methods that **next()** and **hasNext().** I override the **toString()** method of Object class to print experiment object's instance variables together. I also write a function named **print()** to print each experiment in the list.

- User can add a new experiment to linked list.
- User can remove an experiment from linked list.
- User can set an experiment to old one in linked list.
- User can get an experiment in linked list.
- User can list all completed experiments in given day from linked list.
- User can sort experiments using accuracy in given day in linked list.

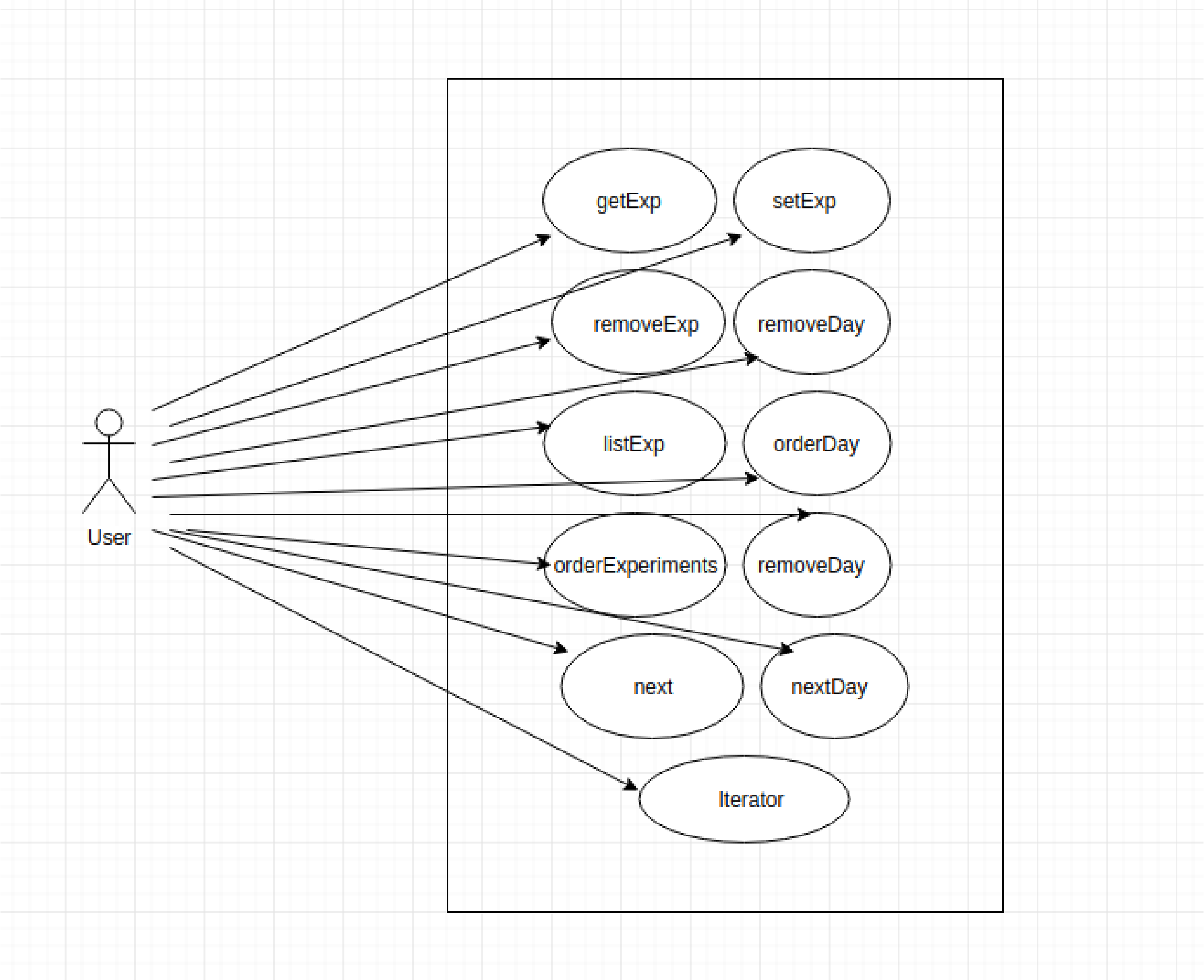
- User can sort and doesn’t damage experiment list using accuracy.
- User can use the iterator to move in linked list.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams



2.3 Other Diagrams (optional)

2.4 Problem Solution Approach

Firstly I need to inform that there is no missing part.

