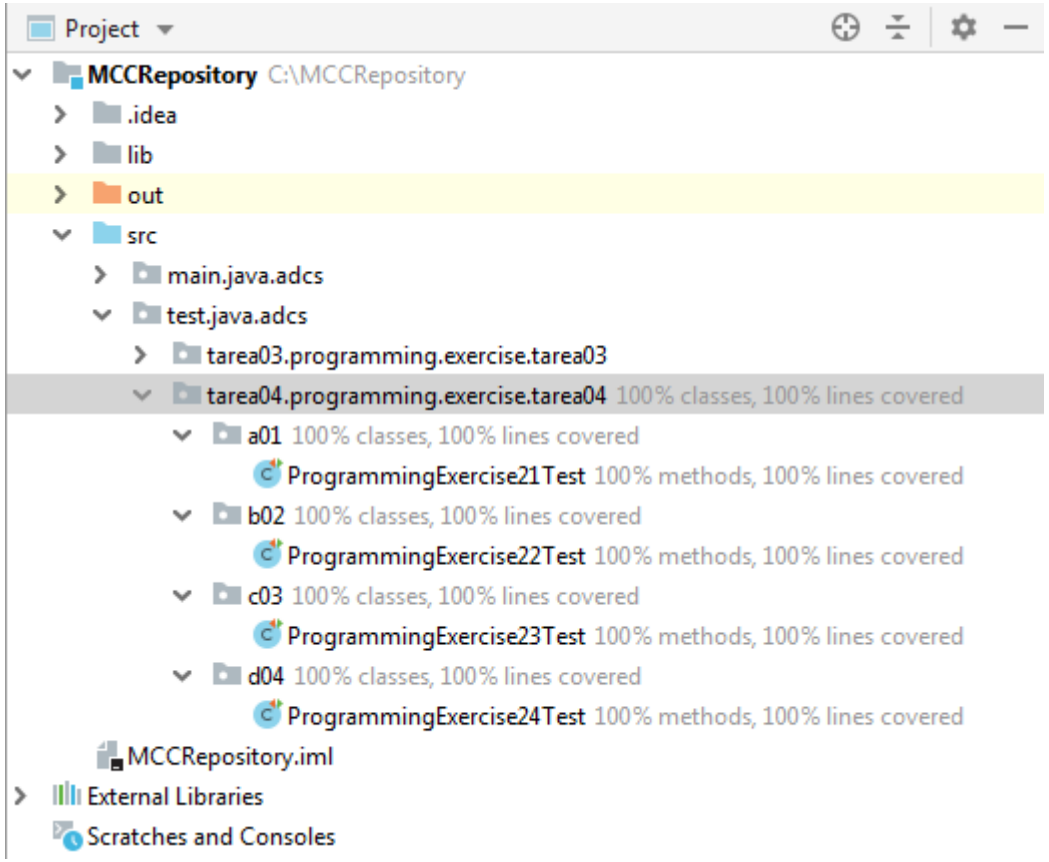
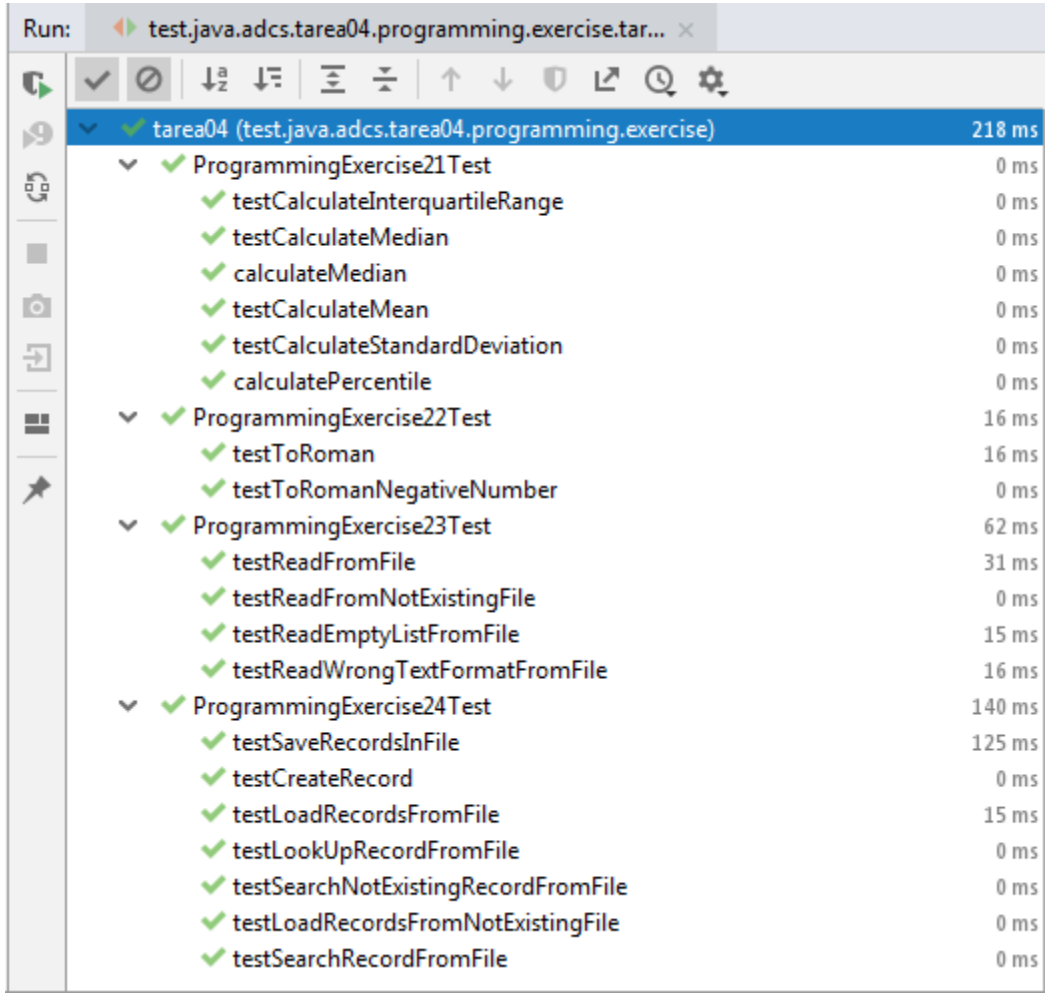
