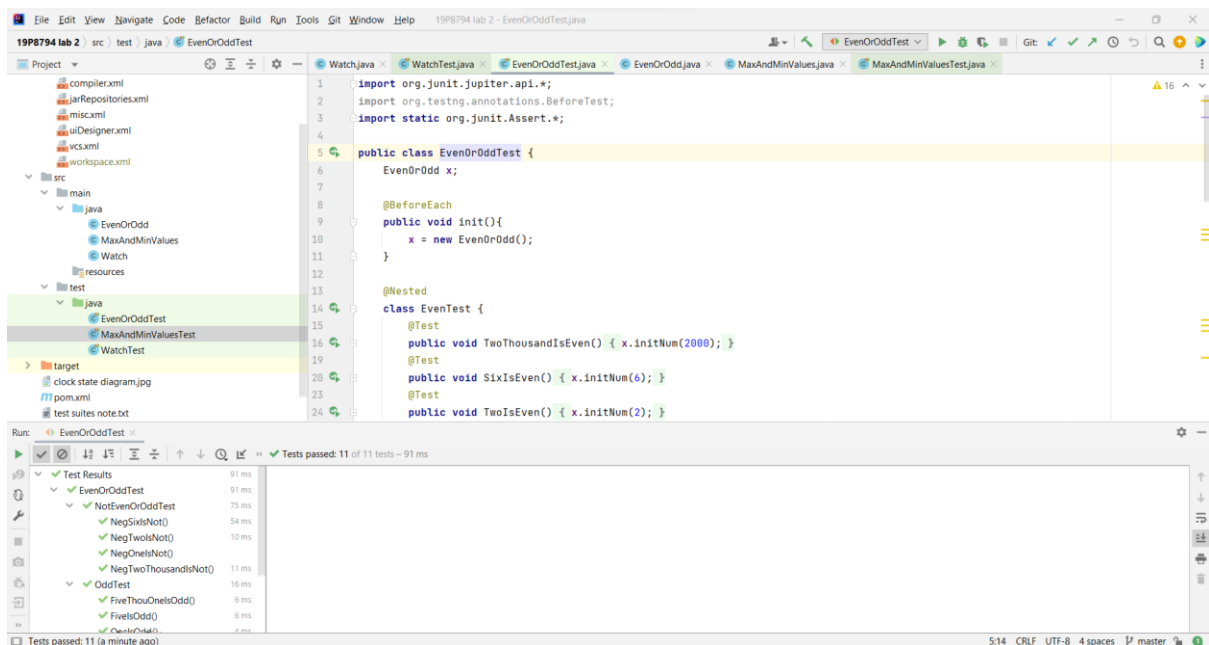
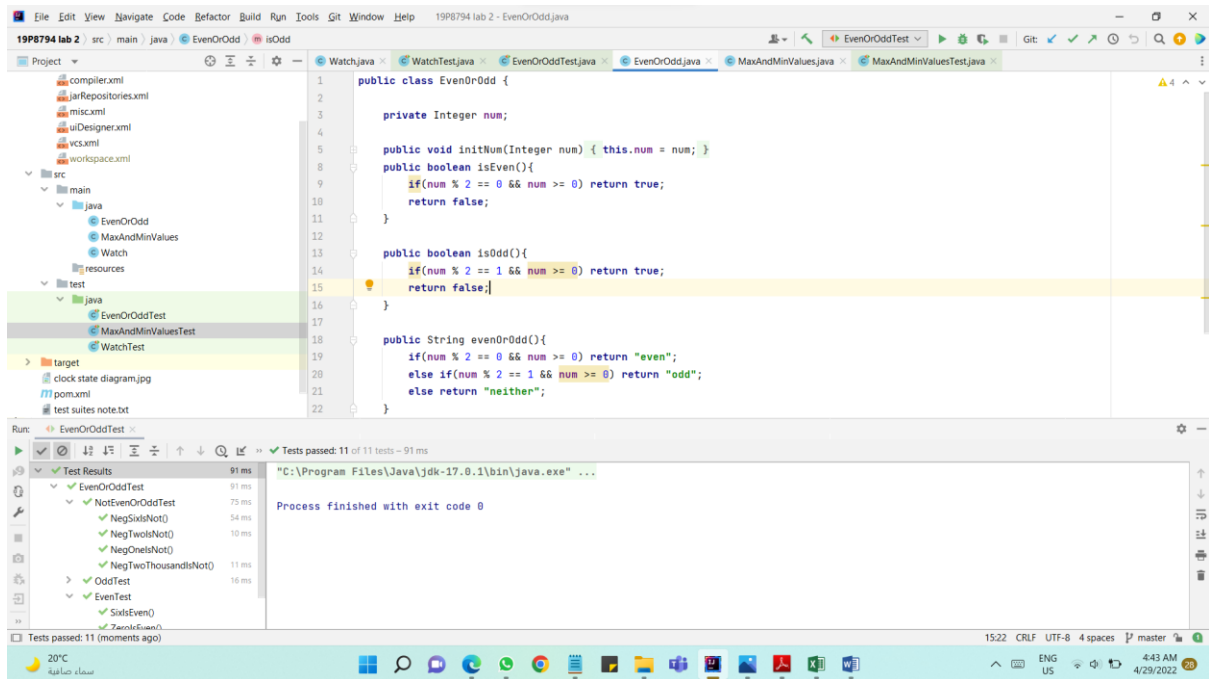
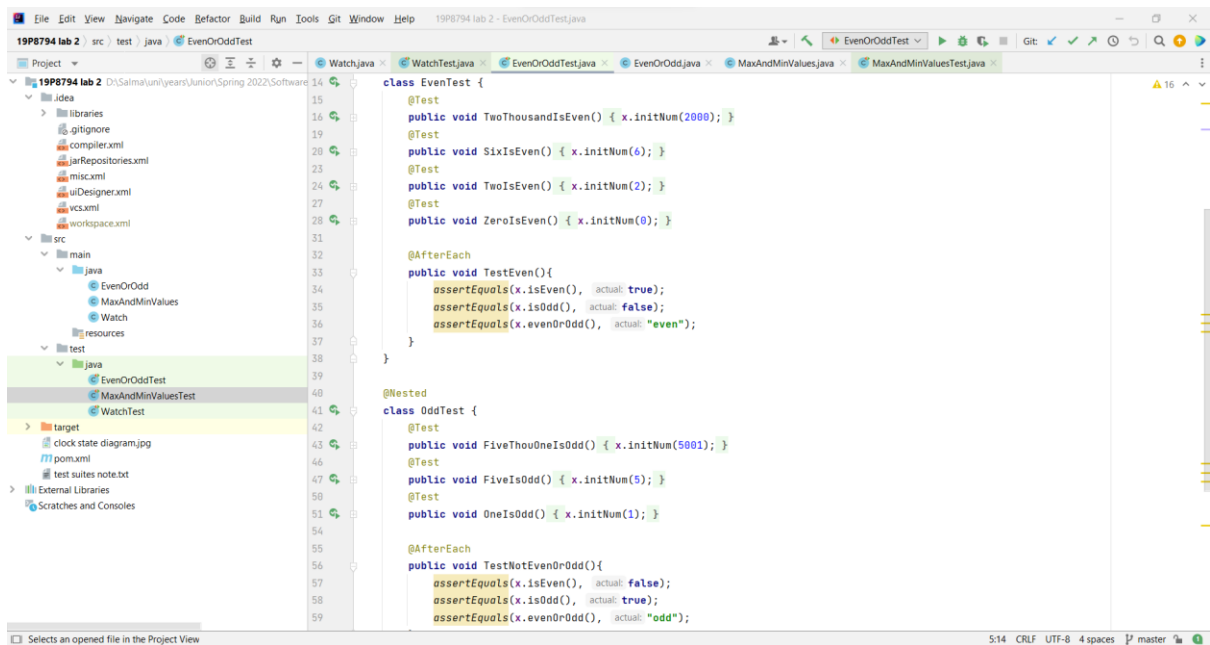


