import java.util.Arrays;

public class MinMaxFinder {

public void main(String[] args) {

findMinMax(new int[] {56,34,7,3,54,3,34,34,53});

}

public int[] findMinMax(int[] is) {

Arrays.sort(is);

int min = is[0];

int max = is[is.length-1];

int res[]= {min,max};

System.out.println("(Minimum value index 0,Maximum value index 1) : "+res[0]+","+res[1]);

return res;

}

}

```

#### Testcases class

```sh

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Arrays;

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.TestInstance;

import org.junit.jupiter.api.TestInstance.Lifecycle;

@TestInstance(Lifecycle.PER\_CLASS)

class MinMaxFinderTest {

MinMaxFinder minmax;

int[] arr1 = {56,34,7,3,58,3,34,1,53};

int[] arr2 = {56,37,7,99,31,34,8,10};

int[] arr3 = {11,89,29,121,12,55};

int[] expectedval1 = {1,58};

int[] expectedval2 = {7,99};

int[] expectedval3 = {11,121};

@BeforeAll

void init() {

minmax = new MinMaxFinder();

}

@Test

void arr1testcase() {

fun(arr1,expectedval1);

fun(arr2,expectedval2);

fun(arr3,expectedval3);

}

private void fun(int[] arr, int[] expectedval) {

// TODO Auto-generated method stub

int[] actualval = minmax.findMinMax(arr);

assertNotNull(Arrays.toString(actualval),"No data Found");

assertEquals(Arrays.toString(expectedval),Arrays.toString(actualval));

assertArrayEquals(expectedval ,actualval,"mismatch");

assertTrue(actualval.length==2);

assertTrue(actualval[0]<actualval[1]);

}

}

```

#### Output

```sh

(Minimum value index 0,Maximum value index 1) : 1,58

(Minimum value index 0,Maximum value index 1) : 7,99

(Minimum value index 0,Maximum value index 1) : 11,121

```

![screenshots](https://github.com/sayuyagami/Training\_assignments/blob/master/screenshots/Maxmintestcases.png)

>2.Modify the above method to return a single object representing min and max value of the pass array. Define new sets of Junít Test cases of this modified method.

#### Main class

```sh

import java.util.Arrays;

public class Objectarray {

public void main(String[] args) {

findMinMax(new int[] {56,34,7,3,54,3,34,34,53});

}

public Object[] findMinMax(int[] is) {

// TODO Auto-generated method stub

Arrays.sort(is);

int min = is[0];

int max = is[is.length-1];

Object res[]= {min,max};

System.out.println("(Minimum value,Maximum value) : "+Arrays.toString(res));

return res;

}

}

```

#### Testcase class

```sh

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Arrays;

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.TestInstance;

import org.junit.jupiter.api.TestInstance.Lifecycle;

@TestInstance(Lifecycle.PER\_CLASS)

class ObjectarrayTest {

Objectarray objminmax;

int[] arr1 = {59\*2,34,11\*2,72,52,9,34-2,18,55};

int[] arr2 = {5622,3737-18,1897+1,9929,2456,1720,2898/2,2001};

int[] arr3 = {11000,89123,29890,12100,12567,55110};

Object expectedval1[] = {9,118};

Object expectedval2[] = {1449,9929};

Object expectedval3[] = {11000,89123};

@BeforeAll

void init() {

objminmax = new Objectarray();

}

@Test

void objtestcase1() {

fun(arr1,expectedval1);

fun(arr2,expectedval2);

fun(arr3,expectedval3);

}

private void fun(int[] arr, Object[] expval) {

Object actualval[] = objminmax.findMinMax(arr);

assertNotNull(actualval);

assertNotSame(expval,actualval);//varies in object reference

assertEquals(Arrays.toString(expval),Arrays.toString(actualval));

assertIterableEquals(Arrays.asList(expval),Arrays.asList(actualval));

}

}

```