

**Gebze Technical University
Computer Engineering**

CSE 222 - 2019 Spring

HOMEWORK 2 REPORT

**STUDENT NAME: AHMET MELIH YANALAK
STUDENT NUMBER: 151044044**

Course Assistant: AYSE SERBETCI TURAN

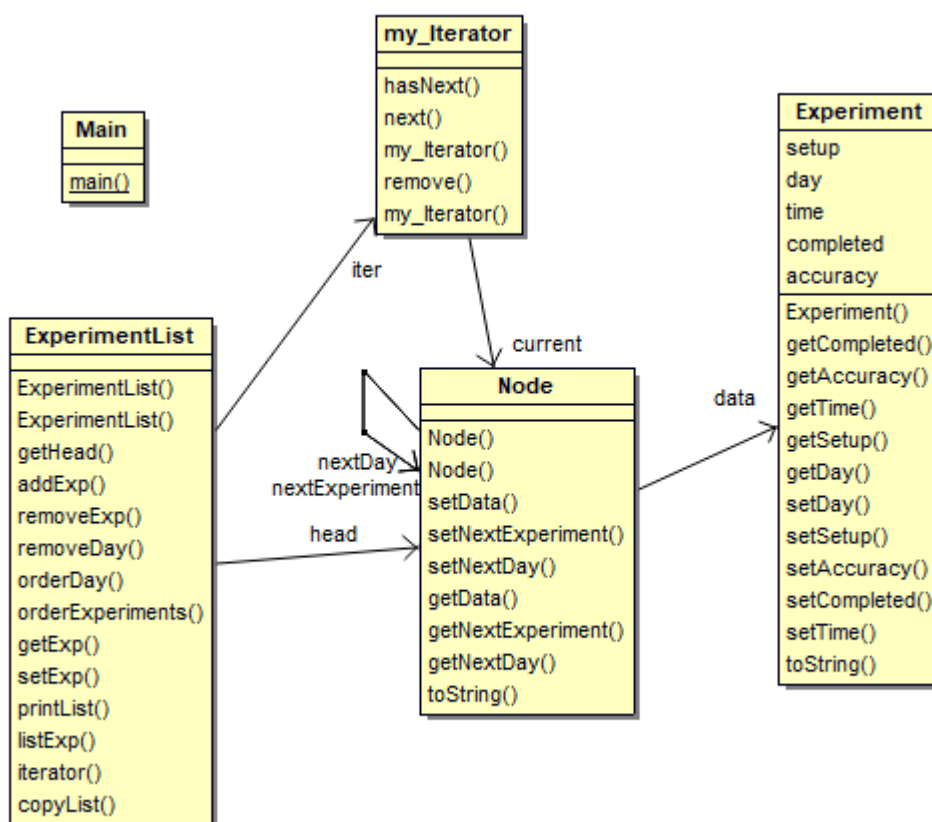
1 INTRODUCTION

1.1 Problem Definition

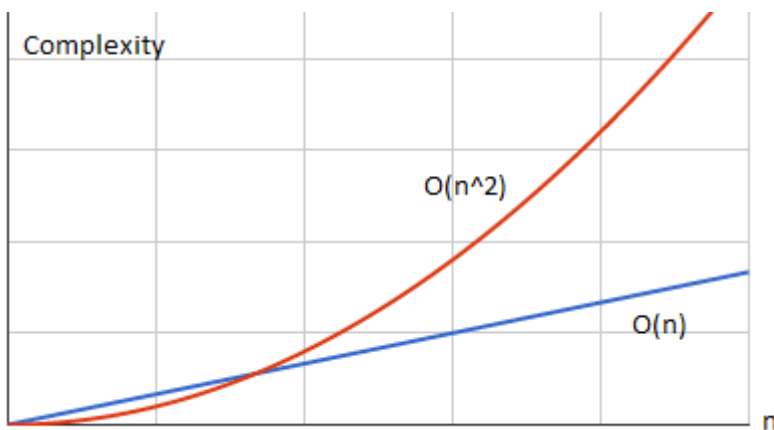
Due to constant size of arrays and collections that use underlying arrays , making changes on collections is quiet inefficient. There must be a new collection created and the original collection must be copied to there which is really unefficient in terms of memory usage and complexity. Also in order to reach days , each element of collection must be visited until specified data is found which is a slow technique to deal with.