# CSE338 LAB 2 REPORT

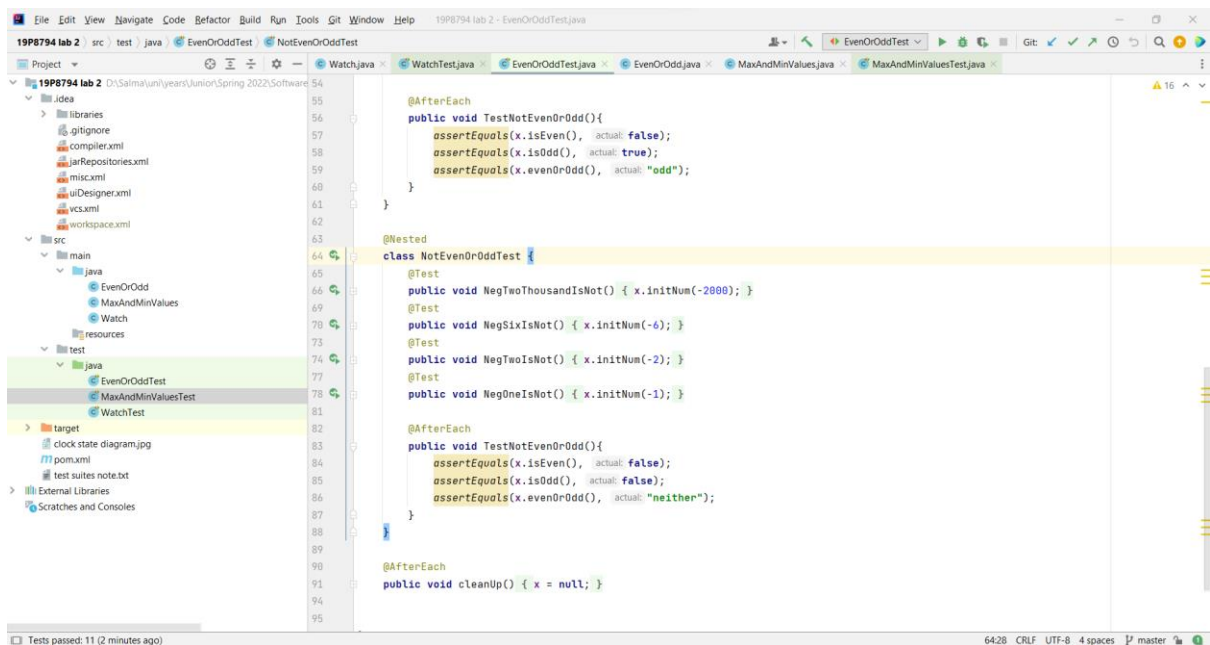
## Screenshots

### Q1 a)





```
14 class EvenTest {
15     @Test
16     public void TwoThousandIsEven() { x.initNum(2000); }
17
18     @Test
19     public void SixIsEven() { x.initNum(6); }
20
21     @Test
22     public void TwoIsEven() { x.initNum(2); }
23
24     @Test
25     public void ZeroIsEven() { x.initNum(0); }
26
27     @AfterEach
28     public void TestEven(){
29         assertEquals(x.isEven(), actual: true);
30         assertEquals(x.isOdd(), actual: false);
31         assertEquals(x.evenOrOdd(), actual: "even");
32     }
33 }
34
35 @Nested
36 class OddTest {
37     @Test
38     public void FiveThouOneIsOdd() { x.initNum(5001); }
39
40     @Test
41     public void FiveIsOdd() { x.initNum(5); }
42
43     @Test
44     public void OneIsOdd() { x.initNum(1); }
45
46     @AfterEach
47     public void TestNotEvenOrOdd(){
48         assertEquals(x.isEven(), actual: false);
49         assertEquals(x.isOdd(), actual: true);
50         assertEquals(x.evenOrOdd(), actual: "odd");
51     }
52 }
```



```
54
55
56 @AfterEach
57 public void TestNotEvenOrOdd(){
58     assertEquals(x.isEven(), actual: false);
59     assertEquals(x.isOdd(), actual: true);
60     assertEquals(x.evenOrOdd(), actual: "odd");
61 }
62
63 @Nested
64 class NotEvenOrOddTest {
65     @Test
66     public void NegTwoThousandIsNot() { x.initNum(-2000); }
67
68     @Test
69     public void NegSixIsNot() { x.initNum(-6); }
70
71     @Test
72     public void NegTwoIsNot() { x.initNum(-2); }
73
74     @Test
75     public void NegOneIsNot() { x.initNum(-1); }
76
77     @AfterEach
78     public void TestNotEvenOrOdd(){
79         assertEquals(x.isEven(), actual: false);
80         assertEquals(x.isOdd(), actual: false);
81         assertEquals(x.evenOrOdd(), actual: "neither");
82     }
83 }
84
85 @AfterEach
86 public void cleanUp() { x = null; }
87
88
89
90
91
92
93
94
95
```

## Q1 b)

The screenshot displays the IntelliJ IDEA IDE with two tabs open: `MaxAndMinValues.java` and `MaxAndMinValuesTest.java`.

