Grado en Ciencia, Gestión e Ingeniería de Servicios

Introducción a la Programación – Ejercicios

Índice

EXERCISE 1 - Gravity Calculator

In this assignment, you will create a program that computes the distance an object will fall in Earth's gravity.

Part One

1. Open the corresponding project in Codeboard.io: https://codeboard.io/projects/347144?ltiSessionId=11040097&ltiUserId=19131&ltiNonce=8699b568ea7043e6da5adcdd4cec5e9e

2. Check the following code is already included in the Main class

```java
class GravityCalculator {
    public static void main(String[] arguments) {
        double gravity = -9.81;  // Earth's gravity in m/s^2
        double initialVelocity = 0.0;
        double fallingTime = 10.0;
        double initialPosition = 0.0;
        double finalPosition = 0.0;
        System.out.println("The object's position after " + fallingTime +
                " seconds is " + finalPosition + " m.");
    }
}
```

3. Compile and Execute the program in Codeboard using the Compile and Run buttons of Codeboard.

What is the output of the unmodified program? Use the text edit you will find in the corresponding task in Aula Virtual to write down such output.

Part Two

Modify the example program to compute the position of an object after falling for 10 seconds, outputting the position in meters. The formula in Math notation is:

$x(t) = 0.5 \times at^2 + v_i t + x_i$

Variable	Meaning	Value
a	Acceleration (m/s^2)	-9.81
t	Time (s)	10
v_i	Initial velocity (m/s)	0
x_i	Initial position	0

Note: The correct value is -490.5 m. Java will output more digits after the decimal place, but that is unimportant.

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

EXERCISE 2 – Comparator

In this assignment, you will create a program that compares two numbers read from the input and outputs a message reporting the result.

1. Open the corresponding Project in Codeboard.io: https://codeboard.io/projects/274830?ltiSessionId=11040101&ltiUserId=19131&ltiNonce=c6c12c60f37d377c51c8b851198f2c39

2. In the main method, add the code needed to:

   a. Declare two integer variables (num1, num2).
   b. Print a message asking the user to enter a number.
   c. Set the value of num1 to the number read from the screen.
   d. Print a message asking the user to enter another number.
   e. Set the value of num2 to the new input read from the screen.
   f. Check whether num1 is equal, lower or higher than num2. (tip: use a combination of `if - else` statements).
   g. According to the result of the diferente comparisons, print a message similar to "*num1* is *xxx* than *num2*".
3. Run the program a number of times, checking that it works properly with diferent pairs of inputs.

(NOTE: to enter input in Codeboard.io use the text box at the bottom of the screen).

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

EXERCISE 3 – Time Conversor

In this assignment, you will create a program that takes as input a number of seconds and outputs its equivalent in terms of hours, minutes and seconds.

1. Open the corresponding Project in Codeboard.io: https://codeboard.io/projects/276718?ltiSessionId=11040105&ltiUserId=19131&ltiNonce=0b8e722389a8f24027ee925589dafc1c

2. In the main method, add the code needed to:

   a. Use the Scanner class to read the number of seconds entered by the user.
   b. Implement the following algorithm:

```
Read seconds
minutes = seconds / 60
seconds = seconds - (minutes * 60)
hours = minutes / 60
minutes = minutes - (hours * 60)
Write hours, minutes, seconds
```

(NOTE: to enter input in Codeboard.io use the text box at the bottom of the screen).

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

Exercise 4 - Foo Corporation

Foo Corporation needs a program to calculate how much to pay their hourly employees. The US Department of Labor requires that employees get paid time and a half for any hours over 40 that they work in a single week. For example, if an employee works 45 hours, they get 5 hours of overtime, at 1.5 times their base pay. The State of Massachusetts requires that hourly employees be paid at least $8.00 an hour. Foo Corp requires that an employee not work more than 60 hours in a week.