2 METHOD

2.1 Class Diagrams



2.2 Other Diagrams



Complexities of methods in ExperimentList Class

addExp(): This method scans the list from head till find the place that experiment will be added on. There is a loop which iterates that's why the complexity is $O(n)$.

Remove(): This method's complexity is equal to $O(n)$ because of the same loop to find the place.

RemoveDay(): Complexity of this method is $O(n^2)$ because it calls the remove method in a while loop which indicates all experiments in that day.

OrderDay(): Complexity of this method is $O(n^2)$ because it uses the algorithm bubble sort.

OrderExp(): Complexity of OrderExp is the same as OrderDay; it also does bubble sort. $O(n^2)$

GetExp(): This is a getter method so it starts with head and tries to find the index that will be returned in a loop. That's why its complexity is $O(n)$.

SetExp(): This method also indicates experiments till it finds its place to set. $O(n)$

ListExp(): This method finds the day and iterates it while printing. $O(n)$

2.3 Problem Solution Approach

A programmer should always care about the usage of computer memory and speed of a program. In this Project, in order to avoid extra data copying, a single linked list structure has been created and also to make functions more efficient, there is a 2 linked list created. In more details, if the user wants to add an experiment into the middle of the list, there will be no data copying necessary. The program only has to change the links of 2 nodes that the new experiment will be added between them. Let's assume that the user wants to add an experiment which will begin on 7th day and there are 100 experiments before 7th day. If a collection is used in this program, it would be slower to reach 7th day before adding but with a second list, it's much faster.