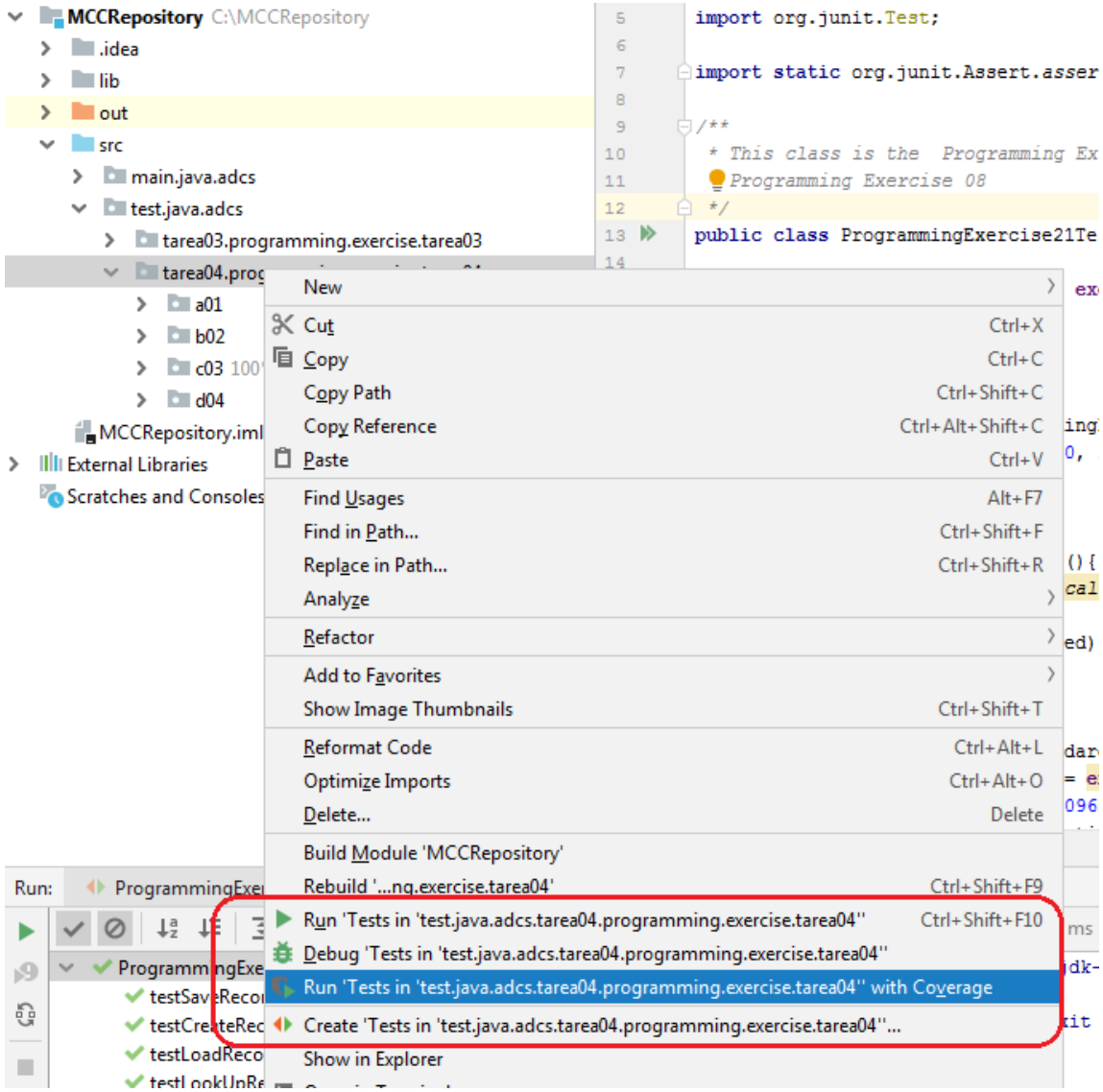