Summary of Rules

- An employee gets paid (hours worked) × (base pay), for each hour up to 40 hours.
- For every hour over 40, they get overtime = (base pay) × 1.5.
- The base pay must not be less than the minimum wage ($8.00 an hour). If it is, print an error.
- If the number of hours is greater than 60, print an error message.

Directions

Write a method that takes the base pay and hours worked as parameters, and prints the total pay or an error. Write a `main` method that calls this method for each of these employees:

	Base Pay	**Hours Worked**
Employee 1	$7.50	35
Employee 2	$8.20	47
Employee 3	$10.00	73

Hint

Do *not* try to write the entire program in one go. It is much easier to write a small piece and test it, then write another small piece and test it. For example, start by writing just a skeleton of your method and your main program. Then add the code to do the normal salary computation, without any special rules. Then add each additional rule, one at a time. You should test your program with simple test inputs to check that you handle each case.

Good luck!

EXERCISE 5 – Print Number Series

In this assignment, you will create a program that prints the following sets of numbers:

1. 7 to 77 increasing by 7 each iteration
2. 2, 5, 8, 11, 14, 17, 20
3. 99, 88, 77, 66, 55, 44, 33, 22 11, 0

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

Exercise 6 - Maraton

A group of MIT friends decide to run the Boston Marathon. Their names and times (in minutes) are below:

Name	Time (minutes)
Elena	341
Thomas	273
Hamilton	278
Suzie	329
Phil	445
Matt	402
Alex	388
Emma	275
John	243
James	334
Jane	412
Emily	393
Daniel	299
Neda	343
Aaron	317
Kate	265

Problem

Find the fastest runner. Print the name and his/her time (in minutes).

Optional: Find the second fastest runner. Print the name and his/her time (in minutes).

Directions

Write a method that takes as input an array of integers and returns the index corresponding to the person with the lowest time. Run this method on the array of times. Print out the name and time corresponding to the returned index.

Write a second method to find the second-best runner. The second method should use the first method to determine the best runner, and then loop through all values to find the second-best (second lowest) time.

Here is a program skeleton to get started:

```
class Marathon {
    public static void main (String[] arguments) {
        String[] names = {
            "Elena", "Thomas", "Hamilton", "Suzie", "Phil", "Matt", "Alex",
            "Emma", "John", "James", "Jane", "Emily", "Daniel", "Neda",
            "Aaron", "Kate"
        };

        int[] times = {
            341, 273, 278, 329, 445, 402, 388, 275, 243, 334, 412, 393, 299,
            343, 317, 265
        };

        for (int i = 0; i < names.length; i++) {
            System.out.println(names[i] + ": " + times[i]);
        }
    }
}
```

EXERCISE 7 –Strings

In this assignment, you will create a program that prompts user for a String and then calls a a number of methods to process the text entered by the user to later inform about the result through the screen:

Method 1 – parseString

1. If the input has less tan 5 characters, between 5 and 15 characters or more tan 15 characters.
2. If the input starts with ‘a’

For instance, if the input is:

Enter a string: “vereda”

The output should be:

“- The input has between 5 and 15 characters and it does not begin with an ‘a’”

Method 2 – reverseString

Prints the reverse of the String. The output shall look like

For instance, if the input is:

Enter a string: “vereda”

The output should be:

“- The reverse of the string ‘vereda’ is ‘aderev’”

Method 3 – checkVowels

Counts the number of vowels (a, e, i, o, u, A, E, I, O, U) contained in the string, and prints the count.

For instance, if the input is:

Enter a string: “lorem ipsum 45”

The output should be:

“The input has 4 vowels”

- You could use toLowerCase() to convert the input String to lowercase to reduce the number of cases to consider.

Method 4 – checkDigits

Counts the number of digits (0-9) contained in the string, and prints the count.

