Swinburne University of Technology

COS20007 Object Oriented Programming

6.1P - Case Study - Iteration 4 - Look Command

PDF generated at 16:35 on Monday 8th May, 2023

```
using System;

namespace SwinAdventure
{
    public interface IhaveInventory
    {
        GameObject Locate(string id);

        public string Name
        {
            get;
        }
    }

}

```

```
using System;

namespace SwinAdventure
{
    public class Player : GameObject, IhaveInventory
    {
        private Inventory _inventory;

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

        public override string FullDescription
        {
            get { return $"{Name}, {base.FullDescription}, you are carrying: {_inventory.ItemList}"; }

        }

        public Inventory Inventory
        {
            get { return _inventory; }
        }


        public GameObject Locate(string Id)
        {
            if (AreYou(Id) == true)
            {
                return this;
            }
            return _inventory.Fetch(Id);
        }


    }
}

```

```
using System;

namespace SwinAdventure
{
    public class Bag : Item, IhaveInventory
    {
        private Inventory _inventory;

        public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
        {
            _inventory = new Inventory();
        }

        public GameObject Locate(string id)
        {
            if (this.AreYou(id))
            {
                return this;
            }
            else
            {
                return _inventory.Fetch(id);
            }
        }

        public override string FullDescription
        {
            get { return $"in the {this.Name} you can see:" + _inventory.ItemList;}

        }

        public Inventory Inventory
        {
            get { return _inventory; }

        }
    }

}

```

```
using System;

namespace SwinAdventure
{
    public abstract class Command : IdentifiableObject
    {
        public Command(string[] ids): base(ids)
        {

        }

        public abstract string Execute(Player p, string[] text);

    }

}

```

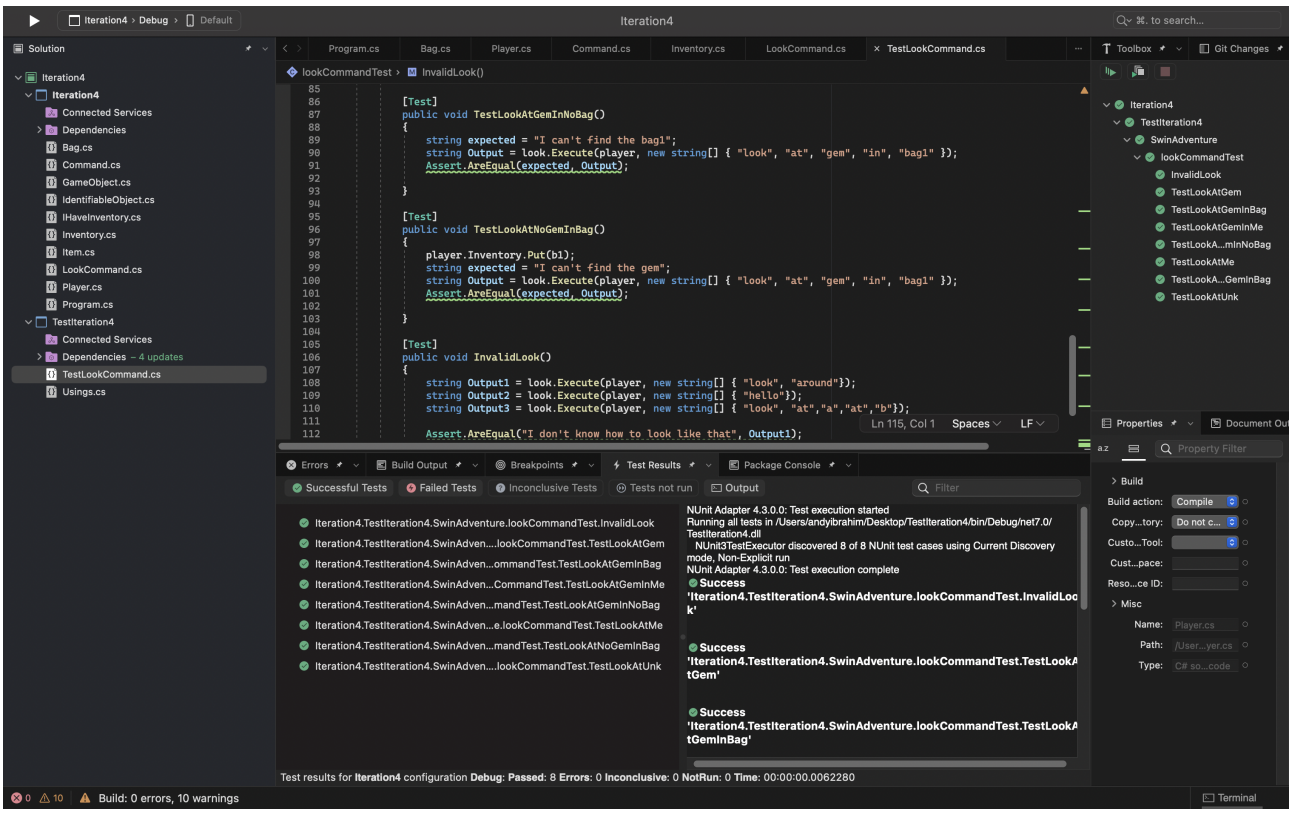
```
using System;
using System.ComponentModel;
using System.Numerics;

namespace SwinAdventure
{
    public class LookCommand : Command
    {
        public LookCommand(): base(new string[] { "look" }) { }


        public override string Execute(Player p, string[] text)
        {
            IhaveInventory container;
            string thingId;

            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;
            }

            else
            {
                container = FetchContainer(p, text[4]);
            }


            if (container == null)
            {
                return $"I can't find the {text[4]}";
            }

            thingId = text[2];
```

```
54
55              return LookAtIn(thingId, container);
56
57          }
58
59
60          private IhaveInventory FetchContainer(Player p, string containerId)
61          {
62              return p.Locate(containerId) as IhaveInventory;
63          }
64
65
66
67          private string LookAtIn(string thingId, IhaveInventory container)
68          {
69              if (container.Locate(thingId) == null)
70              {
71                  return $"I can't find the {thingId}";
72              }
73              else
74              {
75                  return container.Locate(thingId).FullDescription;
76              }
77          }
78
79
80      }
81
82  }
83
```

```
using System;
using System.Numerics;
using System.Xml.Linq;
using Microsoft.VisualStudio.TestPlatform.Utilities;
using NUnit.Framework;

namespace SwinAdventure
{
    [TestFixture]

    public class lookCommandTest
    {

        LookCommand look;
        Player player;
        Bag b1;
        Item gem;

    [SetUp]
        public void SetUp()
        {

            gem = new Item(new string[] { "gem" }, "a gem", "this is a gem");
            b1 = new Bag(new string[] { "bag1" }, "a bag1", "This is a bag1");
            player = new Player("andy", "music producer");
            look = new LookCommand();


        }

        [Test]
        public void TestLookAtMe()
        {
            string Output = look.Execute(player, new string[] { "look", "at",
↪   "inventory" });

            string expected = $"andy, music producer, you are carrying: ";

            Assert.AreEqual(expected, Output);

        }

        [Test]
        public void TestLookAtGem()
        {
            player.Inventory.Put(gem);

            string Output = look.Execute(player, new string[] { "look", "at", "gem"
↪   });
            string expected= $"{gem.FullDescription}";

            Assert.AreEqual(expected, Output);

