

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## 2.4P - Case Study Iteration 1 - Identifiable Object

---

PDF generated at 02:23 on Friday 24<sup>th</sup> March, 2023

```
1  using System;
2
3  namespace SwinAdventure
4  {
5      public class IdentifiableObject
6      {
7          private List<string> _identifiers;
8          public IdentifiableObject(string[] identifiers)
9          {
10              _identifiers = new List<string>();
11              foreach (string identifier in identifiers)
12                  _identifiers.Add(identifier.ToLower());
13          }
14
15          public bool AreYou(string identifier)
16          {
17              return _identifiers.Contains(identifier.ToLower());
18          }
19
20          public string FirstId
21          {
22              get
23              {
24                  if (_identifiers.Count == 0)
25                      return "";
26                  else
27                      return _identifiers.First();
28              }
29          }
30
31          public void AddIdentifier(string identifier)
32          {
33              _identifiers.Add(identifier.ToLower());
34          }
35      }
36  }
```

```
1 namespace SwinAdventure
2 {
3     public class IdentifiableObjectTest
4     {
5         private IdentifiableObject _testIdentifiers;
6
7         private IdentifiableObject _testEmptyIdentifiers;
8
9         [SetUp]
10        public void Setup()
11        {
12            _testIdentifiers = new IdentifiableObject(new string[] { "Trung",
↪ "Kien", "Nguyen" });
13
14            _testEmptyIdentifiers = new IdentifiableObject(new string[] { });
15        }
16
17        [TestCase("Kien")]
18        [TestCase("Trung")]
19        [TestCase("Nguyen")]
20        public void TestAreYou(string testIdentifier)
21        {
22            Assert.IsTrue(_testIdentifiers.AreYou(testIdentifier));
23        }
24
25        [TestCase("TrungKien")]
26        [TestCase("KienNguyen")]
27        public void TestNotAreYou(string testIdentifier)
28        {
29            Assert.IsFalse(_testIdentifiers.AreYou(testIdentifier));
30        }
31
32        [TestCase("kIeN")]
33        [TestCase("tRuNG")]
34        [TestCase("ngUYEn")]
35        public void TestCaseInsensitive(string testIdentifier)
36        {
37            Assert.IsTrue(_testIdentifiers.AreYou(testIdentifier));
38        }
39
40        [Test]
41        public void TestFirstID()
42        {
43            Assert.AreEqual("trung", _testIdentifiers.FirstId);
44            Assert.AreNotEqual("kien", _testIdentifiers.FirstId);
45        }
46
47        [Test]
48        public void TestFirstIDWithNoID()
49        {
50            Assert.AreEqual("", _testEmptyIdentifiers.FirstId);
51        }
52        [Test]
```

```
53     public void TestAddID()
54     {
55         // Test before adding
56         Assert.IsFalse(_testIdentifiers.AreYou("TrungKienNguyen"));
57
58         _testIdentifiers.AddIdentifier("TrungKienNguyen");
59         // Test added identifier
60         Assert.IsTrue(_testIdentifiers.AreYou("TrungKienNguyen"));
61
62         // Test whether others is modified or not
63         Assert.IsTrue(_testIdentifiers.AreYou("Trung"));
64         Assert.IsTrue(_testIdentifiers.AreYou("Kien"));
65         Assert.IsTrue(_testIdentifiers.AreYou("Nguyen"));
66     }
67 }
68 }
```