For instance, if the input is:

Enter a string: “lorem ipsum 45”

The output should be:

“The input has 2 digits”

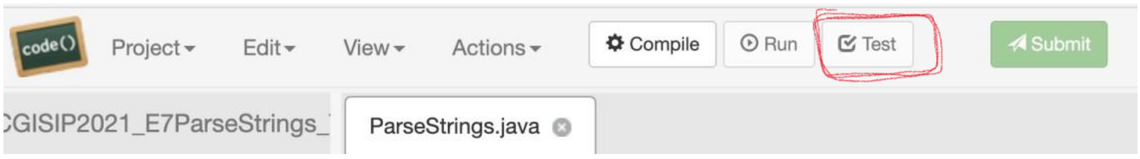
- To check if a char c is a digit, you can use boolean expression (c >= '0' && c <= '9'); or use built-in boolean function Character.isDigit(c).

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

Testing Instructions

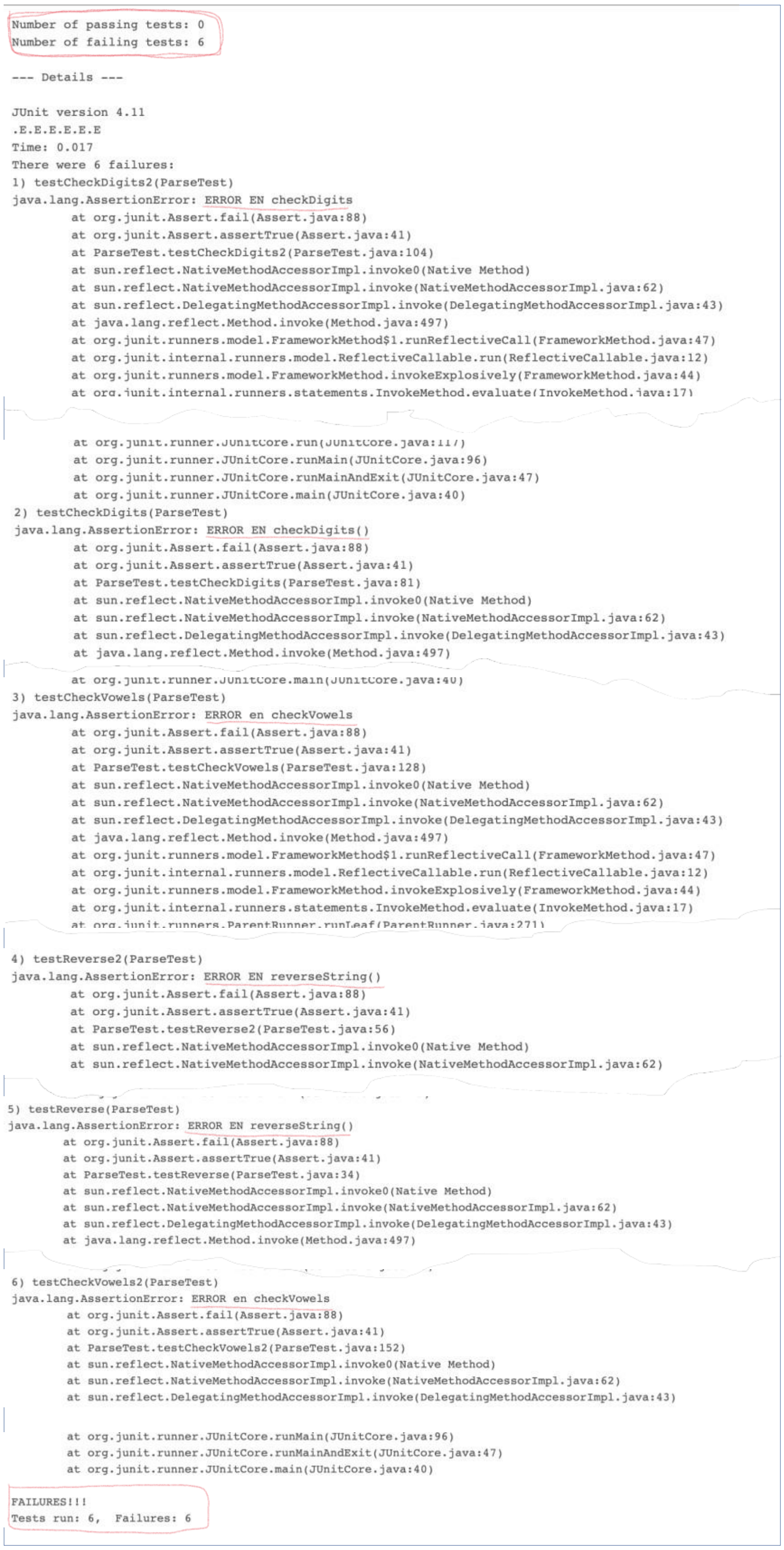
Once you think you have reached a solution, you could try if it works properly without submitting. To that end, click the *Test* button shown in the next figure..