**MaxAndMinValues.java** contains the following code:

```
1 public class MaxAndMinValues {
2     Integer arr[];
3
4     public void initArray(Integer arr[]){
5         this.arr = arr;
6         for(int i = 0; i < arr.length; i++){
7             this.arr[i] = arr[i];
8         }
9     }
10
11     public Integer Max(){
12         Integer max = -99999;
13         for(int i = 0; i < this.arr.length; i++){
14             if(arr[i] > max) max = this.arr[i];
15         }
16         return max;
17     }
18
19     public Integer Min(){
20         Integer min = 99999;
21         for(int i = 0; i < this.arr.length; i++){
22             if(arr[i] < min) min = this.arr[i];
23         }
24         return min;
25     }
26 }
```

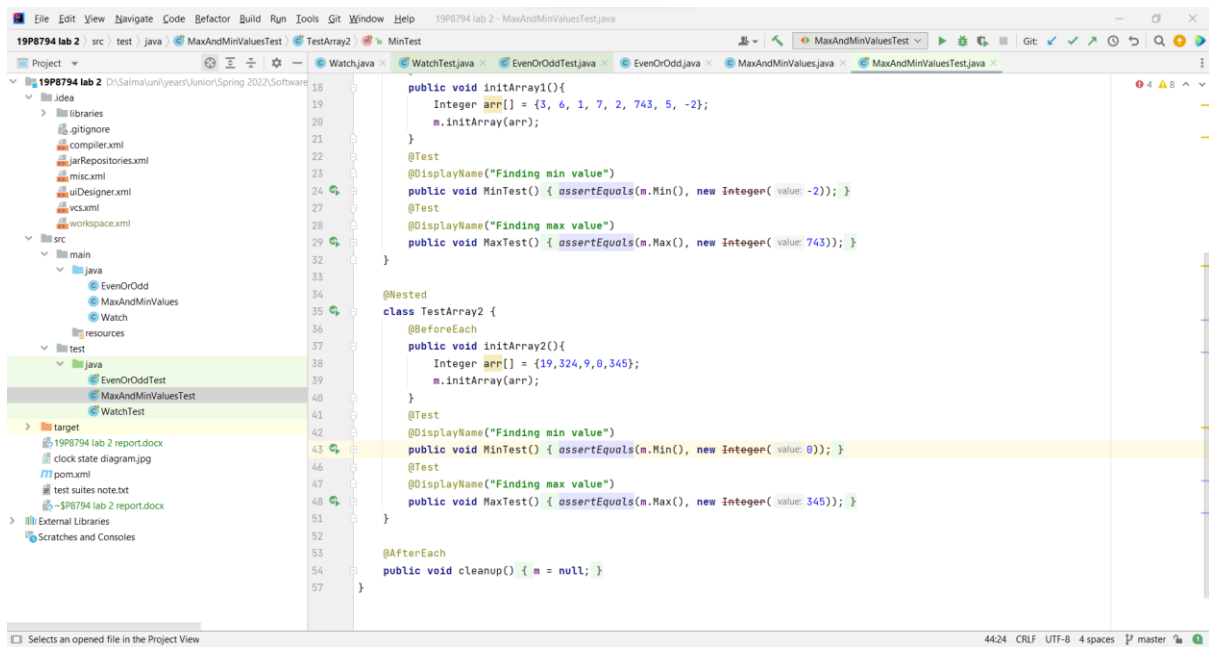
**MaxAndMinValuesTest.java** contains the following code:

```
1 import org.junit.jupiter.api.*;
2 import org.junit.jupiter.api.*;
3 import static org.junit.Assert.*;
4 import static org.junit.Assert.*;
5
6 public class MaxAndMinValuesTest {
7     MaxAndMinValues m;
8
9     @BeforeEach
10    public void init() { m = new MaxAndMinValues(); }
11
12    @Nested
13    class TestArray1 {
14        @BeforeEach
15        public void initArray1(){
16            Integer arr[] = {3, 6, 1, 7, 2, 743, 5, -2};
17            m.initArray(arr);
18        }
19        @Test
20        // ...
21    }
22 }
```

The **Run** tab at the bottom shows the test results for `MaxAndMinValuesTest`:

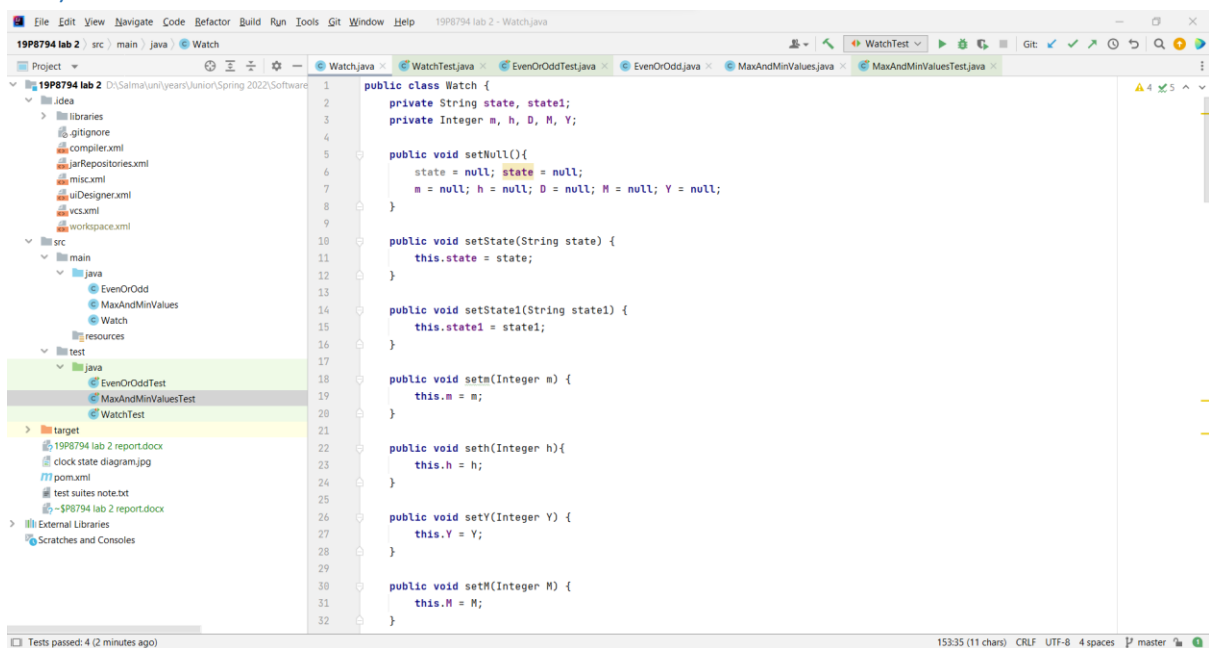
- Test Results: 4 of 4 tests - 51 ms
- MaxAndMinValuesTest: 51 ms
- TestArray2: 46 ms
- TestArray1: 5 ms
- Finding max value: 5 ms
- Finding min value: 5 ms

The status bar at the bottom indicates: 44:24 CRLF UTF-8 4 spaces master.

