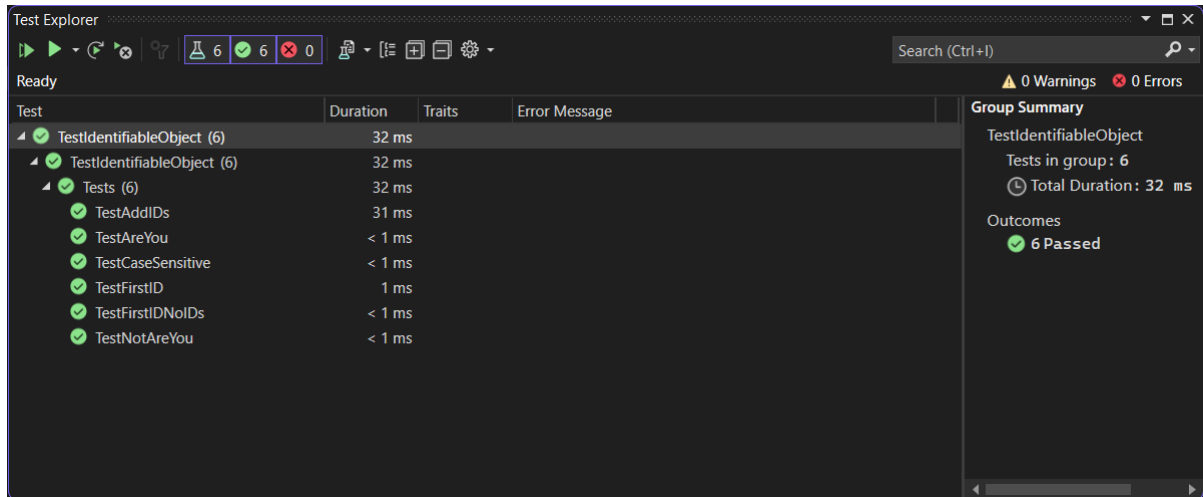


2.4P - Case Study Iteration 1 - Identifiable Object

Jayden Kong, 104547242

Tests passing:



```
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        private List<string> _identifiers;
12
13        public string FirstID
14        {
15            get
16            {
17                if (_identifiers.Count > 0)
18                {
19                    return _identifiers[0];
20                }
21                else
22                {
23                    return "";
24                }
25            }
26        }
27
28        public IdentifiableObject(string[] idents)
29        {
30            _identifiers = new List<string>();
31            for (int i = 0; i < idents.Length; i++)
32            {
33                _identifiers.Add(idents[i].ToLower());
34            }
35        }
36
37        public bool AreYou(string id)
38        {
39            bool result = false;
40
41            foreach (string ident in _identifiers)
42            {
43                if (ident == id.ToLower())
44                {
45                    result = true;
46                    break;
47                }
48            }
49        }
50    }
51 }
```

```
50         return result;
51     }
52
53     public void AddIdentifier(string id)
54     {
55         _identifiers.Add(id.ToLower());
56     }
57 }
58 }
59
```

```
1 namespace SwinAdventure
2 {
3     internal class Program
4     {
5         static void Main(string[] args)
6         {
7             IdentifiableObject id = new IdentifiableObject(new string[]
8                 { "id1", "id2" });    // Example of Identifiable Object
9         }
10 }
11
```

```
1 using SwinAdventure;
2
3 namespace TestIdentifiableObject
4 {
5     public class Tests
6     {
7
8         [Test]
9         public void TestAreYou()
10        {
11            IdentifiableObject myIdents = new IdentifiableObject(new string [] { "fred", "bob" });
12
13            bool fred = myIdents.AreYou("fred");
14            Assert.That(fred, Is.EqualTo(true));
15            bool bob = myIdents.AreYou("bob");
16            Assert.That(bob, Is.EqualTo(true));
17        }
18
19        [Test]
20
21        public void TestNotAreYou()
22        {
23            IdentifiableObject myIdents = new IdentifiableObject(new string [] { "fred", "bob" });
24
25            bool wilma = myIdents.AreYou("wilma");
26            Assert.That(wilma, Is.EqualTo(false));
27            bool boby = myIdents.AreYou("boby");
28            Assert.That(boby, Is.EqualTo(false));
29        }
30
31        [Test]
32
33        public void TestCaseSensitive()
34        {
35            IdentifiableObject myIdents = new IdentifiableObject(new string [] { "fred", "bob" });
36
37            bool fred = myIdents.AreYou("FRED");
38            Assert.That(fred, Is.EqualTo(true));
39            bool bob = myIdents.AreYou("bOB");
40            Assert.That(bob, Is.EqualTo(true));
41        }
42
43        [Test]
44
45        public void TestFirstID()
46        {
```

```
47     IdentifiableObject myIdsents = new IdentifiableObject(new string [] { "fred", "bob" });
48
49     string firstID = myIdsents.FirstID;
50     Assert.That(firstID, Is.EqualTo("fred"));
51 }
52
53 [Test]
54
55 public void TestFirstIDNoIDs()
56 {
57     IdentifiableObject myIdsents = new IdentifiableObject(new string [] {});
58
59     string firstID = myIdsents.FirstID;
60     Assert.That(firstID, Is.EqualTo(""));
61 }
62
63 [Test]
64
65 public void TestAddIDs()
66 {
67     IdentifiableObject myIdsents = new IdentifiableObject(new string [] { "fred", "bob" });
68     myIdsents.AddIdentifier("wilma");
69
70     bool fred = myIdsents.AreYou("fred");
71     Assert.That(fred, Is.EqualTo(true));
72     bool bob = myIdsents.AreYou("bob");
73     Assert.That(bob, Is.EqualTo(true));
74     bool wilma = myIdsents.AreYou("wilma");
75     Assert.That(wilma, Is.EqualTo(true));
76 }
77 }
78 }
```