SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

2.4P - Case Study Iteration 1 - Identifiable Object

PDF generated at 01:26 on Friday 10th March, 2023

```csharp
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace SwinAdventure
8   {
9       public class IdentifiableObject
10      {
11          List<string> _identifiers = new List<string>();
12
13          public IdentifiableObject(string[] idents)
14          {
15              foreach (string id in idents)
16              {
17                  AddIdentifier(id);
18              }
19          }
20
21          public bool AreYou(string id)
22          {
23              foreach (string identifier in _identifiers)
24              {
25                  if (id.ToLower() == identifier)
26                  {
27                      return true;
28                  }
29              }
30              return false;
31          }
32
33          public string FirstId
34          {
35              get
36              {
37                  if (_identifiers.Count == 0)
38                  {
39                      return "";
40                  }
41                  else
42                  {
43                      return _identifiers.First();
44                  }
45              }
46          }
47
48          public void AddIdentifier(string id)
49          {
50              _identifiers.Add(id.ToLower());
51          }
52      }
53  }
```

```csharp
using NUnit.Framework.Internal;

namespace TestSwinAdventure
{
    [TestFixture]
    public class TestIdentifiableObject
    {
        IdentifiableObject id;
        IdentifiableObject emptyId;

        [SetUp]
        public void SetUp()
        {
            id = new IdentifiableObject(new string[] { "shah", "shehraz" });
            emptyId = new IdentifiableObject(new string[] { });
        }

        [Test]
        public void TestAreYou()
        {
            Assert.That(id.AreYou("shah"), Is.True);
        }

        [Test]
        public void TestNotAreYou()
        {
            Assert.That(id.AreYou("paul"), Is.False);
        }

        [Test]
        public void TestCaseSensitive()
        {
            Assert.That(id.AreYou("sHaH"), Is.True);
        }

        [Test]
        public void TestFirstId()
        {
            Assert.That(id.FirstId, Is.SameAs("shah"));
        }

        [Test]
        public void TestFirstIdWithNoIds()
        {
            Assert.That(emptyId.FirstId, Is.SameAs(""));
        }

        [Test]
        public void TestAddId()
        {
            id.AddIdentifier("jacob");
            Assert.That(id.AreYou("jacob"), Is.True);
        }
```

```
54
55        }
56    }
```