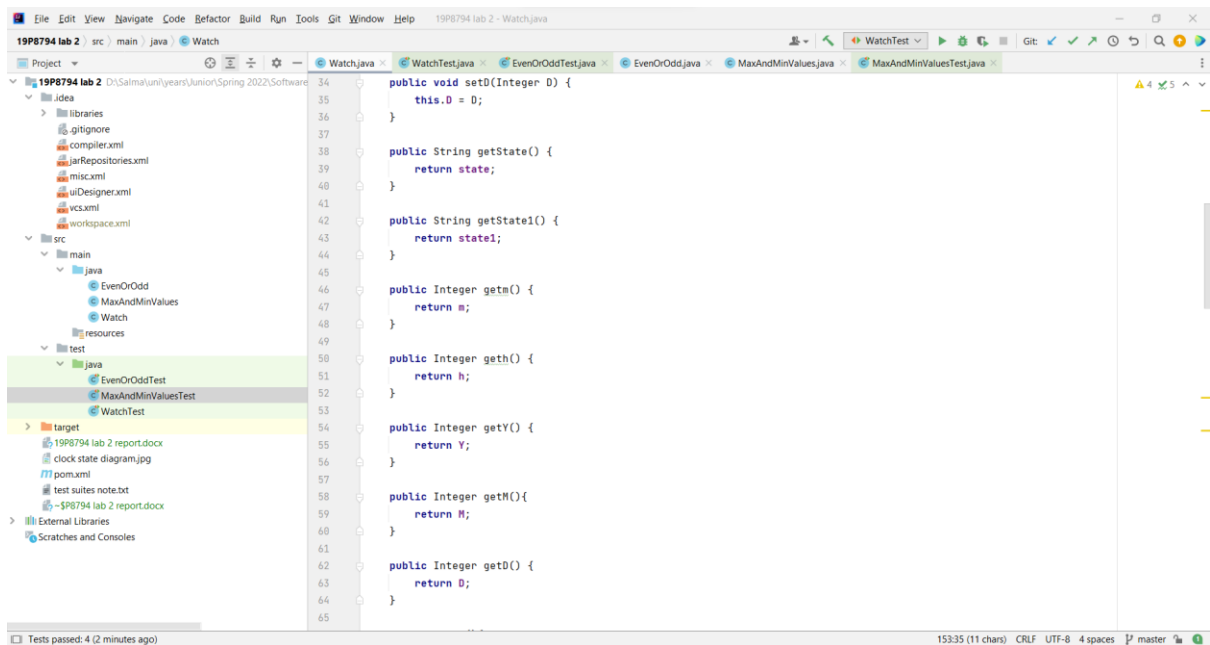


```
18 public void initArray1(){
19     Integer arr[] = {3, 0, 1, 7, 2, 743, 5, -2};
20     m.initArray(arr);
21 }
22 @Test
23 @DisplayName("Finding min value")
24 public void MinTest() { assertEquals(m.Min(), new Integer( value: -2)); }
25 @Test
26 @DisplayName("Finding max value")
27 public void MaxTest() { assertEquals(m.Max(), new Integer( value: 743)); }
28 }
29
30 @Nested
31 class TestArray2 {
32     @BeforeEach
33     public void initArray2(){
34         Integer arr[] = {19,324,9,0,345};
35         m.initArray(arr);
36     }
37     @Test
38     @DisplayName("Finding min value")
39     public void MinTest() { assertEquals(m.Min(), new Integer( value: 0)); }
40     @Test
41     @DisplayName("Finding max value")
42     public void MaxTest() { assertEquals(m.Max(), new Integer( value: 345)); }
43 }
44
45 @AfterEach
46 public void cleanup() { m = null; }
47 }
```

Q2)

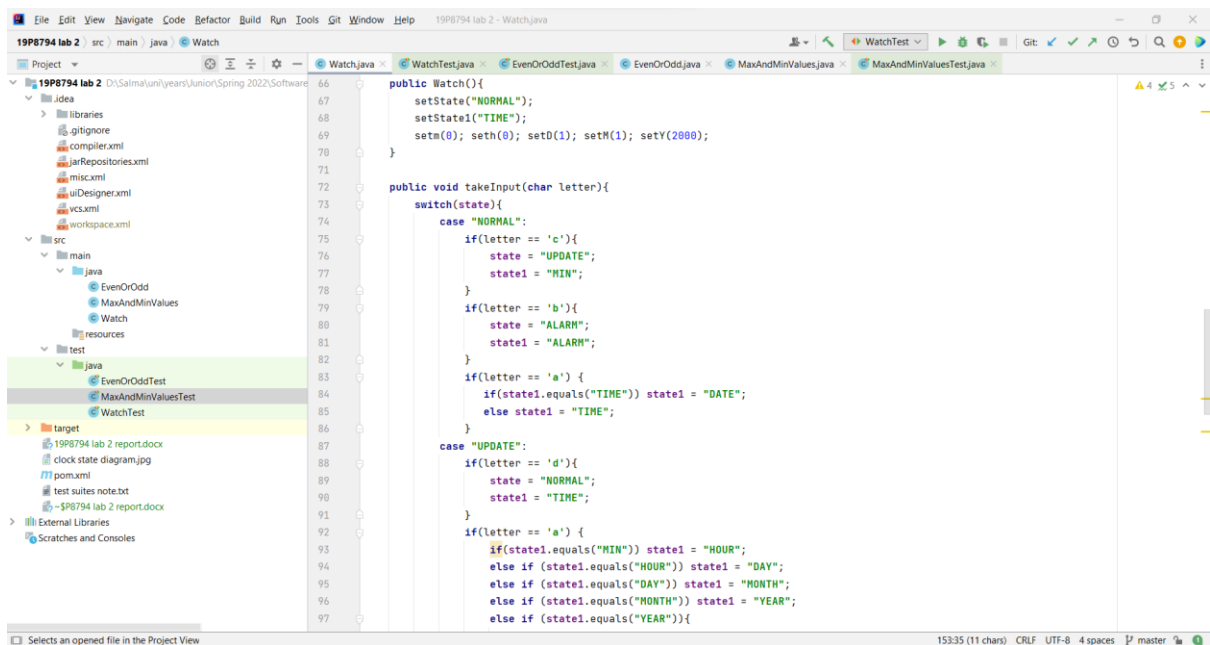


```
1 public class Watch {
2     private String state, state1;
3     private Integer m, h, D, M, Y;
4
5     public void setNull(){
6         state = null; state1 = null;
7         m = null; h = null; D = null; M = null; Y = null;
8     }
9
10    public void setState(String state) {
11        this.state = state;
12    }
13
14    public void setState1(String state1) {
15        this.state1 = state1;
16    }
17
18    public void setm(Integer m) {
19        this.m = m;
20    }
21
22    public void seth(Integer h){
23        this.h = h;
24    }
25
26    public void setY(Integer Y) {
27        this.Y = Y;
28    }
29
30    public void setM(Integer M) {
31        this.M = M;
32    }
33 }
```