I used single linked list structure to remove and add new experiments into my list. When I add a new node to my link list it is great to adding without shifting my whole list. But there are some unefficient things that to access to an experiment is hard. Because we have to move on all experiments using next node.

There are some other problems when we remove or set an experiment we have to check if previous node or nodes references our removable node. To solve that problem I used temp objects to save previous and next nodes. After my work I reassign to my new node or removed nodes.

When we try to sort in a day of linked list I used array to save the all experiments variables in given day and I sorted the array then I created a new linked list and added to ExperimentList linked list.

Time Complexity

1-) addExp()

In that method I used too many instructions but there are two while loops nested that causes n^2 so $T(n) = O(n^2)$.

2-) getExp()

In that method I used too many instructions but there is only one while loop that means n so $T(n) = O(n)$.

3-) setExp()

In that method I used too many instructions but there is only one while loop that means n so $T(n) = O(n)$.

4-) removeExp()

In that method I used too many instructions but there are two while loops nested that causes n^2 so $T(n) = O(n^2)$.

5-) removeDay()

In that method I used too many instructions but there are two while loops nested that causes n^2 so $T(n) = O(n^2)$.

6-) listExp()

In that method I used too many instructions but there is only one while loop that means n so $T(n) = O(n)$.

7-) orderDay()

In that method I used too many instructions but there are two for loops nested that causes n^2 so $T(n) = O(n^2)$.

8-) orderExperiments()

In that method I used too many instructions but there are two for loops nested that causes n^2 so $T(n) = O(n^2)$.

9-) hasNext() && next()

In that method I used too many instructions but there is no loop that means hasNext() and next() methods each one have $T(n) = O(1)$.

All getters and setters are $T(n) = O(1)$, print() method has one while loop so $T(n) = O(n)$.

