SortArray.java

SortArray.java (%USERPROFILE%\Documents\CSE-337\Assignments\Assignment 3 Implementation\src\so

```java
package sort;

import java.util.Arrays;

/**
* Rachel Glomski
* CSE 337 - Assignment 3
*
* This class has the ability to sort a given array in ascending
* or descending order
*/

public class SortArray{


    public static void main(String[] args){
        SortArray sortArray = new SortArray();
    }


    public int[] ascendingSort(int[] arr){
        Arrays.sort(arr); //built-in ascending sorting
        return arr;
    }

    public int[] descendingSort(int[] arr){
        Arrays.sort(arr);
        //steps into array and swaps first and last element until
        //reaching the middle, reversing the array
        for(int i = 0; i < arr.length/2; i++){
            int tempSort = arr[i];
            arr[i] = arr[arr.length-1-i];
            arr[arr.length-1-i] = tempSort;
        }
        return arr;
    }
}
```

SortArrayTest.java

File Edit Search Markers Folding View Utilities Macros Plugins Help

SortArrayTest.java (%USERPROFILE%\Documents\CSE-337\Assignments\Assignment 3 Implementa

```java
package sort;

import org.junit.Test;
import static org.junit.Assert.*;
import static org.junit.Assert.fail;

public class SortArrayTest{

//Tests a positive range of numbers for an Ascending Array
@Test
    public void testPositiveRangeAsc(){
        int[] testArray = {5, 64, 4500, 1, 2, 3};
        int[] expArrayAsc = {1, 2, 3, 5, 64, 4500};

        SortArray sortArray = new SortArray();

        int[] testArrayAsc = sortArray.ascendingSort(testArray);
        assertArrayEquals(expArrayAsc, testArrayAsc);
    }

//Tests a positive range of numbers for a Descending Array
@Test
    public void testPositiveRangeDes(){
        int[] testArray = {5, 64, 4500, 1, 2, 3};
        int[] expArrayDes = {4500, 64, 5, 3, 2, 1};

        SortArray sortArray = new SortArray();

        int[] testArrayDes = sortArray.descendingSort(testArray);
        assertArrayEquals(expArrayDes, testArrayDes);
    }

//Tests a negative range of numbers for an Ascending Array
@Test
    public void testNegativeRangeAsc(){
        int[] testArray = {-20,-250,-5,-15,-1,-2};
        int[] expArrayAsc = {-250,-20,-15,-5,-2,-1};

        SortArray sortArray = new SortArray();

        int[] testArrayAsc = sortArray.ascendingSort(testArray);
        assertArrayEquals(expArrayAsc, testArrayAsc);
    }
```

25,31

(continued)

```java
//Tests a negative range of numbers for a Descending Array
@Test
    public void testNegativeRangeDes(){
        int[] testArray = {-20,-250,-5,-15,-1,-2};
        int[] expArrayDes = {-1,-2,-5,-15,-20,-250};

        SortArray sortArray = new SortArray();

        int[] testArrayDes = sortArray.descendingSort(testArray);
        assertArrayEquals(expArrayDes, testArrayDes);
    }

//Tests a negative and positive range of numbers for an Ascending Array
@Test
    public void testBothAsc(){
        int[] testArray = {-5, 3, 3, -2, 0, 1};
        int[] expArrayAsc = {-5, -2, 0, 1, 3, 3};

        SortArray sortArray = new SortArray();

        int[] testArrayAsc = sortArray.ascendingSort(testArray);
        assertArrayEquals(expArrayAsc, testArrayAsc);
    }

//Tests a negative and positive range of numbers for a Descending Array
@Test
    public void testBothDes(){
        int[] testArray = {-5, 3, 3, -2, 0, 1};
        int[] expArrayDes = {3, 3, 1, 0, -2, -5};

        SortArray sortArray = new SortArray();

        int[] testArrayDes = sortArray.descendingSort(testArray);
        assertArrayEquals(expArrayDes, testArrayDes);
    }
}
```