As a result, your program will be run with a different number of test inputs, and the system will inform about the result of those executions, as the following figures illustrate.

First message states that 0 out of the 6 tests run have passed, i.e. the solution has been tried with different inputs and it has not worked properly. More specifically, if we go through the rest of the information on the screen dump we can see that errors have been raised when running the checkDigits() method (2 times), the checkVowels method (2 times) and the reverseString() method (2 times).

Finally, a new summary states again that 6 tests were run and all of them failed. All this given, you should check your code before submission to fix existing problems since, apparently, the solutions you have provided are not working properly.



```
Number of passing tests: 0
Number of failing tests: 6

--- Details ---

JUnit version 4.11
.E.E.E.E.E.E
Time: 0.017
There were 6 failures:
1) testCheckDigits2(ParseTest)
java.lang.AssertionError: ERROR EN checkDigits
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testCheckDigits2(ParseTest.java:104)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:47)
        at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
        at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:44)
        at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)

        at org.junit.runner.JUnitCore.run(JUnitCore.java:117)
        at org.junit.runner.JUnitCore.runMain(JUnitCore.java:96)
        at org.junit.runner.JUnitCore.runMainAndExit(JUnitCore.java:47)
        at org.junit.runner.JUnitCore.main(JUnitCore.java:40)
2) testCheckDigits(ParseTest)
java.lang.AssertionError: ERROR EN checkDigits()
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testCheckDigits(ParseTest.java:81)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)

        at org.junit.runner.JUnitCore.main(JUnitCore.java:40)
3) testCheckVowels(ParseTest)
java.lang.AssertionError: ERROR en checkVowels
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testCheckVowels(ParseTest.java:128)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:47)
        at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
        at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:44)
        at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
        at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:271)

4) testReverse2(ParseTest)
java.lang.AssertionError: ERROR EN reverseString()
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testReverse2(ParseTest.java:56)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

5) testReverse(ParseTest)
java.lang.AssertionError: ERROR EN reverseString()
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testReverse(ParseTest.java:34)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)

6) testCheckVowels2(ParseTest)
java.lang.AssertionError: ERROR en checkVowels
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.assertTrue(Assert.java:41)
        at ParseTest.testCheckVowels2(ParseTest.java:152)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

        at org.junit.runner.JUnitCore.runMain(JUnitCore.java:96)
        at org.junit.runner.JUnitCore.runMainAndExit(JUnitCore.java:47)
        at org.junit.runner.JUnitCore.main(JUnitCore.java:40)

FAILURES!!!
Tests run: 6,  Failures: 6
```

EXERCISE 8/9 – Book/Library

The libraries of SmallTownX need a new electronic rental system, and it is up to you to build it. SmallTownX has two libraries. Each library offers many books to rent. Customers can print the list of available books, borrow, and return books.