3 RESULT

3.1 Test Cases

AddExp method

Case 1 -> when head is null

Case 2 -> when add to the head

Case 3 -> when add to the end of the list

Case 4 -> when add to the middle of the list

RemoveExp method

Case 1 -> removing from head

Case 2 -> removing from end

Case 3 -> removing from middle

RemoveDay method

Case 1 -> removing first day

Case 2 -> removing day in the middle

Case 3 -> removing last day

listExp method

Case 1 -> listing first day

Case 2 -> listing last day

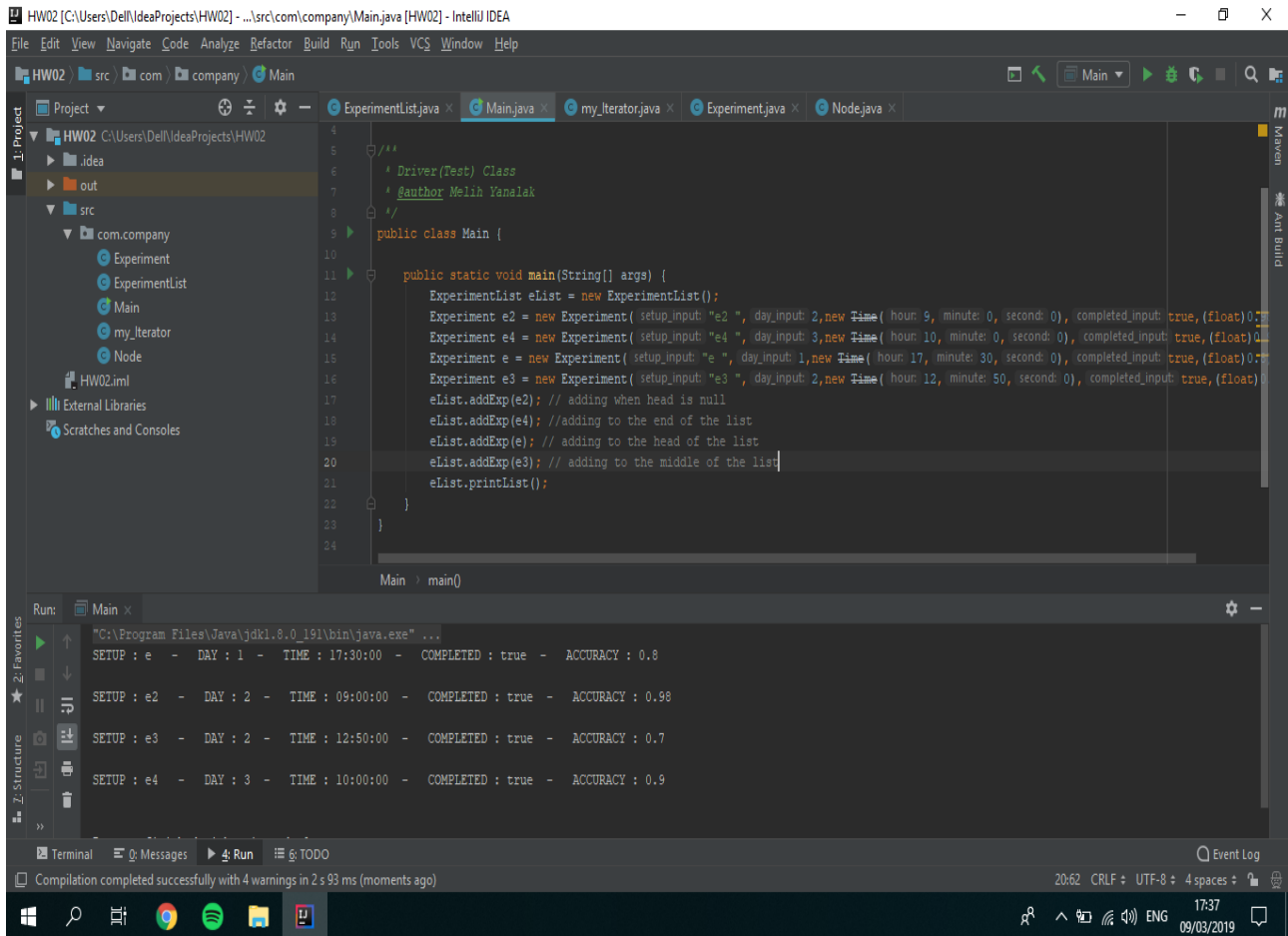
Case 3 -> listing day in the middle

orderDay,orderExperiments,getExp,setExp methods

OUTPUTS ARE BELOW

3.2 Running Results

Add method with all cases



```
HW02 [C:\Users\Del\IdeaProjects\HW02] - ...src\com\company\Main.java [HW02] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HW02 / src / com / company / Main
Project HW02 C:\Users\Del\IdeaProjects\HW02
  .idea
  out
  src
    com.company
      Experiment
      ExperimentList
      Main
      my_iterator
      Node
  HW02.iml
  External Libraries
  Scratches and Consoles
  Maven
  Ant Build
  Run: Main x
  Run: "C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
  SETUP : e - DAY : 1 - TIME : 17:30:00 - COMPLETED : true - ACCURACY : 0.8
  SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98
  SETUP : e3 - DAY : 2 - TIME : 12:50:00 - COMPLETED : true - ACCURACY : 0.7
  SETUP : e4 - DAY : 3 - TIME : 10:00:00 - COMPLETED : true - ACCURACY : 0.9
  Terminal Messages Run TODO
  Compilation completed successfully with 4 warnings in 2 s 93 ms (moments ago)
  20:62 CRLF UTF-8 4 spaces
  17:37 09/03/2019
```

```
/**
 * Driver(Test) Class
 * @author Melih Yamalak
 */
public class Main {

    public static void main(String[] args) {
        ExperimentList eList = new ExperimentList();
        Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2,new Time( hour: 9, minute: 0, second: 0), completed_input: true,(float)0.8);
        Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 3,new Time( hour: 10, minute: 0, second: 0), completed_input: true,(float)0.9);
        Experiment e = new Experiment( setup_input: "e ", day_input: 1,new Time( hour: 17, minute: 30, second: 0), completed_input: true,(float)0.7);
        Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2,new Time( hour: 12, minute: 50, second: 0), completed_input: true,(float)0.7);
        eList.addExp(e2); // adding when head is null
        eList.addExp(e4); //adding to the end of the list
        eList.addExp(e); // adding to the head of the list
        eList.addExp(e3); // adding to the middle of the list
        eList.printList();
    }
}
```