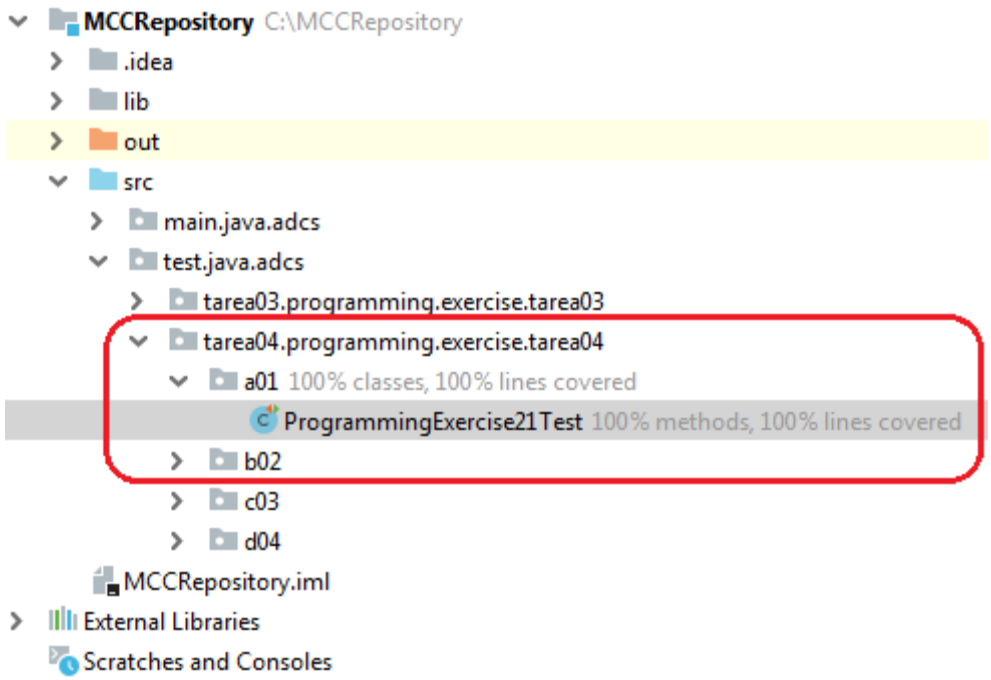
Programming Exercises 21-24

