CIS 285

Lab 10

Jordan Stebner-Hoang

1. Unit testing
2. Initial source code of SelectionSort.java

public class SelectionSort {

    private int temp;

    /\*\* Creates a new instance of SelectionSort \*/

       public SelectionSort() {

       }

       /\* A simple SelectionSort algorithm

        \* pre-condition:

        \* post-condition:

        \* inputs:

        \* outputs:

        \* special conditions:

        \*/

       public int[] basicSelectionSort(int[] x) {

           for (int i = 0; i < x.length; ++i) {

               for (int j= i+1; j < x.length; ++j) {

                   if (x[i] > x[j]) {

                       temp = x[i];

                       x[i] = x[j];

                       temp = x[j];

                   }

               } // end of inner for loop

           } // end of outer for loop

           return x;

       } // end of basicSelectionSort method

   }

1. Source code of unit tests

import static org.junit.Assert.\*;

import org.junit.Test;

public class testSelectionSort {

    @Test

    public void test() {

        testPositives();

        testNegatives();

        testMixed();

        testDuplicates();

    }

    // 1) testPositives – testing a list of all positive integers

    public void testPositives() {

        int expected[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        int nums[] = {4, 1, 6, 7, 5, 8, 9, 3, 2};

        SelectionSort sorter = new SelectionSort();

        nums = sorter.basicSelectionSort(nums);

        assertArrayEquals(expected, nums);

    }

    // 2) testNegatives – testing a list of all negative integers

    public void testNegatives() {

        int expected[] = {-9, -8, -7, -6, -5, -4, -3, -2, -1};

        int nums[] = {-4, -1, -6, -7, -5, -8, -9, -3, -2};

        SelectionSort sorter = new SelectionSort();

        nums = sorter.basicSelectionSort(nums);

        assertArrayEquals(expected, nums);

    }

    // 3) testMixed – testing a list containing positive, negative and zeros.

    public void testMixed() {

        int expected[] = {-9, -7, -5, -3, -1, 2, 4, 6, 8};

        int nums[] = {4, -7, -5, -1, -9, -3, 6, 2, 8};

        SelectionSort sorter = new SelectionSort();

        nums = sorter.basicSelectionSort(nums);

        assertArrayEquals(expected, nums);

    }

    // 4) testDuplicates  –  testing  a  list  containing  one  or  more  duplicate

    //                       number, for both positive and negative numbers

    public void testDuplicates() {

        int expected[] = {-9, -8, -7, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4};

        int nums[] = {-4, -1, -6, 1, -7, -5, 3, -8, 2, -9, -3, 4, -2};

        SelectionSort sorter = new SelectionSort();

        nums = sorter.basicSelectionSort(nums);

        assertArrayEquals(expected, nums);

    }

}

1. Outputs of unit tests

Test 1 (positives and negative)

%TESTS 1,test(testSelectionSort)

%FAILED 1,test(testSelectionSort)  
%TRACES   
arrays first differed at element [1]; expected:<2> but was:<1>

%TRACEE

%TESTE 1,test(testSelectionSort)  
%RUNTIME14

Changed “temp = x[j]” at line 20 to “x[j] = temp”, Test 2 (positives and negatives)

All tests passed

Test 3 (mixed and duplicates, using change from test 2)

All tests passed

1. Final outputs of all unit tests

All tests passed

1. Final source code of SelectionSort.java

public class SelectionSort {

    private int temp;

    /\*\* Creates a new instance of SelectionSort \*/

    public SelectionSort() {

    }

    /\* A simple SelectionSort algorithm

    \* pre-condition:

    \* post-condition:

    \* inputs:

    \* outputs:

    \* special conditions:

    \*/

    public int[] basicSelectionSort(int[] x) {

        for (int i = 0; i < x.length; ++i) {

            for (int j= i+1; j < x.length; ++j) {

                if (x[i] > x[j]) {

                    temp = x[i];

                    x[i] = x[j];

                    x[j] = temp;

                }

            } // end of inner for loop

        } // end of outer for loop

        return x;

    } // end of basicSelectionSort method

}

1. Git for configuration management

<https://github.com/jstebner/lab10>

* 1. Screenshots of Github

Graphical user interface, application

Description automatically generated

* 1. Local git folder

Text

Description automatically generated with medium confidence