Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 4 - Look Command

PDF generated at 23:14 on Thursday 28th September, 2023

```
using System;
namespace Iteration4
{
    public interface IHaveInventory
    {
        Game_Object Locate(string id);


        public string Name
        {
            get;
        }
    }
}

```

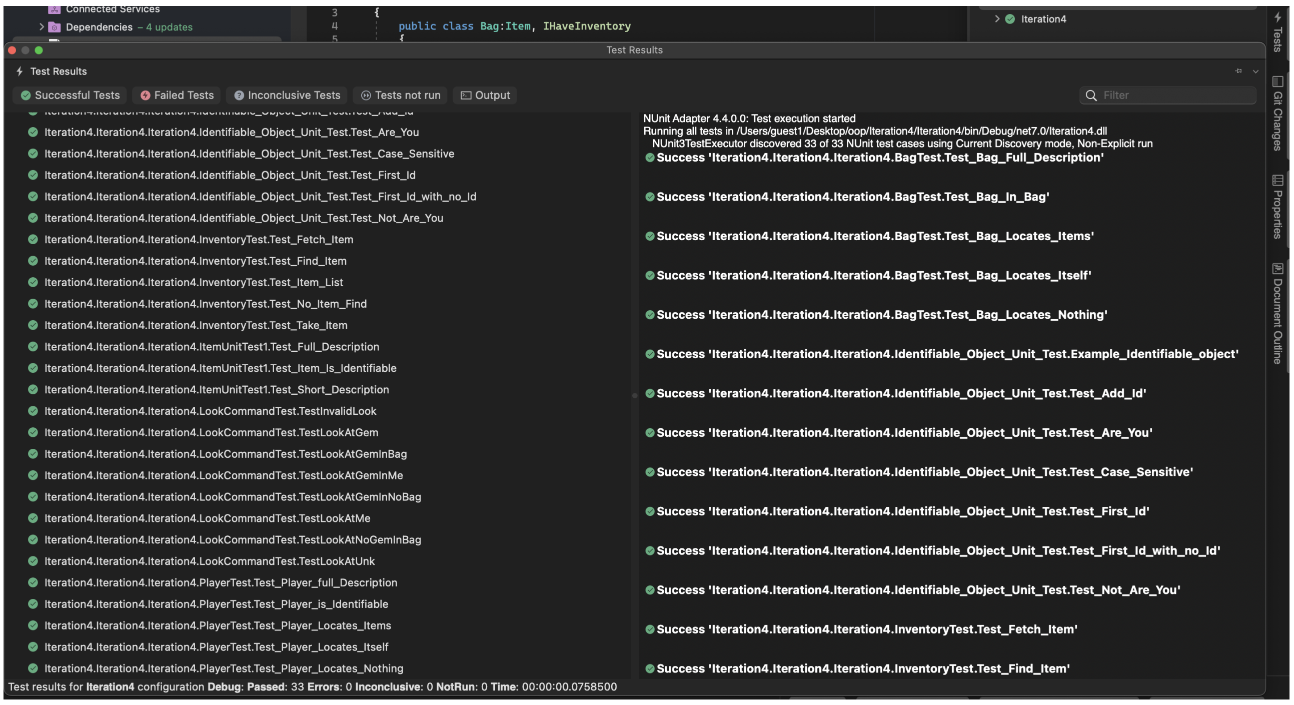
```
using System;
namespace Iteration4
{
    public class Player: Game_Object, IHaveInventory
    {
        Inventory _inventory = new Inventory();

        public Player(string name, string desc):base(new string[] {"me",
    "inventory"}, name, desc)
        {
            //Inventory _inventory = new() Inventory;
        }

        public Game_Object Locate(string id)
        {
            if(AreYou(id) == true)
            {
                return this;
            }
            return _inventory.Fetch(id);
        }
        public Inventory Inv
        {
            get
            {
                return _inventory;
            }
        }
        public override string FullDescription //! Can only override virtual
    properties
        {
            get
            {
                return ("You are " + Name + " " + base.FullDescription + "." + "\nYou
    are carrying\n" + Inv.ItemList);
            }
        }
    }
}

