Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 2 - Players Items and Inventory

PDF generated at 16:44 on Sunday 1st October, 2023

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration2
{
    public class GameObject : IdentifiableObject
    {
        private string Odescription;
        private string Oname;

        public GameObject(string[]ids,string name,string desc):base(ids)
        {
            Oname = name;
            Odescription = desc;
        }
        public string Name
        {
            get => Oname;
        }
        public string ShortDescription
        {
            get => Oname + "("+FirstId+")";
        }
        public virtual string  FullDescription
        {
            get => Odescription;
        }


    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration2
{
    public class Player:GameObject
    {
        private Inventory Oinventory;
        public Player(string name, string desc):base(new string[] { "me", "inventory" }, name, desc)
        {
            Oinventory = new Inventory();
        }
        public GameObject Located(string id)
        {
            if (AreYou(id) == true)

            {
                return this ;
            }
            return Oinventory.Fetch(id);
        }
        public  override string FullDescription
        {
            get { return $"You are {Name} {base.FullDescription}\nYou are carrying:\n{Oinventory.Itemlist}"; }
        }
        public Inventory Inventory
        {
            get { return Oinventory; }
        }
    }
}
```

```
using CaseStudy_Iteration2;
using NUnit.Framework;
using System.Runtime.CompilerServices;

namespace Iteration2Test
{
    [TestFixture]
    public class TestPlayer
    {
        private bool result;
        private Player Oplayer;
        private Item Oitem1;
        private Item Oitem2;

        [SetUp]
        public void Setup()
        {
            Oplayer = new Player("Tung","professional");
            Oitem1 = new Item(new String[] { "akm", "m24" }, "gun", "This is power gun");
            Oitem2 = new Item(new String[] { "silver", "dark" }, "sword", "This is a beautiful sword");
        }
        [Test]
        public void TestIdentifiale()
        {
            Assert.IsTrue(Oplayer.AreYou("me"));
            Assert.IsTrue(Oplayer.AreYou("inventory"));
        }
        [Test]
        public void TestLocatesItem()
        {
            Oplayer.Inventory.Put(Oitem1);
            Assert.That(Oplayer.Located("akm"), Is.EqualTo(Oitem1));
            Oplayer.Inventory.Put(Oitem2);
            Assert.That(Oplayer.Located("dark"), Is.EqualTo(Oitem2));
        }
        [Test]
        public void TestLocatesItself()
        {
            Assert.That(Oplayer.Located("me"), Is.EqualTo(Oplayer));
        }
        [Test]
        public void TestLocateNothing()
        {

            Oplayer.Inventory.Put(Oitem1);
            Oplayer.Inventory.Put(Oitem2);
            if (Oplayer.Located("Nam") != Oitem1 || Oplayer.Located("Nam") != Oitem2)
            {
                result = false;
            }
            Assert.False(result);
```

```
52          }
53          [Test]
54          public void TestDescription()
55          {
56              string text = "You are Tung professional\nYou are
    ↪  carrying:\n\tgun(m24)\n\tsword(silver) ";
57              StringAssert.Contains(Oplayer.FullDescription,text);
58          }
59      }
60  }
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration2
{
    public class Item:GameObject
    {
        public Item(string[]idens,string name, string desc):base(idens,name,desc)
        {

        }

    }
}
```

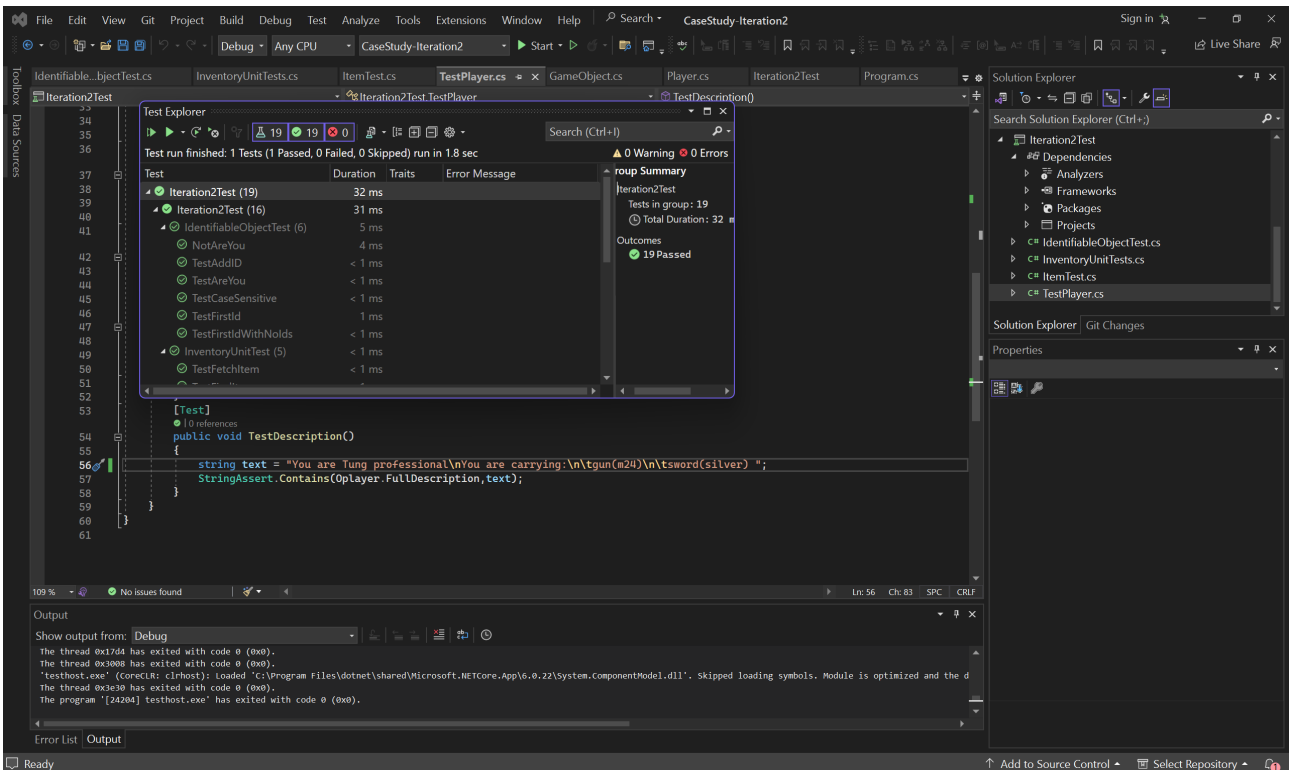
```
using CaseStudy_Iteration2;
using NUnit.Framework;