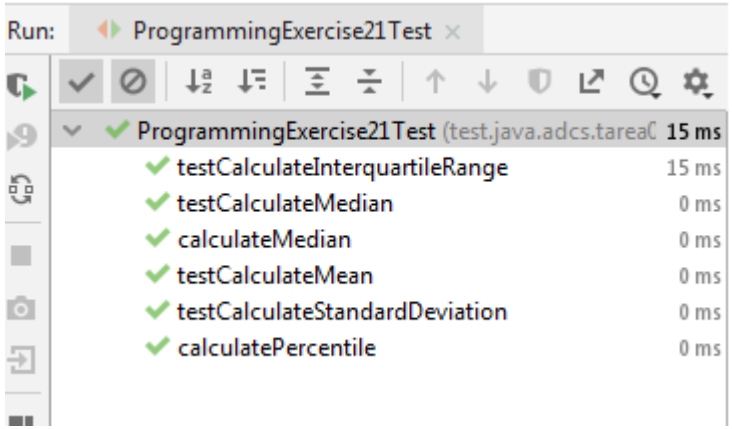



Run:
test.java.adcs.tarea04.programming.exercise.tar...
tarea04 (test.java.adcs.tarea04.programming.exercise) 218 ms
ProgrammingExercise21Test 0 ms
testCalculateInterquartileRange 0 ms
testCalculateMedian 0 ms
calculateMedian 0 ms
testCalculateMean 0 ms
testCalculateStandardDeviation 0 ms
calculatePercentile 0 ms
ProgrammingExercise22Test 16 ms
testToRoman 16 ms
testToRomanNegativeNumber 0 ms
ProgrammingExercise23Test 62 ms
testReadFromFile 31 ms
testReadFromNotExistingFile 0 ms
testReadEmptyListFromFile 15 ms
testReadWrongTextFormatFromFile 16 ms
ProgrammingExercise24Test 140 ms
testSaveRecordsInFile 125 ms
testCreateRecord 0 ms
testLoadRecordsFromFile 15 ms
testLookUpRecordFromFile 0 ms
testSearchNotExistingRecordFromFile 0 ms
testLoadRecordsFromNotExistingFile 0 ms
testSearchRecordFromFile 0 ms

Programming Exercises21




MCCRepository C:\MCCRepository
.idea
lib
out
src
main.java.adcs
test.java.adcs
tarea03.programming.exercise.tarea03
tarea04.programming.exercise.tarea04
a01 100% classes, 100% lines covered
ProgrammingExercise21Test 100% methods, 100% lines covered
b02
c03
d04
MCCRepository.iml
External Libraries
Scratches and Consoles


Run:
ProgrammingExercise21Test
ProgrammingExercise21Test (test.java.adcs.tareaC 15 ms
testCalculateInterquartileRange 15 ms
testCalculateMedian 0 ms
calculateMedian 0 ms
testCalculateMean 0 ms
testCalculateStandardDeviation 0 ms
calculatePercentile 0 ms

ProgrammingExercise21Test.java

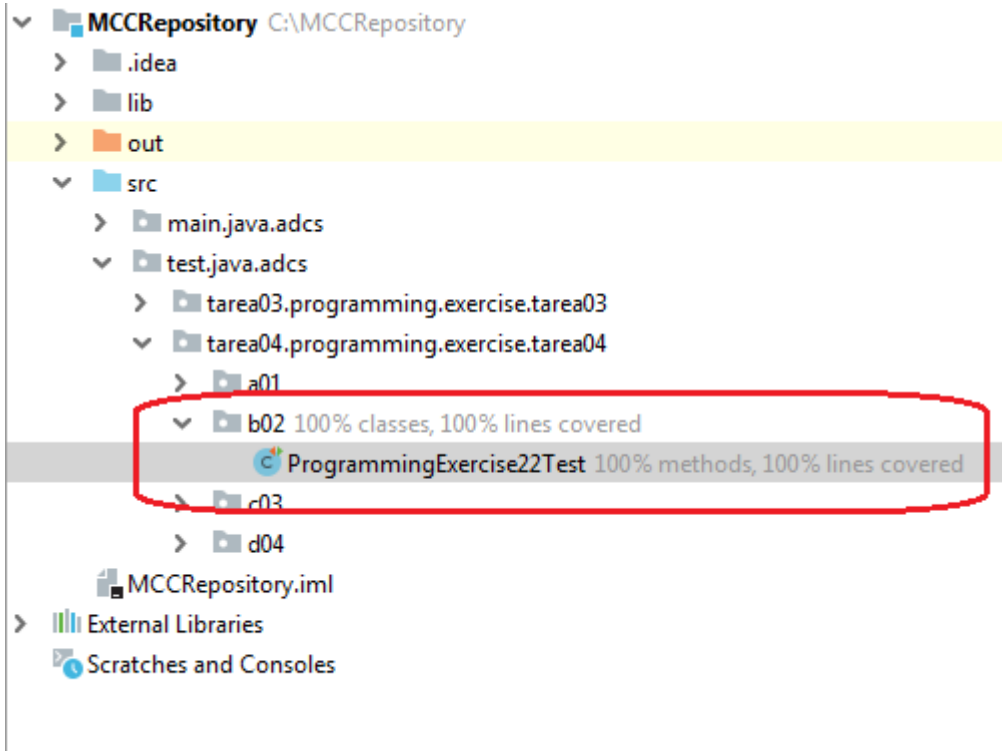
```java
package test.java.adcs.tarea04.programming.exercise.tarea04.a01;

import main.java.adcs.tarea01.programming.exercise.h08.ProgrammingExercise08;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.assertEquals;

/**
 * This class is the  Programming Exercise 21 used to test the
 * Programming Exercise 08
 */
public class ProgrammingExercise21Test {

    private ProgrammingExercise08 exercise08;
    private Double [] numbers;

    @Before
    public void setup(){
        exercise08 = new ProgrammingExercise08();
        numbers = new Double[]{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0};
    }

    @Test
    public void testCalculateMean(){
        Double mean = exercise08.calculateMean(numbers);
        Double expected = 5.5d;
        assertEquals(mean, expected);
    }
```