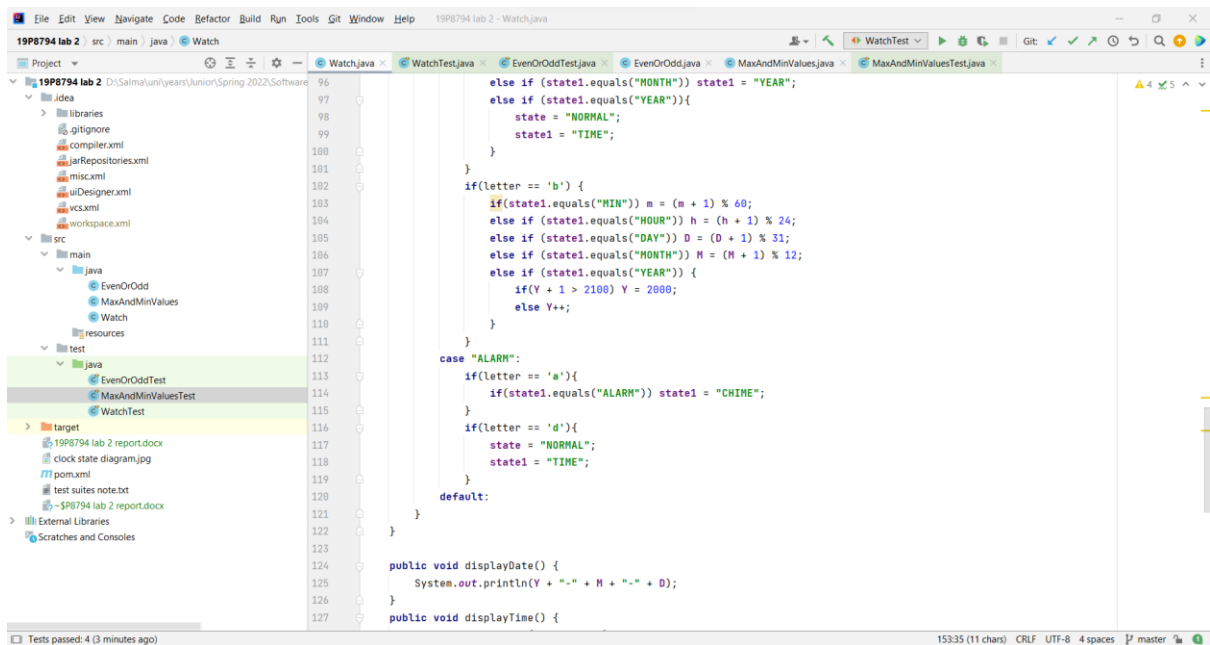
```
1  public void setD(Integer D) {
2      this.D = D;
3  }
4
5  public String getState() {
6      return state;
7  }
8
9  public String getState1() {
10     return state1;
11 }
12
13 public Integer getm() {
14     return m;
15 }
16
17 public Integer geth() {
18     return h;
19 }
20
21 public Integer getY() {
22     return Y;
23 }
24
25 public Integer getM(){
26     return M;
27 }
28
29 public Integer getD() {
30     return D;
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
```

Tests passed: 4 (2 minutes ago) 153:35 (11 chars) CRLF UTF-8 4 spaces master



```
1  public Watch(){
2      setState("NORMAL");
3      setState1("TIME");
4      setm(0); seth(0); setD(1); setY(2000);
5  }
6
7  public void takeInput(char letter){
8      switch(state){
9          case "NORMAL":
10             if(letter == 'c'){
11                 state = "UPDATE";
12                 state1 = "MIN";
13             }
14             if(letter == 'b'){
15                 state = "ALARM";
16                 state1 = "ALARM";
17             }
18             if(letter == 'a') {
19                 if(state1.equals("TIME")) state1 = "DATE";
20                 else state1 = "TIME";
21             }
22             case "UPDATE":
23                 if(letter == 'd'){
24                     state = "NORMAL";
25                     state1 = "TIME";
26                 }
27                 if(letter == 'a') {
28                     if(state1.equals("MIN")) state1 = "HOUR";
29                     else if (state1.equals("HOUR")) state1 = "DAY";
30                     else if (state1.equals("DAY")) state1 = "MONTH";
31                     else if (state1.equals("MONTH")) state1 = "YEAR";
32                     else if (state1.equals("YEAR")){
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```

Selects an opened file in the Project View 153:35 (11 chars) CRLF UTF-8 4 spaces master



```
19P8794 lab 2 - Watch.java
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Project 19P8794 lab 2 src main java Watch
libraries .gitignore compiler.xml jarRepositories.xml misc.xml uiDesigner.xml vcs.xml workspace.xml
src main java EvenOrOdd MaxAndMinValues Watch
resources
test java EvenOrOddTest MaxAndMinValuesTest WatchTest
target 19P8794 lab 2 report.docx clock state diagram.jpg pom.xml test suites note.txt 19P8794 lab 2 report.docx
External Libraries
Scratches and Consoles