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
SETUP : e - DAY : 1 - TIME : 17:30:00 - COMPLETED : true - ACCURACY : 0.8
SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98
SETUP : e3 - DAY : 2 - TIME : 12:50:00 - COMPLETED : true - ACCURACY : 0.7
SETUP : e4 - DAY : 3 - TIME : 10:00:00 - COMPLETED : true - ACCURACY : 0.9
```

Terminal Messages Run TODO

Compilation completed successfully with 4 warnings in 2 s 93 ms (moments ago)

20:62 CRLF UTF-8 4 spaces

17:37 09/03/2019

RemoveExp method with all cases

The screenshot displays the IntelliJ IDEA IDE with a project named HW02. The main class, Main.java, contains the following code:

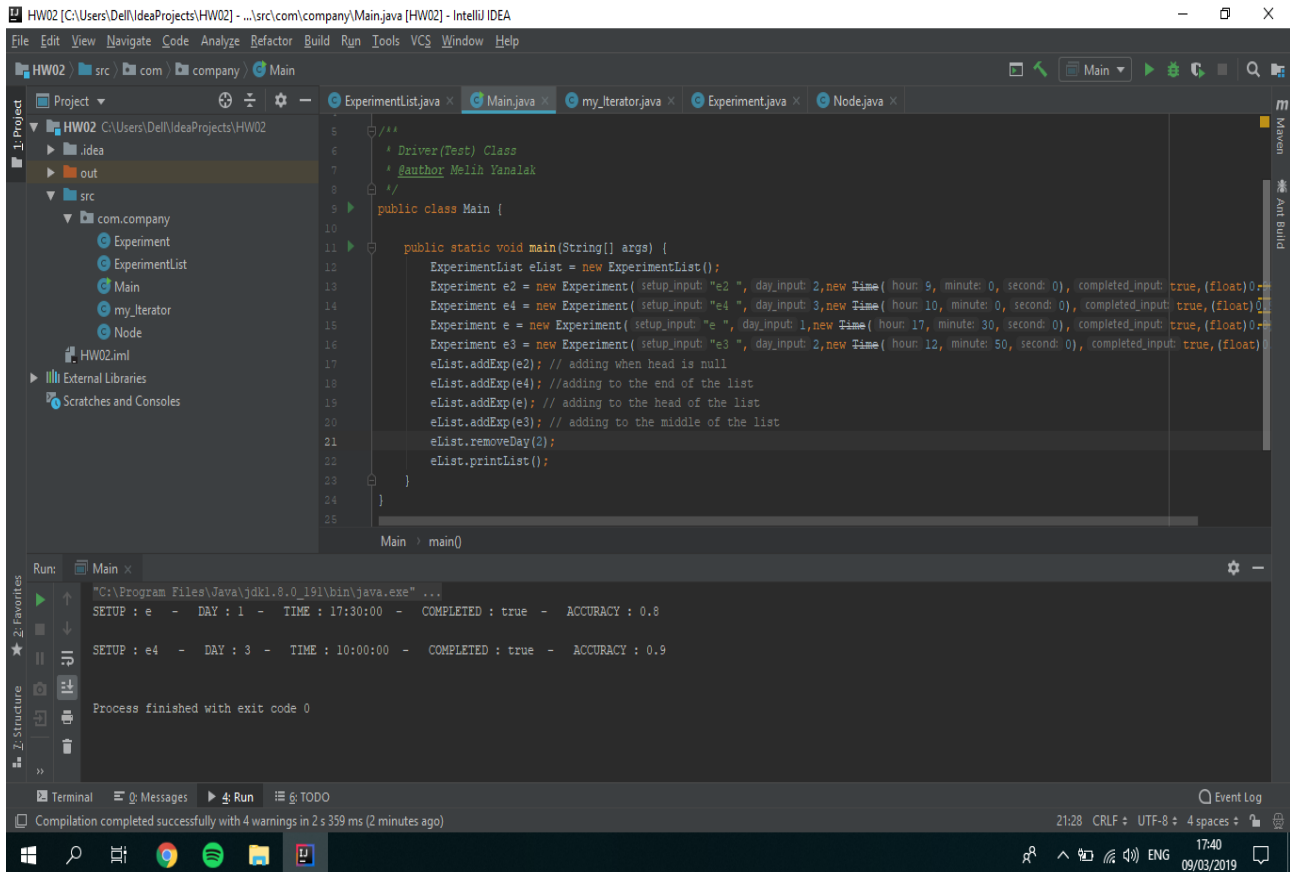
```
1  //
2  public class Main {
3
4      public static void main(String[] args) {
5          ExperimentList eList = new ExperimentList();
6          Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float) 0.98);
7          Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 3, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float) 0.98);
8          Experiment e = new Experiment( setup_input: "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float) 0.98);
9          Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float) 0.98);
10         eList.addExp(e2); // adding when head is null
11         eList.addExp(e4); // adding to the end of the list
12         eList.addExp(e); // adding to the head of the list
13         eList.addExp(e3); // adding to the middle of the list
14         eList.removeExp( day: 1, index: 0); // removing head
15         eList.removeExp( day: 2, index: 1); // removing from middle of the list
16         eList.removeExp( day: 3, index: 0); // removing last experiment in list
17         eList.printList();
18     }
19 }
20
21 Main > main()
```

The Run window shows the output of the program:

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98
Process finished with exit code 0
```