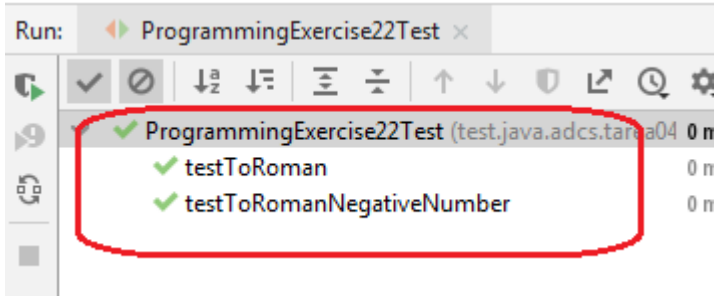
ProgrammingExercise21Test.java

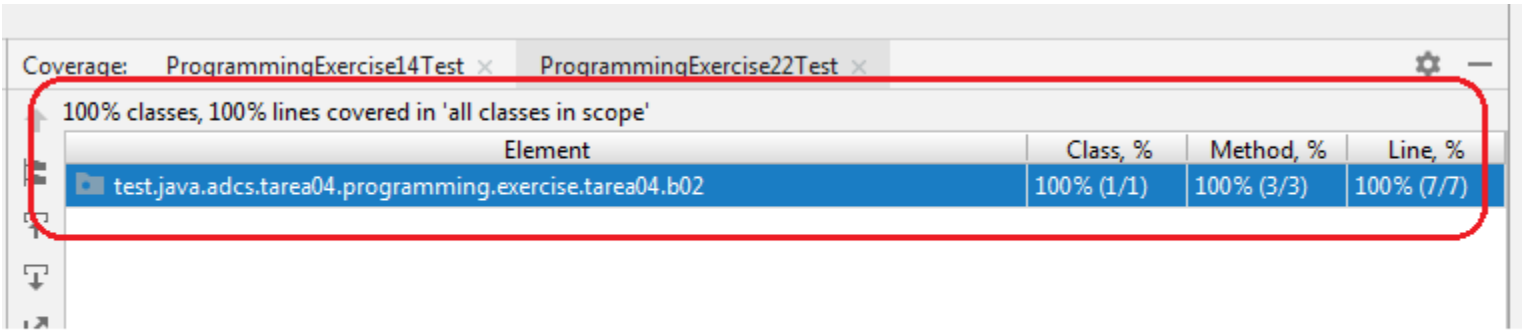
```java
    }

    @Test
    public void testCalculateStandardDeviation(){
        Double standardDeviation = exercise08.calculateStandardDeviation(numbers);
        Double expected = 2.6076809620810595d;
        assertEquals(standardDeviation, expected);
    }

    @Test
    public void testCalculateMedian(){
        Double median = exercise08.calculateMedian(numbers);
        Double expected = 5.5d;
        assertEquals(median, expected);
    }

    @Test
    public void testCalculateInterquartileRange(){
        Double interquartileRange = exercise08.calculateInterquartileRange(numbers);
        Double expected = 6.0d;
        assertEquals(interquartileRange, expected);
    }

    @Test
    public void calculateMedian(){
        Integer median = exercise08.median( l: 2, r: 6);
        Integer expected = 4;
        assertEquals(median, expected);
    }

```

```
57
58      @Test
59      public void calculatePercentile(){
60          Double percentile = exercise08.calculatePercentile( percentile: 20, numbers);
61          Double expected = 2.0d;
62          assertEquals(percentile, expected);
63      }
64
65  }
66
```