Watch.java
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127

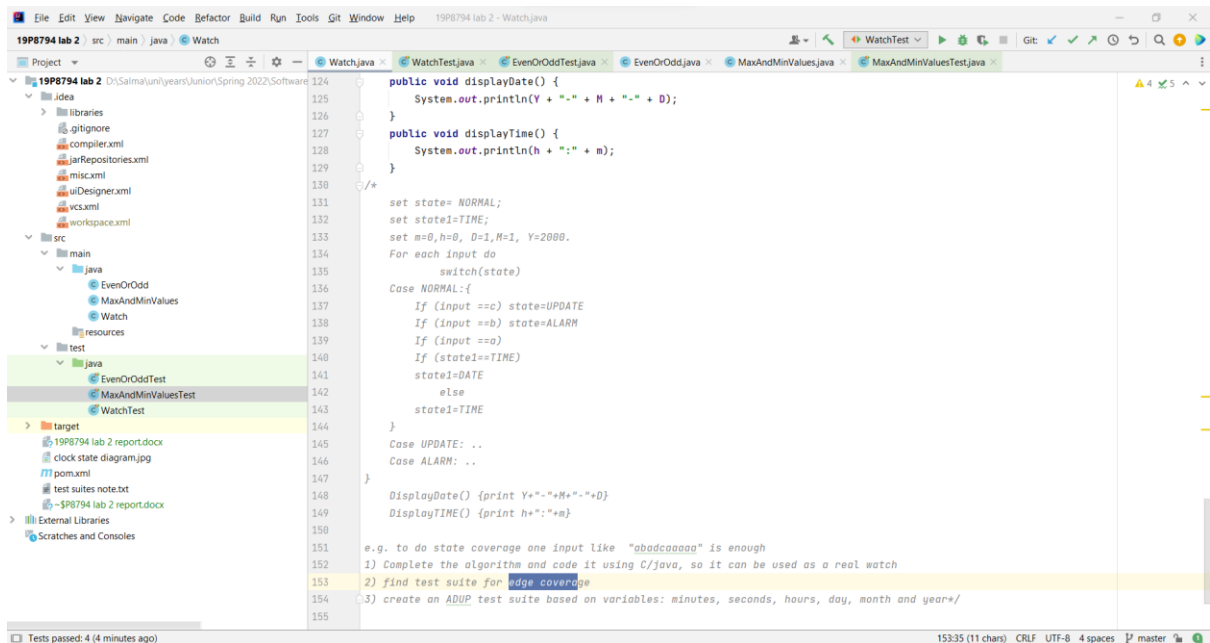
else if (state1.equals("MONTH")) state1 = "YEAR";
else if (state1.equals("YEAR")){
    state = "NORMAL";
    state1 = "TIME";
}
}
if(letter == 'b') {
    if(state1.equals("MIN")) m = (m + 1) % 60;
    else if (state1.equals("HOUR")) h = (h + 1) % 24;
    else if (state1.equals("DAY")) D = (D + 1) % 31;
    else if (state1.equals("MONTH")) M = (M + 1) % 12;
    else if (state1.equals("YEAR")) {
        if(Y + 1 > 2100) Y = 2000;
        else Y++;
    }
}
case "ALARM":
    if(letter == 'a'){
        if(state1.equals("ALARM")) state1 = "CHIME";
    }
    if(letter == 'd'){
        state = "NORMAL";
        state1 = "TIME";
    }
}
default:
}

public void displayDate() {
    System.out.println(Y + "-" + M + "-" + D);
}

public void displayTime() {

```

Tests passed: 4 (3 minutes ago) 153:35 (11 chars) CRLF UTF-8 4 spaces master



```
19P8794 lab 2 - Watch.java
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Project 19P8794 lab 2 src main java Watch
libraries .gitignore compiler.xml jarRepositories.xml misc.xml uiDesigner.xml vcs.xml workspace.xml
src main java EvenOrOdd MaxAndMinValues Watch
resources
test java EvenOrOddTest MaxAndMinValuesTest WatchTest
target 19P8794 lab 2 report.docx clock state diagram.jpg pom.xml test suites note.txt 19P8794 lab 2 report.docx
External Libraries
Scratches and Consoles

Watch.java
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155

public void displayDate() {
    System.out.println(Y + "-" + M + "-" + D);
}

public void displayTime() {
    System.out.println(h + ":" + m);
}

/*
set state= NORMAL;
set state1=TIME;
set m=0,h=0, D=1,M=1, Y=2000.
For each input do
    switch(state)
    Case NORMAL:{
        If (input ==c) state=UPDATE
        If (input ==b) state=ALARM
        If (input ==a)
            state1=DATE
            state1=TIME
        }
    Case UPDATE: ..
    Case ALARM: ..
}

DisplayDate() {print Y+"-"+M+"-"+D}
DisplayTIME() {print h+" ":"m}

e.g. to do state coverage one input like "abadcaaaaa" is enough
1) Complete the algorithm and code it using C/java, so it can be used as a real watch
2) find test suite for edge coverage
3) create an ADUP test suite based on variables: minutes, seconds, hours, day, month and year*/

```

Tests passed: 4 (4 minutes ago) 153:35 (11 chars) CRLF UTF-8 4 spaces master

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named "19P8794 lab 2" with a directory structure including "src" and "test".
- Code Editor:** Displays the source code for `WatchTest.java` and `EdgeTest.java`.
  - `WatchTest.java` (lines 1-20):

```
1 import org.junit.jupiter.api.*;
2 import org.junit.jupiter.api.Test;
3 import org.testing.annotations.*;
4
5 import java.util.Vector;
6
7 import static org.junit.Assert.*;
8
9 public class WatchTest {
10     Watch watch;
11     //static Vector<String> //edgeTCs = new Vector<String>();
12
13     @BeforeEach
14     public void init(){
15         watch = new Watch();
16     }
17
18     @Nested
19     @DisplayName("Edge Testing")
20     ...
```
  - `EdgeTest.java` (lines 21-117):

```
21 class EdgeTest{
22     @Test
23     public void EdgeTestNormal(){...}
24
25     @Test
26     public void EdgeTestUpdate(){...}
27
28     @Test
29     public void EdgeTestAlarm(){...}
30
31     //TODO fix annot @Test
32     //@@Test
33     //public void outputSuite(){
34     //    //System.out.println("Edge Test Suite = "); //+ edgeTCs);
35     //}
36 }
```
- Run Window:** Shows the execution of `WatchTest`. It indicates that 6 tests passed in 80 ms. The test results are as follows:

Test Name	Duration
ADUP Testing	80 ms
ADUP10	80 ms
ADUP20	75 ms
ADUP30	5 ms
Edge Testing	
EdgeTestNormal()	
EdgeTestAlarm()	
EdgeTestUpdate()	

