Swinburne University of Technology

COS20007 Object Oriented Programming

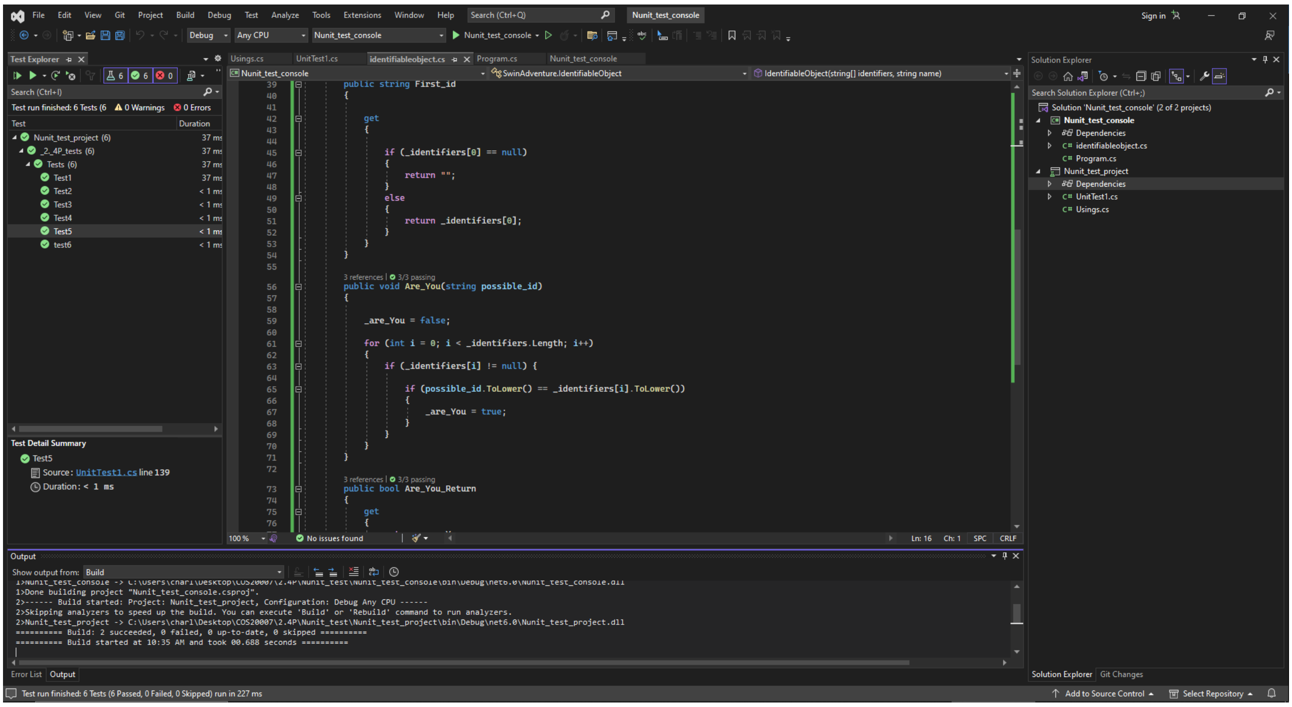
2.4P - Case Study Iteration 1 - Identifiable Object

