SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

2.4P - Case Study Iteration 1 - Identifiable Object

PDF generated at 20:40 on Monday 20th March, 2023

```csharp
1   using System;
2   using System.Collections;
3   using System.Collections.Generic;
4   using System.Linq;
5   using System.Runtime.CompilerServices;
6   using System.Security.Cryptography.X509Certificates;
7   using System.Text;
8   using System.Threading.Tasks;
9
10  namespace Iteration1
11  {
12      public class IdentifiableObject
13      {
14
15          private List<string> _identifiers;
16
17          public IdentifiableObject(string[] idents)
18          {
19              _identifiers = new List<string>();
20
21
22              foreach (string ident in idents)
23              {
24                  _identifiers.Add(ident.ToLower());
25              }
26
27          }
28          public bool AreYou(string id)
29          {
30              return _identifiers.Contains(id.ToLower());
31
32          }
33          public string FirstId
34          {
35              get
36              {
37                  if ( _identifiers.Count == 0 )
38                  {
39                      return "";
40
41                  }
42                  else
43                  {
44                      return _identifiers[0];
45                  }
46              }
47          }
48          public void AddIdentifier(string id)
49
50          {
51              _identifiers.Add(id.ToLower());
52          }
53
```

```
54        }
55
56
57  }
58
59
60
61
62
63
64
65
66
67
68
69
70
71
```

```csharp
1   using Iteration1;
2   using NuGet.Frameworks;
3   using NUnit.Framework.Constraints;
4   using System.Runtime.CompilerServices;
5
6   namespace TestIteration1
7
8   {
9       public class TestIteration1
10      {
11          private IdentifiableObject _Object;
12          private IdentifiableObject _EmptyObject;
13
14
15          [SetUp]
16          public void Setup()
17          {
18              _Object = new IdentifiableObject(new string[] { "fred", "bob" });
19              _EmptyObject = new IdentifiableObject(new string[] { });
20
21          }
22
23          [Test]
24          public void TestAreYou()
25          {
26              Assert.IsTrue(_Object.AreYou("bob"));
27              Assert.IsTrue(_Object.AreYou("fred"));
28          }
29
30          [Test]
31          public void TestNotAreYou()
32          {
33              Assert.IsFalse(_Object.AreYou("wilma"));
34          }
35          [Test]
36          public void TestCaseSensetive()
37          {
38              Assert.IsTrue(_Object.AreYou("FReD"));
39              Assert.IsTrue(_Object.AreYou("bOB"));
40          }
41          [Test]
42          public void TestFirstID()
43          {
44              Assert.AreEqual("fred", _Object.FirstId);
45          }
46          [Test]
47          public void TestFirstIDWithNoIDs()
48          {
49              Assert.AreEqual("",_EmptyObject.FirstId);
50          }
51          [Test]
52          public void TestAddId()
53          {
```

```
54              _Object.AddIdentifier("wilma");
55              Assert.IsTrue(_Object.AreYou("wilma"));
56              Assert.IsTrue(_Object.AreYou("fred"));
57              Assert.IsTrue(_Object.AreYou("bob"));
58
59          }
60
61
62
63
64
65
66
67
68
69
70
71
72      }
73  }
```