The status bar at the bottom indicates that the compilation was successful with 4 warnings in 2 s 374 ms (moments ago).

RemoveDay method



The screenshot displays the IntelliJ IDEA IDE with a project named HW02. The main editor shows the `Main.java` file, which contains a `Driver` class and a `Main` class. The `Main` class has a `main` method that creates an `ExperimentList` and adds three experiments. The `removeDay` method is called on the list.

```
1 // Driver (Test) Class
2 // Author Melih Yamalak
3 //
4
5 public class Main {
6
7     public static void main(String[] args) {
8         ExperimentList eList = new ExperimentList();
9         Experiment e2 = new Experiment( "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float) 0.8);
10        Experiment e4 = new Experiment( "e4 ", day_input: 3, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float) 0.9);
11        Experiment e = new Experiment( "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float) 0.8);
12        Experiment e3 = new Experiment( "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float) 0.8);
13        eList.addExp(e2); // adding when head is null
14        eList.addExp(e4); // adding to the end of the list
15        eList.addExp(e); // adding to the head of the list
16        eList.addExp(e3); // adding to the middle of the list
17        eList.removeDay(2);
18        eList.printList();
19    }
20 }
21
22 Main > main()
```

The Run window shows the output of the program:

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
SETUP : e - DAY : 1 - TIME : 17:30:00 - COMPLETED : true - ACCURACY : 0.8
SETUP : e4 - DAY : 3 - TIME : 10:00:00 - COMPLETED : true - ACCURACY : 0.9
Process finished with exit code 0
```

The status bar at the bottom indicates that the compilation was successful with 4 warnings in 2 s 359 ms (2 minutes ago).

OrderDay method

The screenshot displays the IntelliJ IDEA IDE with a Java project named 'HW02'. The main editor shows the 'Main.java' file, which contains a 'Main' class with a 'main' method. The 'main' method initializes an 'ExperimentList' object 'eList' and adds three 'Experiment' objects: 'e2', 'e4', and 'e3'. It then calls 'eList.orderDay(2)' to sort the experiments by day. The 'Run' output window shows the results of the execution, displaying the setup, day, time, completion status, and accuracy for each experiment.