```

```
using System;
namespace Iteration4
{
    public class Bag:Item, IHaveInventory
    {
        Inventory _inventory = new Inventory();
        public Bag(string[] ids, string name, string desc):base(ids, name, desc)
        {


        }
        public Game_Object Locate(string id)
        {
            if (AreYou(id) == true)
            {
                return this;
            }
            return _inventory.Fetch(id);
        }
        public override string FullDescription
        {
            get
            {
                return ("In this " + Name + " you can see:\n\t" + Inv.ItemList);
            }
        }
        public Inventory Inv
        {
            get
            {
                return _inventory;
            }
        }
    }
}

```

```
using System;
namespace Iteration4
{
    public abstract class Command:Identifiable_object
    {
        public Command(string[] ids):base(ids)
        {
        }
        public abstract string Execute(Player p, string[] text);
    }
}

```

```
using System;
using System.ComponentModel;
using System.Numerics;

namespace Iteration4
{
    public class LookCommand:Command
    {

        IHaveInventory container;
        string thingId;

        public LookCommand() : base(new string[] {"look"} )
        {

        }

        public override string Execute(Player p, string[] text)
        {
            if (text.Length != 3 && text.Length != 5)
            {
                return ("I don't know how to look like that.");
            }

            if (text[0] != "look")
            {
                return ("Error in look input");
            }

            if (text[1] != "at")
            {
                return ("What do you want to look at?");
            }

            if (text.Length == 5 && text[3] != "in")
            {
                return ("What do you want to look in?");
            }
            if (text.Length == 3)
            {
                container = p;
            }
            if (text.Length == 5)
            {
                container = FetchContainer(p, text[4]);
                if (container == null)
                {
                    return ("I cannot find the " + text[4]);
                }
            }
            thingId = text[2];
            return LookAtIn(thingId, container);

        }
        private IHaveInventory FetchContainer(Player p, string containerId)
        {
            return (IHaveInventory)p.Locate(containerId);
        }
        private string LookAtIn(string thingId, IHaveInventory container)
        {
           if (container.Locate(thingId) == null)
           {
               return ("I cannot find the " + thingId + " in the " + container.Name);
           }
            else
            {
                return container.Locate(thingId).FullDescription;
            }

        }

    }
}