PDF generated at 10:37 on Friday 10th March, 2023

Produced by OOP Submission Formatter

```
using System;

namespace SwinAdventure
{

    public class IdentifiableObject
    {

        private string _name;

        private string[] _identifiers = new string[15];

        private bool _are_You;

        private string empty_string = "";


        public IdentifiableObject(string[] identifiers, string name)
        {
            if (identifiers.Length != 0)
            {
                for (int i = 0; i < identifiers.Length; i++)
                {
                    _identifiers[i] = identifiers[i];
                }
            }
                _name = name;


        }
        public void Add_Identifier(string new_identifier)
        {

            _identifiers[2] = new_identifier;
        }

        public string[] Identifiers { get { return _identifiers; } }

        public string First_id
        {

            get
            {

                if (_identifiers[0] == null)
                {
                    return "";
                }
                else
                {
                    return _identifiers[0];
                }
            }
        }

        public void Are_You(string possible_id)
        {

            _are_You = false;

            for (int i = 0; i < _identifiers.Length; i++)
            {
                if (_identifiers[i] != null) {

                    if (possible_id.ToLower() == _identifiers[i].ToLower())
                    {
                        _are_You = true;
                    }
                }
            }
        }

        public bool Are_You_Return
        {
            get
            {
                return _are_You;
            }
        }
    }
}
```

```
using SwinAdventure;
using NUnit.Framework;

namespace _2._4P_tests
{
    public class Tests
    {

        [SetUp]

        public void Setup() { }

        [Test]
        public void Test1()
        {

            //ARRANGE
            string[] input_identifiers = { "id1", "id2", "id3" };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string are_you_test_input = "id2";

            //ACT

            test_object_1.Are_You(are_you_test_input);

            bool are_you_return = test_object_1.Are_You_Return;

            //ASSERT

            if (are_you_return)
            {
                Assert.Pass();
            }
            else
            {
                Assert.Fail();
            }

        }

        [Test]

        public void Test2()
        {
            //ARRANGE
            string[] input_identifiers = { "id1", "id2", "id3" };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string are_you_test_input = "id7";

            //ACT

            test_object_1.Are_You(are_you_test_input);

            bool are_you_return = test_object_1.Are_You_Return;

            //ASSERT

            if (are_you_return)
            {
                Assert.Fail();
            }
            else
            {
                Assert.Pass();
            }
        }

        [Test]

        public void Test3()
        {
            //ARRANGE
            string[] input_identifiers = { "id1", "id2", "id3" };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string are_you_test_input = "ID1";

            //ACT

            test_object_1.Are_You(are_you_test_input);

            bool are_you_return = test_object_1.Are_You_Return;

            //ASSERT

            if (are_you_return)
            {
                Assert.Pass();
            }
            else
            {
                Assert.Fail();
            }
        }

        [Test]

        public void Test4()
        {
            //ARRANGE

            string[] input_identifiers = { "id1", "id2", "id3" };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string first_id_test_string = "id1";

            //ACT

            string first_id = test_object_1.First_id;

            //ASSERT

            if (first_id_test_string == first_id)
            {
                Assert.Pass();
            }
            else
            {
                Assert.Fail();
            }
        }

        [Test]

        public void Test5()
        {
            //ARRANGE

            string[] input_identifiers = { };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string first_id_test_string = "";

            //ACT

            string first_id = test_object_1.First_id;

            //ASSERT

            if (first_id_test_string == first_id)
            {
                Assert.Pass();
            }
            else
            {
                Assert.Fail();
            }
        }

        [Test]

        public void test6()
        {
            //ARRANGE

            string[] input_identifiers = { "id1", "id2" };

            string input_name = "test";

            IdentifiableObject test_object_1 = new IdentifiableObject(input_identifiers, input_name);

            string new_id_string = "id3";

            //ACT

            test_object_1.Add_Identifier(new_id_string);

            string[] identifiers = test_object_1.Identifiers;

            //ASSERT

            if (new_id_string == identifiers[2])
            {
                Assert.Pass();
            }
            else
            {
                Assert.Fail();
            }

        }

    }
}
```



6 ✓ 6 ✗ 0

Search (Ctrl+I)

Test run finished: 6 Tests (6 ⚠ 0 Warnings ✗ 0 Errors

Test	Duration
✓ Nunit_test_project (6)	37 ms
✓ _2._4P_tests (6)	37 ms
✓ Tests (6)	37 ms
✓ Test1	37 ms
✓ Test2	< 1 ms
✓ Test3	< 1 ms
✓ Test4	< 1 ms
✓ Test5	< 1 ms
✓ test6	< 1 ms

Test Detail Summary

✓ Test5

Source: UnitTest1.cs line 139

Duration: < 1 ms

```
        public string First_id
        {

            get
            {

                if (_identifiers[0] == null)
                {
                    return "";
                }
                else
                {
                    return _identifiers[0];
                }
            }
        }

        3 references | ✓ 3/3 passing
        public void Are_You(string possible_id)
        {

            _are_You = false;

            for (int i = 0; i < _identifiers.Length; i++)
            {
                if (_identifiers[i] != null) {

                    if (possible_id.ToLower() == _identifiers[i].ToLower())
                    {
                        _are_You = true;
                    }
                }
            }
        }

        3 references | ✓ 3/3 passing
        public bool Are_You_Return
        {
            get
            {
```

Output

Show output from: Build

```
1>Nunit_test_console -> C:\Users\charl\Desktop\COS20007\2.4P\Nunit_test\Nunit_test_console\bin\Debug\net6.0\Nunit_test_console.dll
1>Done building project "Nunit_test_console.csproj".
2>------ Build started: Project: Nunit_test_project, Configuration: Debug Any CPU ------
2>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
2>Nunit_test_project -> C:\Users\charl\Desktop\COS20007\2.4P\Nunit_test\Nunit_test_project\bin\Debug\net6.0\Nunit_test_project.dll
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build started at 10:35 AM and took 00.688 seconds ==========
```

Test run finished: 6 Tests (6 Passed, 0 Failed, 0 Skipped) run in 227 ms