```
7  //author Melih Yanalak
8
9  public class Main {
10
11     public static void main(String[] args) {
12         ExperimentList eList = new ExperimentList();
13         Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float)0.9);
14         Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 2, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float)0.9);
15         Experiment e = new Experiment( setup_input: "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float)0.7);
16         Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float)0.7);
17         eList.addExp(e2); // adding when head is null
18         eList.addExp(e4); // adding to the end of the list
19         eList.addExp(e); // adding to the head of the list
20         eList.addExp(e3); // adding to the middle of the list
21         eList.orderDay(2); // ordering day 2 according to accuracy
22         eList.listExp( day: 2); // lists day 2
23     }
24 }
25
26
```

Run: Main x

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
SETUP : e3 - DAY : 2 - TIME : 12:50:00 - COMPLETED : true - ACCURACY : 0.7

SETUP : e4 - DAY : 2 - TIME : 10:00:00 - COMPLETED : true - ACCURACY : 0.9

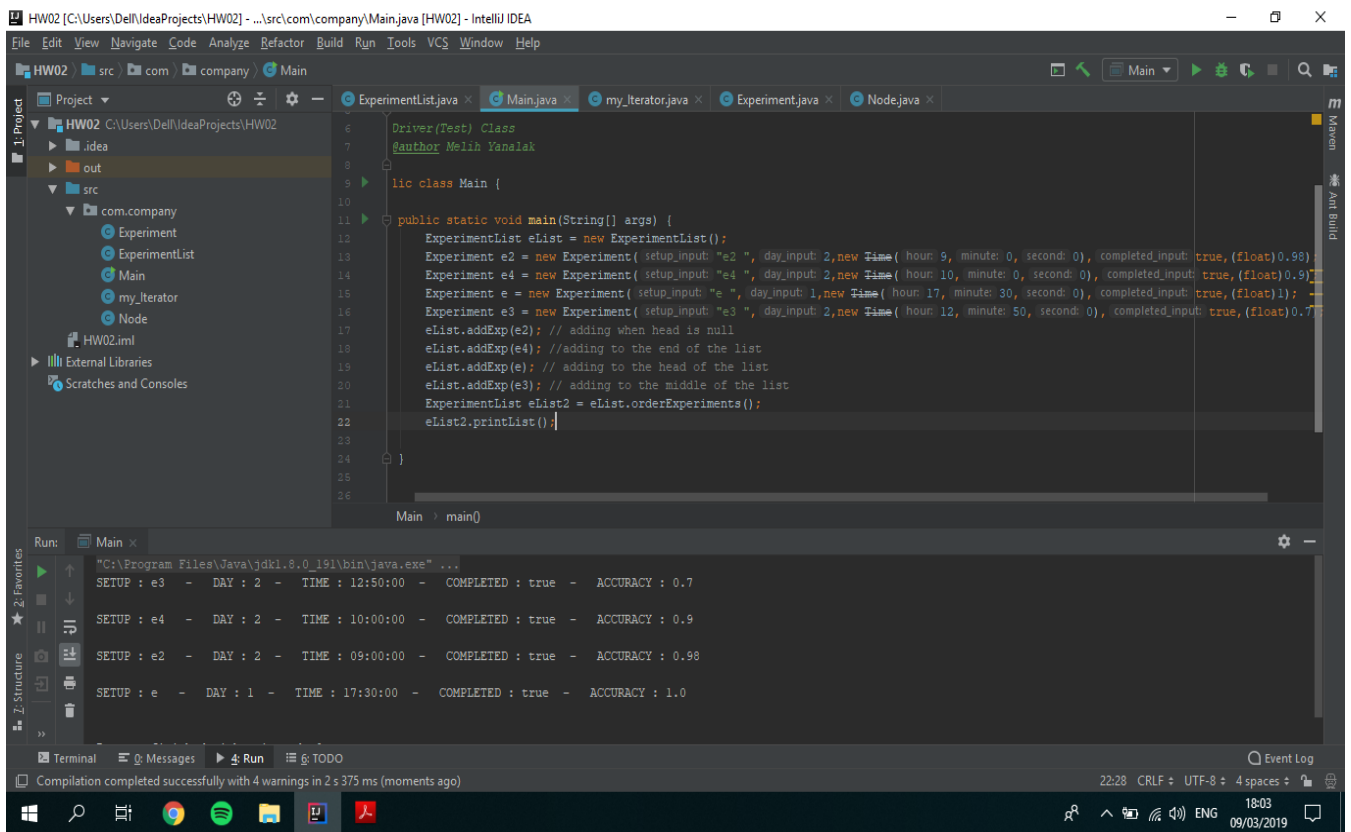
SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98
```

Terminal | Messages | Run | TODO

Compilation completed successfully with 4 warnings in 2 s 422 ms (a minute ago)

22:41 CRLF : UTF-8 : 4 spaces : ENG 18:01 09/03/2019

OrderExperiments method



The screenshot shows the IntelliJ IDEA IDE with a project named HW02. The main editor displays the `Main.java` file, which contains the `OrderExperiments` method. The method is implemented as follows:

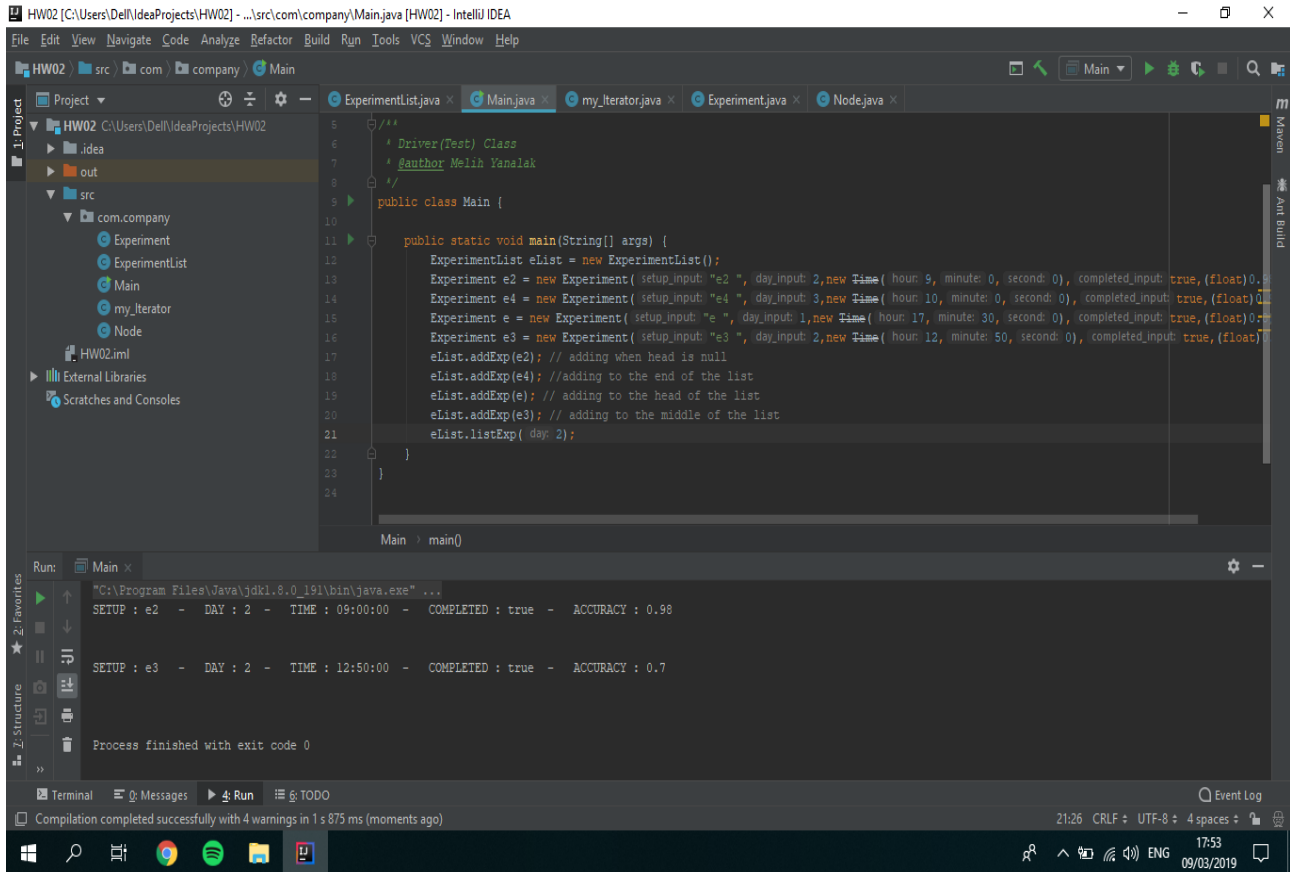
```
public static void main(String[] args) {  
    ExperimentList eList = new ExperimentList();  
    Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float)0.98);  
    Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 2, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float)0.9);  
    Experiment e = new Experiment( setup_input: "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float)1);  
    Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float)0.7);  
    eList.addExp(e2); // adding when head is null  
    eList.addExp(e4); // adding to the end of the list  
    eList.addExp(e); // adding to the head of the list  
    eList.addExp(e3); // adding to the middle of the list  
    ExperimentList eList2 = eList.orderExperiments();  
    eList2.printList();  
}
```