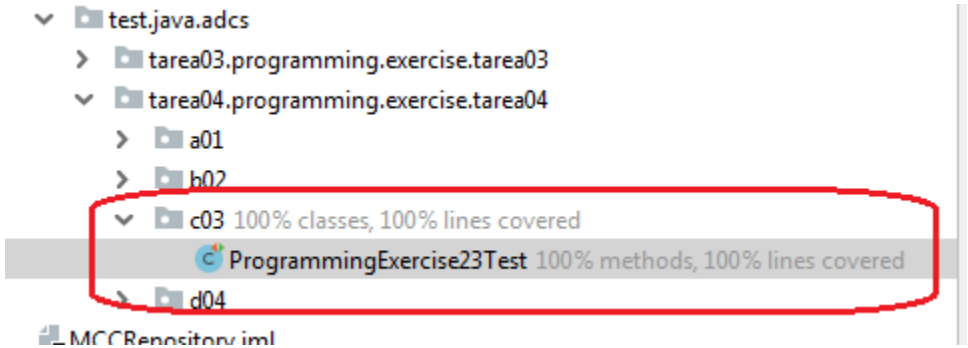
Programming Exercises22






Coverage:
ProgrammingExercise14Test
ProgrammingExercise22Test
100% classes, 100% lines covered in 'all classes in scope'
Element
Class, %
Method, %
Line, %
test.java.adcs.tarea04.programming.exercise.tarea04.b02
100% (1/1)
100% (3/3)
100% (7/7)

Programming Exercises23



ProgrammingExercise23Test.java

```java
package test.java.adcs.tarea04.programming.exercise.tarea04.c03;

import main.java.adcs.tarea02.programming.exercise.a01.MyPowerList;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.List;
import static org.junit.Assert.assertEquals;

/**
 * This class is the  Programming Exercise 23 used to test the
 * Programming Exercise 14
 */
public class ProgrammingExercise23Test {

    private MyPowerList list;
    private String fileName;

    @Before
    public void setup(){
        list = new MyPowerList();
        fileName = "C:\\temp\\output.txt";
        new File(fileName).delete();
    }

    @Test(expected = FileNotFoundException.class)
    public void testReadFromNotExistingFile() throws IOException {
        list.readPeopleFromFile(fileName); }

```

```java
33      @Test
34      public void testReadFromFile() throws IOException {
35          list.add("100");
36          list.add("101");
37          list.add("102");
38          list.saveToFile();
39          List<String[]> lines = list.readFromFile(fileName);
40          String[] record = null;
41          for(int i=0; i < lines.size();i++){
42              record = lines.get(i);
43              for(String item :record){
44                  item = item.replace( target: "[", replacement: "").replace( target: "]", replacement: "").trim();
45              }
46          }
47          assertEquals(lines.size(),  actual: 1);
48          assertEquals(record.length,  actual: 3);
49      }
```

```java
51      @Test
52      public void testReadWrongTextFormatFromFile() throws IOException {
53          list.add("a");
54          list.add("jfk");
55          list.add("XHGC");
56          list.add("8");
57          list.saveToFile();
58          List<String[]> lines = list.readFromFile(fileName);
59          String[] record = null;
60          for(int i=0; i < lines.size();i++){
61              record = lines.get(i);
62              for(String item :record){
63                  item = item.replace( target: "[", replacement: "").replace( target: "]", replacement: "").trim();
64              }
65          }
66          assertEquals(lines.size(),  actual: 1);
67          assertEquals(record.length,  actual: 4);
68      }
```

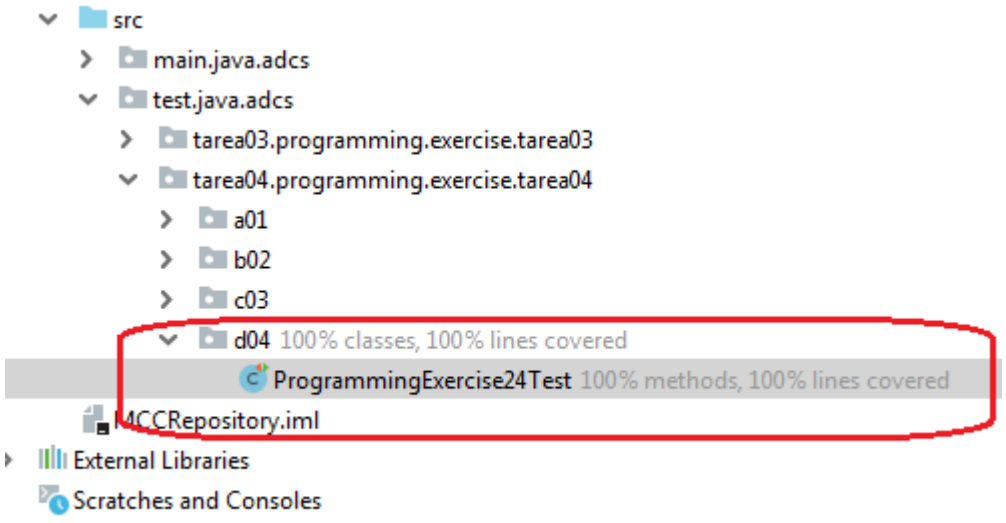
```java
69
70      @Test
71      public void testReadEmptyListFromFile() throws IOException {
72          list.clear();
73          list.saveToFile();
74          List<String[]> lines = list.readFromFile(fileName);
75          String[] record = null;
76          for(int i=0; i < lines.size();i++){
77              record = lines.get(i);
78              for(String item :record){
79                  item = item.replace( target: "[", replacement: "").replace( target: "]", replacement: "").trim();
80              }
81          }
82          assertEquals(lines.size(),  actual: 1);
83          assertEquals(record.length,  actual: 1);
84      }
85
86      @After
87      public void cleanUp() { new File(fileName).delete(); }
90
91  }
92
```