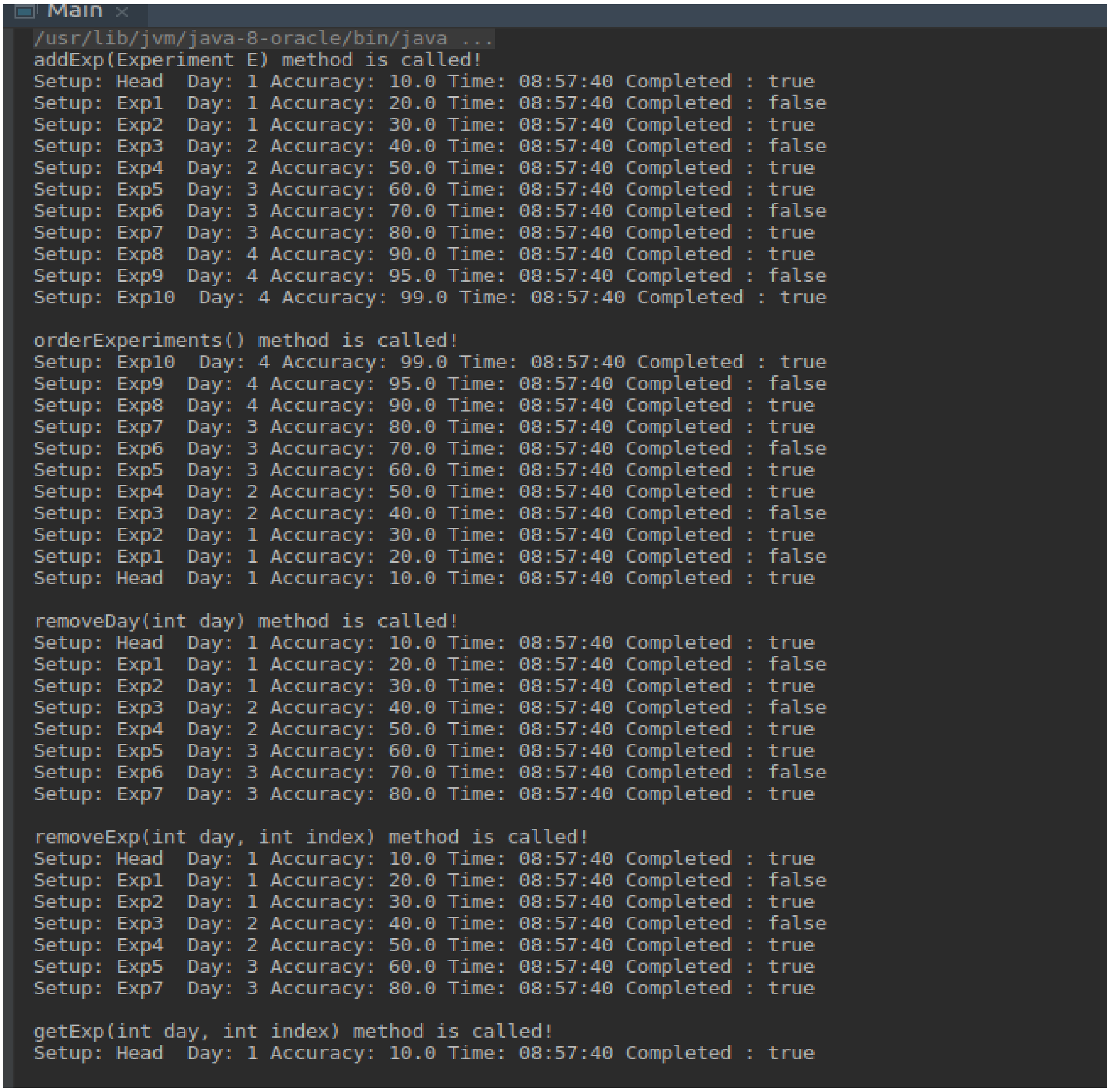
3 RESULT

3.1 Test Cases

Main Test

- I used main to explain what is going on in our programme. All methods are working well. Results of that project are in 3.2.