Problem

We provide four classes, `Book` and `Library` are the core of the program, that provide the functionality for the book database. `LauncherBook` and `LauncherLibrary` are only in charge of running the program: create objects by instantiating the above classes and use them. You must implement the missing methods in `Book` and `Library` to make these classes work properly.

Step One: Implement book

Open the project “Ejercicio 8 – Book”.

First we need a class to model books. Start by creating a class called `Book` by modifying the one provided.

This class defines methods to get the title of a book, find out if it is available, borrow the book, and return the book.

However, the skeleton that we provide is missing the implementations of the methods. Fill in the body of the methods with the appropriate code.

The `main` method in the `LauncherBook` class tests the methods. When you run the program, the output should be:

```
Title (should be The Da Vinci Code): The Da Vinci Code
Rented? (should be false): false
Rented? (should be true): true
Rented? (should be false): false
```

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

Step Two: Implement Library

Open the project “Ejercicio 9 – Library”.

Copy and paste the code of the `Book` class you have just finished in the `Book.java` file.

Next we need to build the class that will represent each library, and manage a collection of books. All libraries have the same hours: 9 AM to 5 PM daily. However, they have different addresses and book collections (i.e., arrays of `Book` objects).

The `main` method of the `LauncherLibrary` class creates two libraries, then performs some operations on the books. However, all the methods and member variables are missing. You will need to define and implement the missing methods.

To do so, read the `main` method in the `LauncherLibrary` class and look at the compile errors to figure out what methods are missing.

Notes

- Some methods will need to be *static* methods, and some need to be *instance* methods.

- Be careful when comparing Strings objects. Use `string1.equals(string2)` for comparing the contents of `string1` and `string2`.
- You should get a small part working at a time. Start by commenting the entire main method, then uncomment it line by line. Run the program, get the first lines working, then uncomment the next line, get that working, etc.
- You must not modify the main method excepto from commenting/uncommenting lines of code.

The output when you run this program should be similar to the following:

```
Library hours:
Libraries are open daily from 9am to 5pm.

Library addresses:
10 Main St.
228 Liberty St.

Borrowing The Lord of the Rings:
You successfully borrowed The Lord of the Rings
Sorry, this book is already borrowed.
Sorry, this book is not in our catalog.

Books available in the first library:
The Da Vinci Code
Le Petit Prince
A Tale of Two Cities

Books available in the second library:
No book in catalog

Returning The Lord of the Rings:
You successfully returned The Lord of the Rings

Books available in the first library:
The Da Vinci Code
Le Petit Prince
A Tale of Two Cities
The Lord of the Rings
```

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

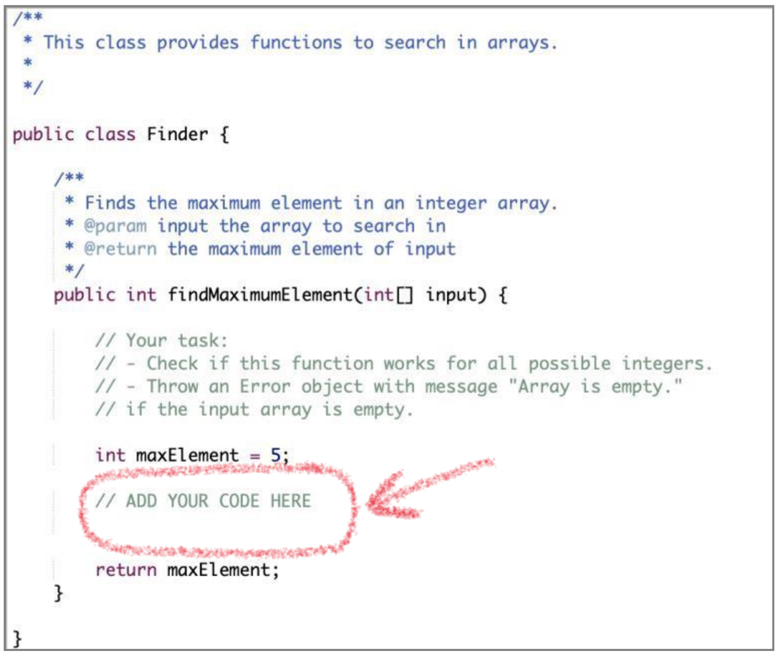
EXERCISE 10 – FindMaxElement

In this assignment, you will create a program that given an array of Integers return the highest value in the array.