```
class EdgeTest{  
    @Test  
    public void EdgeTestNormal(){  
        watch.takeInput( letter: 'a');    //edgeTCs.add("a");  
  
        assertEquals(watch.getState(), actual: "NORMAL");  
        assertEquals(watch.getState1(), actual: "DATE");  
  
        watch.takeInput( letter: 'a');    //edgeTCs.add("a");  
  
        assertEquals(watch.getState(), actual: "NORMAL");  
        assertEquals(watch.getState1(), actual: "TIME");  
    }  
    @Test  
    public void EdgeTestUpdate(){  
        watch.takeInput( letter: 'c'); //from normal to update  
        //edgeTCs.add("c");  
  
        watch.takeInput( letter: 'b');  
        //edgeTCs.add("b");  
  
        assertEquals(watch.getState(), actual: "UPDATE");  
        assertEquals(watch.getState1(), actual: "MIN");  
  
        watch.takeInput( letter: 'a');  
        watch.takeInput( letter: 'b');  
        //edgeTCs.add("a");  
        //edgeTCs.add("b");  
  
        assertEquals(watch.getState(), actual: "UPDATE");  
        assertEquals(watch.getState1(), actual: "HOUR");  
    }  
}
```



```
53 watch.takeInput( letter: 'a');
54 watch.takeInput( letter: 'b');
55 //edgeTCs.add("a");
56 //edgeTCs.add("b");
57 assertEquals(watch.getState(), actual: "UPDATE");
58 assertEquals(watch.getState1(), actual: "DAY");
59
60 watch.takeInput( letter: 'a');
61 watch.takeInput( letter: 'b');
62 //edgeTCs.add("a");
63 //edgeTCs.add("b");
64 assertEquals(watch.getState(), actual: "UPDATE");
65 assertEquals(watch.getState1(), actual: "MONTH");
66
67 watch.takeInput( letter: 'a');
68 watch.takeInput( letter: 'b');
69 //edgeTCs.add("a");
70 //edgeTCs.add("b");
71 assertEquals(watch.getState(), actual: "UPDATE");
72 assertEquals(watch.getState1(), actual: "YEAR");
73
74 watch.takeInput( letter: 'a'); //exit update
75 //edgeTCs.add("a");
76 assertEquals(watch.getState(), actual: "NORMAL");
77 assertEquals(watch.getState1(), actual: "TIME");
78
79 watch.takeInput( letter: 'c'); //to update
80 //edgeTCs.add("c");
81 assertEquals(watch.getState(), actual: "UPDATE");
82 assertEquals(watch.getState1(), actual: "MIN");
83 watch.takeInput( letter: 'd'); //exit update
84 //edgeTCs.add("d");
```

```
84 //edgeTCs.add("d");
85 assertEquals(watch.getState(), actual: "NORMAL");
86 assertEquals(watch.getState1(), actual: "TIME");
87
88 assertEquals(watch.getM(), new Integer( value: 1));
89 assertEquals(watch.getH(), new Integer( value: 1));
90 assertEquals(watch.getD(), new Integer( value: 2));
91 assertEquals(watch.getM(), new Integer( value: 2));
92 assertEquals(watch.getY(), new Integer( value: 2001));
93
94 System.out.print("\nDate: ");
95 watch.displayDate();
96 System.out.print("\nTime: ");
97 watch.displayTime();
98 }
99 @Test
100 public void EdgeTestAlarm(){
101     watch.takeInput( letter: 'b'); //edgeTCs.add("b");
102     assertEquals(watch.getState(), actual: "ALARM");
103     assertEquals(watch.getState1(), actual: "ALARM");
104     watch.takeInput( letter: 'a'); //edgeTCs.add("a");
105     assertEquals(watch.getState(), actual: "ALARM");
106     assertEquals(watch.getState1(), actual: "CHIME");
107     watch.takeInput( letter: 'd'); //edgeTCs.add("d");
108     assertEquals(watch.getState(), actual: "NORMAL");
109     assertEquals(watch.getState1(), actual: "TIME");
110 }
111
112 //TODO fix annot @Test
113 //@Test
114 //public void outputSuite(){
115
116 //define:
117 //setNull method
118 //setters
119 //constructor (none to use associations here since it uses setters immediately)
120 //takeInput
121 //use:
122 //takeInput
123
124 //thus, all def-use =
125 //1. setters to takeInput
126 //2. setNull to takeInput
127 //3. takeInput to takeInput
```