3.2 Running Results



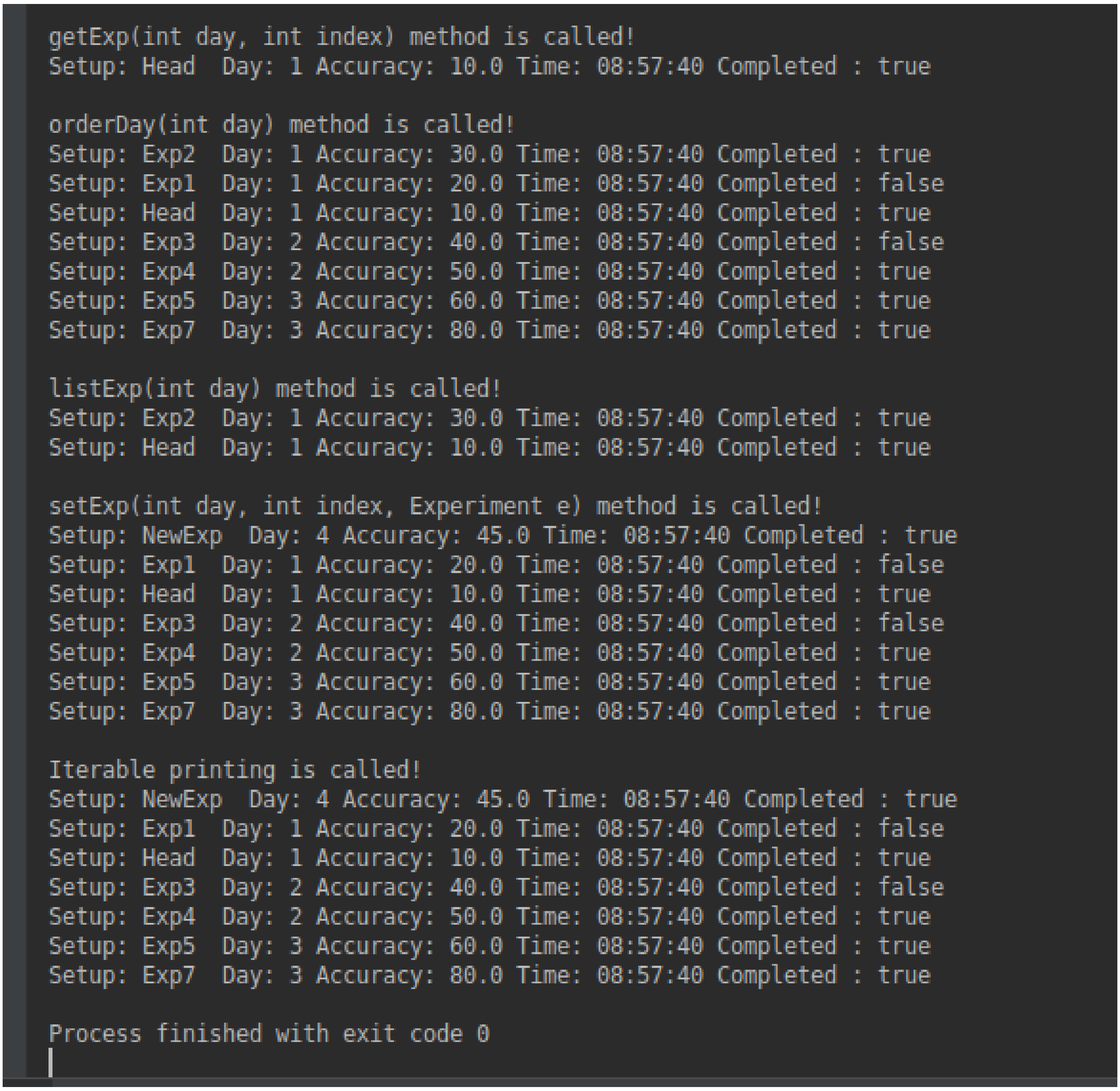
```
/usr/lib/jvm/java-8-oracle/bin/java ...
addExp(Experiment E) method is called!
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp6  Day: 3 Accuracy: 70.0 Time: 08:57:40 Completed : false
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true
Setup: Exp8  Day: 4 Accuracy: 90.0 Time: 08:57:40 Completed : true
Setup: Exp9  Day: 4 Accuracy: 95.0 Time: 08:57:40 Completed : false
Setup: Exp10  Day: 4 Accuracy: 99.0 Time: 08:57:40 Completed : true

orderExperiments() method is called!
Setup: Exp10  Day: 4 Accuracy: 99.0 Time: 08:57:40 Completed : true
Setup: Exp9  Day: 4 Accuracy: 95.0 Time: 08:57:40 Completed : false
Setup: Exp8  Day: 4 Accuracy: 90.0 Time: 08:57:40 Completed : true
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true
Setup: Exp6  Day: 3 Accuracy: 70.0 Time: 08:57:40 Completed : false
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true

removeDay(int day) method is called!
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp6  Day: 3 Accuracy: 70.0 Time: 08:57:40 Completed : false
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true

removeExp(int day, int index) method is called!
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true

getExp(int day, int index) method is called!
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
```



```
getExp(int day, int index) method is called!
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true

orderDay(int day) method is called!
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true

listExp(int day) method is called!
Setup: Exp2  Day: 1 Accuracy: 30.0 Time: 08:57:40 Completed : true
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true

setExp(int day, int index, Experiment e) method is called!
Setup: NewExp  Day: 4 Accuracy: 45.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true

Iterable printing is called!
Setup: NewExp  Day: 4 Accuracy: 45.0 Time: 08:57:40 Completed : true
Setup: Exp1  Day: 1 Accuracy: 20.0 Time: 08:57:40 Completed : false
Setup: Head  Day: 1 Accuracy: 10.0 Time: 08:57:40 Completed : true
Setup: Exp3  Day: 2 Accuracy: 40.0 Time: 08:57:40 Completed : false
Setup: Exp4  Day: 2 Accuracy: 50.0 Time: 08:57:40 Completed : true
Setup: Exp5  Day: 3 Accuracy: 60.0 Time: 08:57:40 Completed : true
Setup: Exp7  Day: 3 Accuracy: 80.0 Time: 08:57:40 Completed : true

Process finished with exit code 0
```

In that results,

- addExp is used to create the linked list that includes 4 days totally 10 experiments.
- orderExperiments is used to sort the all experiments using the accuracy and it didn’t damage the linked list with changing.
- removeDay is used to remove 4th day successfully.
- removeExp is used to remove 3rd day and 2nd experiment (Indexes start with 1).
- getExp is used to get experiment in 1st day 1st index (Indexes start with 1).
- orderDay is used to sort given day using accuracy that choosed 1st day.
- listExp is used to list all completed experiments in linked list in given day.
- setExp is used to set a new experiment to given day and index that choosed 1st day and 1st index.
- The last print is about using iterator for to printing.