Coverage: ProgrammingExercise14Test × ProgrammingExercise23Test ×

100% classes, 100% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
test.java.adcs.tarea04.programming.exercise.tarea04.c03	100% (1/1)	100% (6/6)	100% (46/46)

Programming Exercises24



ProgrammingExercise24Test.java

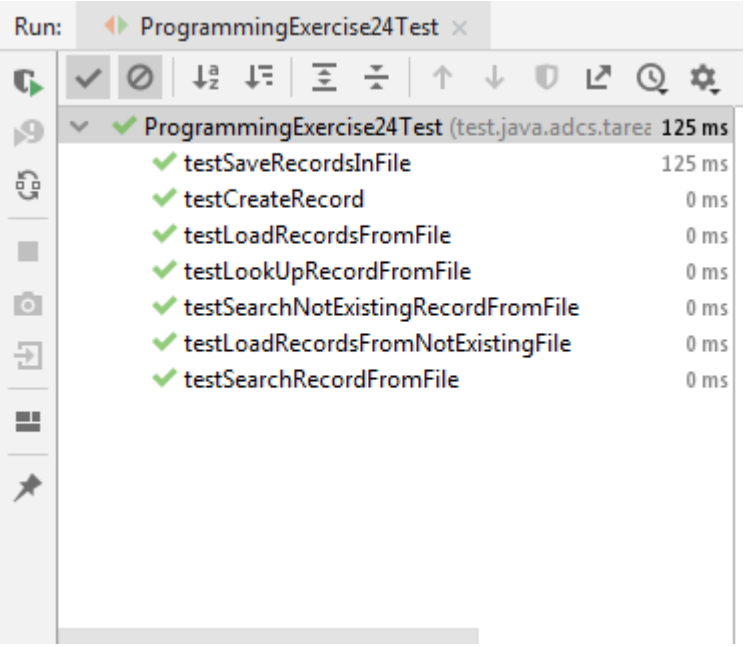
```java
package test.java.adcs.tarea04.programming.exercise.tarea04.d04;

import main.java.adcs.tarea02.programming.exercise.a01.MyPowerList;
import main.java.adcs.tarea02.programming.exercise.f06.Person;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.List;
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

/**
 * This class is the  Programming Exercise 24 used to test the
 * Programming Exercise 15
 */
public class ProgrammingExercise24Test {
    private static final String AGENDA_ARTISTAS_Y_ACTRICES_CINE_MEXICANO = "C:\\temp\\personajesCineMexicano.txt";
    MyPowerList<Person> agenda;

    @Before
    public void setup() { agenda = new MyPowerList(); }

    @Test
    public void testCreateRecord(){
        agenda.add(new Person( name: "Juan Camaney",  address: "Calle de la viuda sin numero",  phone: "1234567890",  email: "juan.camaney@cinedeoro.com.mx"));
        agenda.add(new Person( name: "Mauricio Garces",  address: "Calle - modisto de señoras copetonas",  phone: "9876543210",  email: "modisto.de.rucas@eltodasmias.com"));
        assertEquals(agenda.size(),  actual: 2);
    }