```
132     @Nested
133     @DisplayName("ADUP Testing")
134     class ADUPTest{
135         //TODO fix ADUP1Cs [][] output
136         //static Vector<Vector<String>> ADUP1Cs = new Vector<Vector<String>>(3);
137         //static int i = 0;
138         @Test
139         public void ADUP1(){...}
192         @Test
193         public void ADUP2(){...}
281         @Test
282         public void ADUP3(){...}
329
330
331     }
332
333     /*...*/
346     /*...*/
357     @AfterEach
358     public void cleanUp(){
359         watch.setNull();
360         watch = null;
361     }
362
363 }
```

19P8794 lab 2 | src | test | java | WatchTest | ADUPTest

Run: WatchTest

Tests passed: 6 of 6 tests - 84 ms

Test Results

- WatchTest (84 ms)
  - ADUP Testing (84 ms)
    - ADUP1 (81 ms)
    - ADUP2 (3 ms)
    - ADUP3 (3 ms)
  - Edge Testing
    - EdgeTestNormal() (84 ms)
    - EdgeTestAlarm() (81 ms)
    - EdgeTestUpdate() (3 ms)

ADUP2 testing class: exception occurs when changing "m" value with the following message.  
Cannot invoke "java.lang.Integer.intValue()" because "this.m" is null

ADUP2 testing class: exception occurs when changing "h" value with the following message.  
Cannot invoke "java.lang.Integer.intValue()" because "this.h" is null

ADUP2 testing class: exception occurs when changing "D" value with the following message.  
Cannot invoke "java.lang.Integer.intValue()" because "this.D" is null

ADUP2 testing class: exception occurs when changing "M" value with the following message.  
Cannot invoke "java.lang.Integer.intValue()" because "this.M" is null

ADUP2 testing class: exception occurs when changing "Y" value with the following message.  
Cannot invoke "java.lang.Integer.intValue()" because "this.Y" is null

Date: null-null-null

Time: null:null

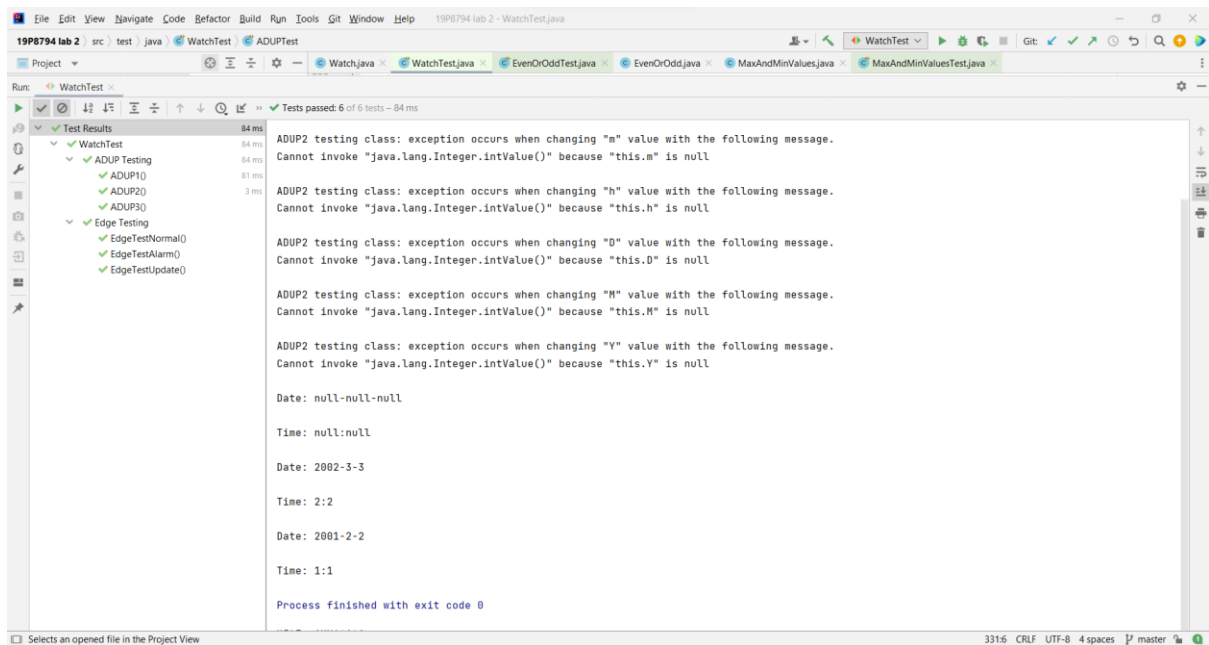
Date: 2002-3-3

Time: 2:2

Date: 2001-2-2

Tests passed: 6 (moments ago)

3316 CRLF UTF-8 4 spaces master



```
ADUP2 testing class: exception occurs when changing "m" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.m" is null

ADUP2 testing class: exception occurs when changing "h" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.h" is null

ADUP2 testing class: exception occurs when changing "D" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.D" is null

ADUP2 testing class: exception occurs when changing "M" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.M" is null

ADUP2 testing class: exception occurs when changing "Y" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.Y" is null

Date: null-null-null

Time: null:null

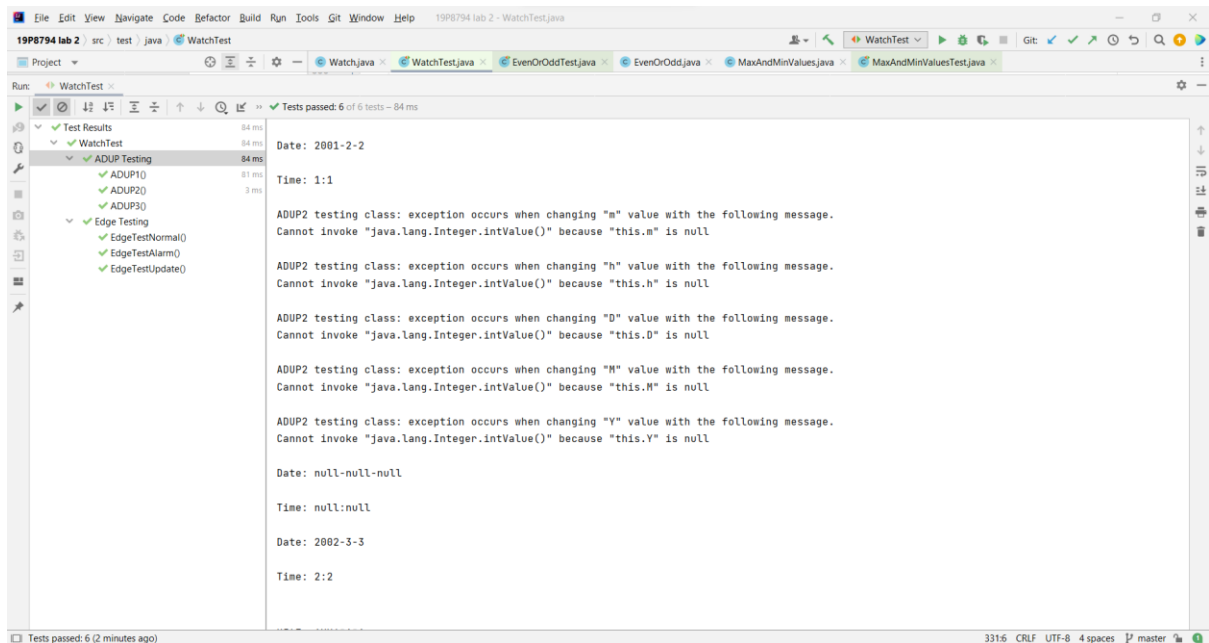
Date: 2002-3-3

Time: 2:2

Date: 2001-2-2

Time: 1:1

Process finished with exit code 0
```



```
Date: 2001-2-2

Time: 1:1

ADUP2 testing class: exception occurs when changing "m" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.m" is null

ADUP2 testing class: exception occurs when changing "h" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.h" is null

ADUP2 testing class: exception occurs when changing "D" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.D" is null

ADUP2 testing class: exception occurs when changing "M" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.M" is null

ADUP2 testing class: exception occurs when changing "Y" value with the following message.
Cannot invoke "java.lang.Integer.intValue()" because "this.Y" is null

Date: null-null-null

Time: null:null

Date: 2002-3-3

Time: 2:2
```