Problem

To achieve your objective, you just need to complete one method. Such method receives as input parameter an array of integer values. It then searches for the highest value in the array and returns as output such value.

Find below the skeleton of the class you will get to get started, which includes the method you must complete:

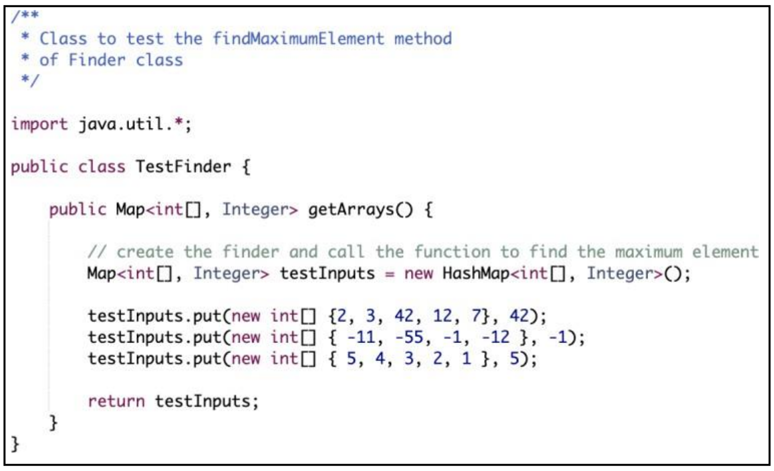


```java
/**
 * This class provides functions to search in arrays.
 *
 */

public class Finder {

    /**
     * Finds the maximum element in an integer array.
     * @param input the array to search in
     * @return the maximum element of input
     */
    public int findMaximumElement(int[] input) {

        // Your task:
        // - Check if this function works for all possible integers.
        // - Throw an Error object with message "Array is empty."
        // if the input array is empty.

        int maxElement = 5;

        // ADD YOUR CODE HERE

        return maxElement;
    }
}
```

Fig. 1. Finder class skeleton

Testing Instructions

In order to run and test your program, the TestFinder class shown below is provided.
Find below the Main class provided:



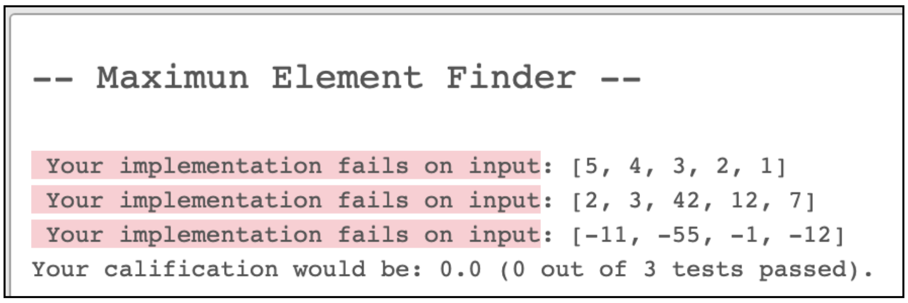
```java
/**
 * Class to test the findMaximumElement method
 * of Finder class
 */

import java.util.*;

public class TestFinder {

    public Map<int[], Integer> getArrays() {

        // create the finder and call the function to find the maximum element
        Map<int[], Integer> testInputs = new HashMap<int[], Integer>();

        testInputs.put(new int[] {2, 3, 42, 12, 7}, 42);
        testInputs.put(new int[] { -11, -55, -1, -12 }, -1);
        testInputs.put(new int[] { 5, 4, 3, 2, 1 }, 5);

        return testInputs;
    }
}
```