```

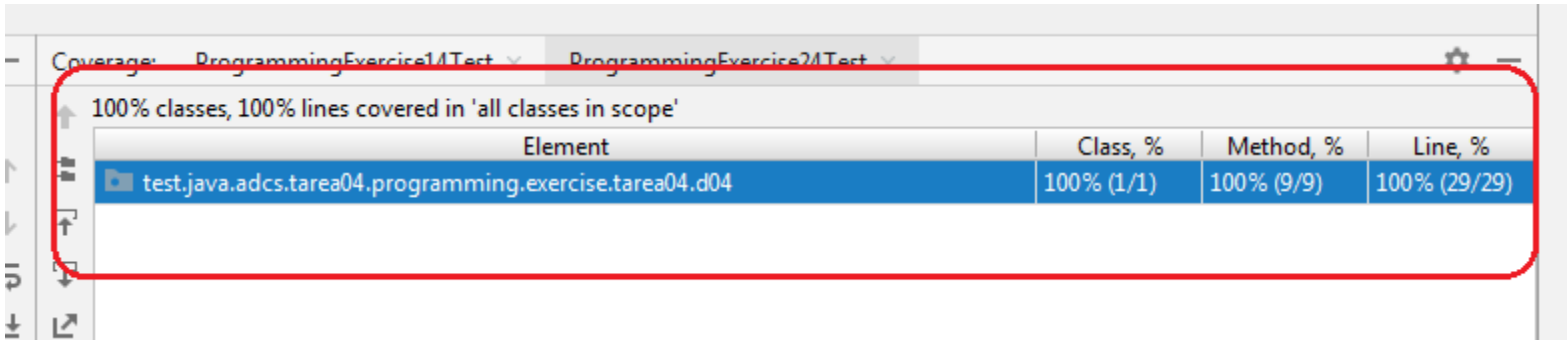
ProgrammingExercise24Test.java

```java
    }

    @Test
    public void testSaveRecordsInFile(){
        agenda.add(new Person( name: "Juan Camaney",  address: "Calle de la viuda sin numero",  phone: "1234567890",  email: "juan.camaney@cinedeoro.com.mx"));
        agenda.add(new Person( name: "Mauricio Garces",  address: "Calle - modisto de señoras copetonas",  phone: "9876543210",  email: "modisto.de.rucas@eltodasmias.com"));
        agenda.saveToFile(AGENDA_ARTISTAS_Y_ACTRICES_CINE_MEXICANO);
        assertTrue(new File(AGENDA_ARTISTAS_Y_ACTRICES_CINE_MEXICANO).exists());
    }
```

```java
    @Test
    public void testLoadRecordsFromFile() throws IOException {
        agenda.readPeopleFromFile(AGENDA_ARTISTAS_Y_ACTRICES_CINE_MEXICANO);
    }

    @Test(expected = FileNotFoundException.class)
    public void testLoadRecordsFromNotExistingFile() throws IOException {
        agenda.readPeopleFromFile( fileName: "C:\\unknownFile.txt"); }

    @Test
    public void testSearchRecordFromFile() throws IOException {
        agenda.add(new Person( name: "Said Olano",  address: "Av Aviacion sin Numero",  phone: "9876543210",  email: "josesaid@gmail.com"));
        agenda.saveToFile(AGENDA_ARTISTAS_Y_ACTRICES_CINE_MEXICANO);
        List<Person> people = agenda.findByName( nameToSearch: "Said Olano");
        assertEquals(people.iterator().next().getName(),  actual: "Said Olano");
    }

    @Test
    public void testSearchNotExistingRecordFromFile(){
        List<Person> people = agenda.findByName( nameToSearch: "Juana la cubana");
        assertEquals(people.size(),  actual: 0);
    }
```

```java
    @Test
    public void testLookUpRecordFromFile(){
        agenda.add(new Person( name: "Mauricio Garces",  address: "Calle - modisto de señoras copetonas",  phone: "9876543210",  email: "modisto.de.rucas@eltodasmias.com"));
        List<Person> people = agenda.findByName( nameToSearch: "Mauricio Garces");
        assertEquals(people.get(0).getPhone(),  actual: "9876543210");
    }

    @After
    public void cleanUp() { agenda.clear(); }
}
```


Run:
ProgrammingExercise24Test
ProgrammingExercise24Test (test.java.adcs.tarea 125 ms
testSaveRecordsInFile 125 ms
testCreateRecord 0 ms
testLoadRecordsFromFile 0 ms
testLookUpRecordFromFile 0 ms
testSearchNotExistingRecordFromFile 0 ms
testLoadRecordsFromNotExistingFile 0 ms
testSearchRecordFromFile 0 ms


Coverage:
ProgrammingExercise14Test
ProgrammingExercise24Test
100% classes, 100% lines covered in 'all classes in scope'
Element
Class, %
Method, %
Line, %
test.java.adcs.tarea04.programming.exercise.tarea04.d04
100% (1/1)
100% (9/9)
100% (29/29)