```

```
using System;
using System.ComponentModel;
using System.Xml.Linq;
using Iteration4;
namespace Iteration4
{
    [TestFixture()]
    public class LookCommandTest
    {
        LookCommand l;
        Player p;
        Item gem;
        Bag b;

        [SetUp()]
        public void Setup()
        {
            l = new ();
            p = new ("Fred", "the mighty programmer");
            b = new(new string[] { "bag" }, "leather bag", "small brown");
            gem = new(new string[] { "gem" }, "gem", "A bright red");

            p.Inv.Put(gem);

        }
        [Test()]
        public void TestLookAtMe()
        {
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "inventory" }),
    ↪   p.FullDescription);
        }
        [Test()]
        public void TestLookAtGem()
        {
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem" }),
    ↪   gem.FullDescription);
        }
        [Test()]
        public void TestLookAtUnk()
        {
            p.Inv.Take("gem");
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem"}), "I
    ↪   cannot find the gem in the Fred");
        }

        [Test()]
        public void TestLookAtGemInMe()
        {
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪   "inventory" }), gem.FullDescription);


        }

        [Test()]
        public void TestLookAtGemInBag()
        {
            b.Inv.Put(gem);
            p.Inv.Put(b);

            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪ "bag" }), gem.FullDescription);
        }

        [Test()]
        public void TestLookAtGemInNoBag()
        {
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪ "bag" }), "I cannot find the bag");
        }
        [Test()]
        public void TestLookAtNoGemInBag()
        {
            p.Inv.Put(b);

            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪ "bag" }), "I cannot find the gem in the leather bag");
        }
        [Test()]
        public void TestInvalidLook()
        {
            Assert.AreEqual(l.Execute(p, new string[] { "look", "around" }), "I don't
    ↪ know how to look like that.");
            Assert.AreEqual(l.Execute(p, new string[] { "Hello", "Sanya", "Baweja"}),
    ↪ "Error in look input");
            Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "a", "at", "b"
    ↪ }), "What do you want to look in?");


        }
    }
}
```



```
{
    public class Bag:Item, IHaveInventory
    {
```

```
Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Are_You
Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Case_Sensitive
Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_First_Id
Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_First_Id_with_no_Id
Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Not_Are_You
Iteration4.Iteration4.Iteration4.InventoryTest.Test_Fetch_Item
Iteration4.Iteration4.Iteration4.InventoryTest.Test_Find_Item
Iteration4.Iteration4.Iteration4.InventoryTest.Test_Item_List
Iteration4.Iteration4.Iteration4.InventoryTest.Test_No_Item_Find
Iteration4.Iteration4.Iteration4.InventoryTest.Test_Take_Item
Iteration4.Iteration4.Iteration4.ItemUnitTest1.Test_Full_Description
Iteration4.Iteration4.Iteration4.ItemUnitTest1.Test_Item_Is_Identifiable
Iteration4.Iteration4.Iteration4.ItemUnitTest1.Test_Short_Description
Iteration4.Iteration4.Iteration4.LookCommandTest.TestInvalidLook
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtGem
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtGemInBag
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtGemInMe
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtGemInNoBag
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtMe
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtNoGemInBag
Iteration4.Iteration4.Iteration4.LookCommandTest.TestLookAtUnk
Iteration4.Iteration4.Iteration4.PlayerTest.Test_Player_full_Description
Iteration4.Iteration4.Iteration4.PlayerTest.Test_Player_is_Identifiable
Iteration4.Iteration4.Iteration4.PlayerTest.Test_Player_Locates_Items
Iteration4.Iteration4.Iteration4.PlayerTest.Test_Player_Locates_Itself
Iteration4.Iteration4.Iteration4.PlayerTest.Test_Player_Locates_Nothing
```

```
NUnit Adapter 4.4.0.0: Test execution started
Running all tests in /Users/guest1/Desktop/oop/Iteration4/Iteration4/bin/Debug/net7.0/Iteration4.dll
  NUnit3TestExecutor discovered 33 of 33 NUnit test cases using Current Discovery mode, Non-Explicit run
Success 'Iteration4.Iteration4.Iteration4.BagTest.Test_Bag_Full_Description'

Success 'Iteration4.Iteration4.Iteration4.BagTest.Test_Bag_In_Bag'

Success 'Iteration4.Iteration4.Iteration4.BagTest.Test_Bag_Locates_Items'

Success 'Iteration4.Iteration4.Iteration4.BagTest.Test_Bag_Locates_Itself'

Success 'Iteration4.Iteration4.Iteration4.BagTest.Test_Bag_Locates_Nothing'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Example_Identifiable_object'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Add_Id'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Are_You'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Case_Sensitive'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_First_Id'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_First_Id_with_no_Id'

Success 'Iteration4.Iteration4.Iteration4.Identifiable_Object_Unit_Test.Test_Not_Are_You'

Success 'Iteration4.Iteration4.Iteration4.InventoryTest.Test_Fetch_Item'

Success 'Iteration4.Iteration4.Iteration4.InventoryTest.Test_Find_Item'
```

Test results for **Iteration4** configuration **Debug**: **Passed**: 33 **Errors**: 0 **Inconclusive**: 0 **NotRun**: 0 **Time**: 00:00:00.0758500