Fig. 2. TestFinder class

Once you think you have reached a solution, you could try if it works properly just by hitting the Run button. As a result, your program will be run using as inputs the 3 arrays included in the testInputs shown in the previous figure and compared against the values you indicated as valid outputs. In other words:

- The output of findMaximumElement({2, 3, 42, 12, 7}) will be compared against the 42 value.
- The output of findMaximumElement({ -11, -55, -1, -12 }) will be compared against the -1 value.
- The output of findMaximumElement({ 5, 4, 3, 2, 1 }) will be compared against the 5 value.

The system will inform then about the result of those executions, as the following figure illustrates.



```
-- Maximun Element Finder --

Your implementation fails on input: [5, 4, 3, 2, 1]
Your implementation fails on input: [2, 3, 42, 12, 7]
Your implementation fails on input: [-11, -55, -1, -12]
Your calification would be: 0.0 (0 out of 3 tests passed).
```

Fig. 3. Running output

Apparently, the solution provided has failed to get the maximum element of the 3 arrays used as input. The solution provided has consequently passed 0 out 3 tests and its mark would have been consequently 0.0 points.

You can change the testInputs and/or add more testInputs, just by copy/paste the correspondence sentence. For each sentence you should just provide an array of Integer values along with an Integer value that should be the output method.

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard

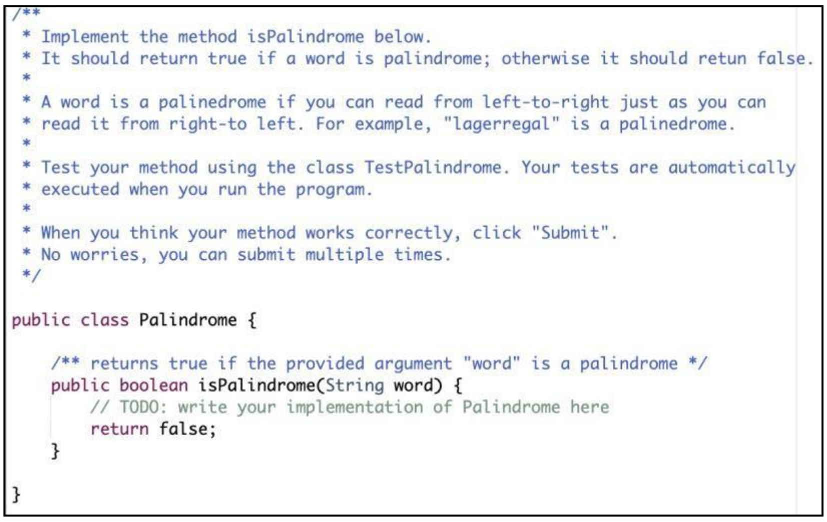
EXERCISE 11 – IsPalindrome

In this assignment, you will create a program that checks whether a word is palindrome.

Problem

To achieve your objective, you just need to complete one method. Such method receives as input a String and returns a Boolean value. Such value will be true if you can read the word from left-to-right just as you can read it from right-to left. For example, "dabalearrozalazorraelabad" is a palinedrome.

Find below the skeleton of the class you will get to get started, which includes the method you must complete:

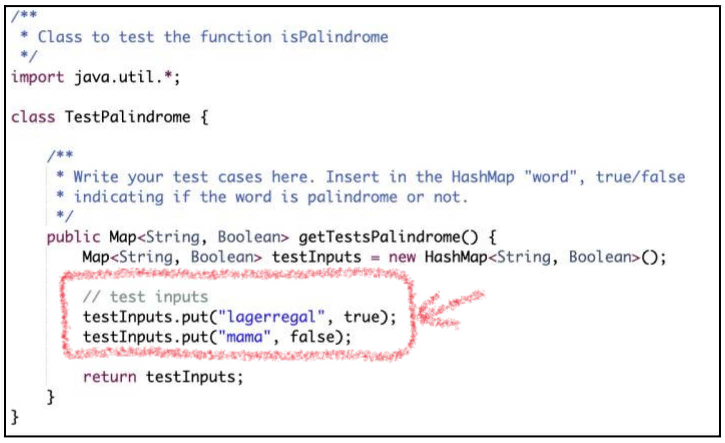


```java
/**
 * Implement the method isPalindrome below.
 * It should return true if a word is palindrome; otherwise it should retun false.
 *
 * A word is a palinedrome if you can read from left-to-right just as you can
 * read it from right-to left. For example, "lagerregal" is a palinedrome.
 *
 * Test your method using the class TestPalindrome. Your tests are automatically
 * executed when you run the program.
 *
 * When you think your method works correctly, click "Submit".
 * No worries, you can submit multiple times.
 */

public class Palindrome {

    /** returns true if the provided argument "word" is a palindrome */
    public boolean isPalindrome(String word) {
        // TODO: write your implementation of Palindrome here
        return false;
    }

}
```

Fig. 1. Palindrome class skeleton

Testing Instructions

In order to run and test your program, the TestPalindrome class shown below is provided:



```java
/**
 * Class to test the function isPalindrome
 */
import java.util.*;

class TestPalindrome {

    /**
     * Write your test cases here. Insert in the HashMap "word", true/false
     * indicating if the word is palindrome or not.
     */
    public Map<String, Boolean> getTestsPalindrome() {
        Map<String, Boolean> testInputs = new HashMap<String, Boolean>();

        // test inputs
        testInputs.put("lagerregal", true);
        testInputs.put("mama", false);

        return testInputs;
    }
}
```

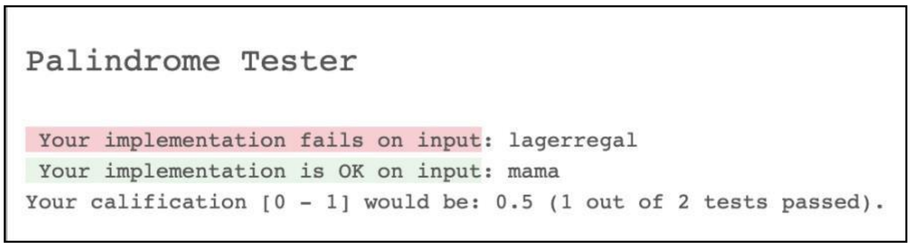
Fig. 2. TestPalindrome class

Once you think you have reached a solution, you could try if it works properly just by hitting the Run button. As a result, the method will be executed a couple of times, and the results will be compared against the Boolean values accompanying each word included in the testInputs. In other words:

- The output of isPalindrome("lagerregal") will be compared against the true value.

- The output of isPalindrome(“mama”) will be compared against the false value.

The system will inform then about the result of those executions, as the following figure illustrates.



```
Palindrome Tester

Your implementation fails on input: lagerregal
Your implementation is OK on input: mama
Your calification [0 - 1] would be: 0.5 (1 out of 2 tests passed).
```

Fig. 3. Running output

Apparently, the solution provided has failed to identify whether “lagerregal” is a palindrome and it has correctly identified that “mama” is not a palindrome. The solution provided has consequently passed 1 out of 2 tests and thus its mark would have been 0.5 points (marks are given in the [0 – 1] range).

You can change the words used as input and/or add more words to test, just by copy/paste the correspondence sentences. For each sentence you should just provide a pair of (Word, Boolean) values so that the Boolean value informs about whether the Word value is or not a palindrome.

Submission Instructions

Once you have finished submit your solution using the Submit button of Codeboard. Keep in mind you can make any number of submissions. The last one is the one that will be persisted in the system. In other words, the last submission overwrites previous submission.