namespace TestItems
{
    [TestFixture]
    public class ItemTests
    {
        private Item Oitem;

        [SetUp]
        public void Setup()
        {
             Oitem = new Item(new String[] { "lavender", "rose" },"flowers","This is a beautiful flower");
        }

        [Test]
        public void TestIndentifiabel()
        {
            Assert.IsTrue(Oitem.AreYou("lavender"));
        }
        [Test]
        public void TestShortDescription()
        {
            Assert.That(Oitem.ShortDescription, Is.EqualTo("flowers(lavender)"));
        }
        [Test]
        public void TestFullDescription()
        {
            Assert.That(Oitem.FullDescription, Is.EqualTo("This is a beautiful flower"));
        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration2
{
    public class Inventory
    {
        private List<Item> items = new List<Item>();

        public Inventory()
        {

        }
        public bool HasItem(string id)
        {
            foreach (var item in items)
            {
                if (item.AreYou(id) == true)
                    return true;
            }
            return false;
        }
        public void Put(Item itm)
        {
            items.Add(itm);
        }
        public Item Fetch(string id)
        {
            foreach (var item in items)
            {
                if (item.AreYou(id) == true)
                {
                    return item;
                }
            }
                    return null;

        }
        public Item Take(string id)
        {
            foreach(var item in items)
            {
                if(item.AreYou(id) == true)
                {
                    items.Remove(item);
                }
            }
            return null;
        }
        public string Itemlist
```

```
54          {
55              get
56              {
57                  string list = string.Empty;
58                  foreach (var item in items)
59                  {
60                      list += "\t" + item.ShortDescription + "\n";
61                  }
62                  return list;
63              }
64          }
65      }
66  }
```

```
using CaseStudy_Iteration2;
using Microsoft.VisualStudio.TestPlatform.Utilities;
using NUnit.Framework;
using TestItems;


namespace Iteration2Test
{
    [TestFixture]
   public class InventoryUnitTest
    {
        private Inventory inventory;
        private Item Oitm1 , Oitm2 ;
        [SetUp]
        public void Setup()
        {
            inventory = new Inventory();
            Oitm1 = new Item(new string[] { "lavender", "rose" }, "flowers", "This is a beautiful flower");
            Oitm2 = new Item(new string[] {"carnation"},"flower","This is a special flower");
        }
        [Test]
        public void TestFindItem()
        {
            inventory.Put(Oitm2);
            Assert.IsTrue(inventory.HasItem("carnation"));
        }
        [Test]
        public void TestNoItemFind()
        {
            inventory.Put(Oitm1);
            Assert.IsFalse(inventory.HasItem("carnation"));
        }
        [Test]
        public void TestFetchItem()
        {
            inventory.Put(Oitm1);
            Assert.That(inventory.Fetch("lavender"), Is.EqualTo(Oitm1));
        }
        [Test]
        public void TestTakeItem()
        {
            inventory.Put(Oitm2);
            inventory.Take("lavender");
            Assert.IsFalse(inventory.HasItem("lavender"));
        }
        [Test]
        public void TestListItem()
        {
            string list = "flowers(lavender)\n"+ "flowers(rose)\n";
            StringAssert.Contains(inventory.Itemlist,list);
        }
```

```
52      }
53
54  }
```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
CaseStudy-Iteration2
Debug
Any CPU
Start
Live Share
Test Explorer
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 1.8 sec
Iteration2Test (19) 32 ms
Iteration2Test (16) 31 ms
IdentifiableObjectTest (6) 5 ms
NotAreYou 4 ms
TestAddID < 1 ms
TestAreYou < 1 ms
TestCaseSensitive < 1 ms
TestFirstId 1 ms
TestFirstIdWithNolds < 1 ms
InventoryUnitTest (5) < 1 ms
TestFetchItem < 1 ms
Tests in group: 19
Total Duration: 32
19 Passed
[Test]
public void TestDescription()
{
string text = "You are Tung professional\nYou are carrying:\n\tgun(m24)\n\tsword(silver) ";
StringAssert.Contains(Oplayer.FullDescription,text);
}
Solution Explorer
Iteration2Test
Dependencies
Analyzers
Frameworks
Packages
Projects
IdentifiableObjectTest.cs
InventoryUnitTests.cs
ItemTest.cs
TestPlayer.cs
Properties
Output
Show output from: Debug
Error List
Output
Ready
Add to Source Control
Select Repository