The Run window shows the output of the program, which is a list of experiments sorted by their completion time. The output is as follows:

```
Run: Main x  
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...  
SETUP : e3 - DAY : 2 - TIME : 12:50:00 - COMPLETED : true - ACCURACY : 0.7  
SETUP : e4 - DAY : 2 - TIME : 10:00:00 - COMPLETED : true - ACCURACY : 0.9  
SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98  
SETUP : e - DAY : 1 - TIME : 17:30:00 - COMPLETED : true - ACCURACY : 1.0
```

The status bar at the bottom indicates that the compilation was successful with 4 warnings in 2 s 375 ms (moments ago). The system clock shows 22:28 on 09/03/2019.

ListExp Method



The screenshot displays the IntelliJ IDEA IDE with a project named HW02. The main editor shows the `Main.java` file, which contains a `Main` class with a `main` method. The `main` method creates an `ExperimentList` object and adds four `Experiment` objects to it. The `listExp` method is called with the argument `day: 2`.

```
5  /**
6   * Driver (Test) Class
7   * @author Melih Yanalak
8   */
9  public class Main {
10
11     public static void main(String[] args) {
12         ExperimentList eList = new ExperimentList();
13         Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float)0.98);
14         Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 3, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float)0.7);
15         Experiment e = new Experiment( setup_input: "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float)0.7);
16         Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float)0.7);
17         eList.addExp(e2); // adding when head is null
18         eList.addExp(e4); // adding to the end of the list
19         eList.addExp(e); // adding to the head of the list
20         eList.addExp(e3); // adding to the middle of the list
21         eList.listExp( day: 2);
22     }
23 }
24
```

The Run window shows the output of the `main` method:

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
SETUP : e2 - DAY : 2 - TIME : 09:00:00 - COMPLETED : true - ACCURACY : 0.98
SETUP : e3 - DAY : 2 - TIME : 12:50:00 - COMPLETED : true - ACCURACY : 0.7
Process finished with exit code 0
```

The bottom status bar indicates that the compilation was successful with 4 warnings in 1 s 875 ms (moments ago).

SetExp and GetExp Methods

The screenshot displays the IntelliJ IDEA IDE with a project named HW02. The main class, Main.java, is open, showing the following code:

```
public class Main {  
    public static void main(String[] args) {  
        ExperimentList eList = new ExperimentList();  
        Experiment e2 = new Experiment( setup_input: "e2 ", day_input: 2, new Time( hour: 9, minute: 0, second: 0), completed_input: true, (float)0.9);  
        Experiment e4 = new Experiment( setup_input: "e4 ", day_input: 3, new Time( hour: 10, minute: 0, second: 0), completed_input: true, (float)0.8);  
        Experiment e = new Experiment( setup_input: "e ", day_input: 1, new Time( hour: 17, minute: 30, second: 0), completed_input: true, (float)0.7);  
        Experiment e3 = new Experiment( setup_input: "e3 ", day_input: 2, new Time( hour: 12, minute: 50, second: 0), completed_input: true, (float)0.6);  
        eList.addExp(e2); //adding when head is null  
        eList.addExp(e4); //adding to the end of the list  
        eList.addExp(e); // adding to the head of the list  
        eList.addExp(e3); // adding to the middle of the list  
        Experiment e5 = new Experiment( setup_input: "e5 ", day_input: 2, new Time( hour: 0, minute: 0, second: 0), completed_input: false, (float)0.5);  
        eList.setExp( day: 2, index: 1, e5);  
        System.out.println(eList.getExp( day: 2, index: 1).getData());  
    }  
}
```

The Run window shows the output of the program:

```
SETUP : e5 - DAY : 2 - TIME : 00:00:00 - COMPLETED : false - ACCURACY : 0.2  
  
Process finished with exit code 0
```

The bottom status bar indicates that the compilation completed successfully with 5 warnings in 1 s 953 ms (moments ago).