```

```
52          }
53
54          [Test]
55          public void TestLookAtUnk()
56          {
57              string Output = look.Execute(player, new string[] { "look", "at", "gem"
    ↪  });
58              string expected = $"I can't find the gem";
59
60              Assert.AreEqual(expected, Output);
61
62          }
63
64          [Test]
65          public void TestLookAtGemInMe()
66          {
67              player.Inventory.Put(gem);
68              string Output = look.Execute(player, new string[] { "look", "at", "gem",
    ↪  "in", "me" });
69              string expected = $"{gem.FullDescription}";
70              Assert.AreEqual(expected, Output);
71
72          }
73
74          [Test]
75          public void TestLookAtGemInBag()
76          {
77              player.Inventory.Put(b1);
78              b1.Inventory.Put(gem);
79
80              string Output = look.Execute(player, new string[] { "look", "at", "gem",
    ↪  "in", "bag1" });
81              string expected = $"{gem.FullDescription}";
82              Assert.AreEqual(expected, Output);
83
84          }
85
86          [Test]
87          public void TestLookAtGemInNoBag()
88          {
89              string expected = "I can't find the bag1";
90              string Output = look.Execute(player, new string[] { "look", "at", "gem",
    ↪  "in", "bag1" });
91              Assert.AreEqual(expected, Output);
92
93          }
94
95          [Test]
96          public void TestLookAtNoGemInBag()
97          {
98              player.Inventory.Put(b1);
99              string expected = "I can't find the gem";
100             string Output = look.Execute(player, new string[] { "look", "at", "gem",
    ↪  "in", "bag1" });
```

```
101             Assert.AreEqual(expected, Output);
102 
103         }
104 
105         [Test]
106         public void InvalidLook()
107         {
108             string Output1 = look.Execute(player, new string[] { "look", "around"});
109             string Output2 = look.Execute(player, new string[] { "hello"});
110             string Output3 = look.Execute(player, new string[] { "look",
    ↪  "at","a","at","b"});
111 
112             Assert.AreEqual("I don't know how to look like that", Output1);
113             Assert.AreEqual("I don't know how to look like that", Output2);
114             Assert.AreEqual("What do you want to look in?", Output3);
115 
116 
117         }
118     }
119 }
```


Iteration4
Solution
Iteration4
Connected Services
Dependencies
Bag.cs
Command.cs
GameObject.cs
IdentifiableObject.cs
IHaveInventory.cs
Inventory.cs
Item.cs
LookCommand.cs
Player.cs
Program.cs
TestIteration4
Connected Services
Dependencies – 4 updates
TestLookCommand.cs
Usings.cs
Program.cs
Bag.cs
Player.cs
Command.cs
Inventory.cs
LookCommand.cs
TestLookCommand.cs
lookCommandTest › InvalidLook()
[Test]
public void TestLookAtGemInNoBag()
{
string expected = "I can't find the bag1";
string Output = look.Execute(player, new string[] { "look", "at", "gem", "in", "bag1" });
Assert.AreEqual(expected, Output);
}
[Test]
public void TestLookAtNoGemInBag()
{
player.Inventory.Put(b1);
string expected = "I can't find the gem";
string Output = look.Execute(player, new string[] { "look", "at", "gem", "in", "bag1" });
Assert.AreEqual(expected, Output);
}
[Test]
public void InvalidLook()
{
string Output1 = look.Execute(player, new string[] { "look", "around"});
string Output2 = look.Execute(player, new string[] { "hello"});
string Output3 = look.Execute(player, new string[] { "look", "at","a","at","b"});
Assert.AreEqual("I don't know how to look like that", Output1);
Ln 115, Col 1 Spaces LF
Toolbox
Git Changes
Iteration4
TestIteration4
SwinAdventure
lookCommandTest
InvalidLook
TestLookAtGem
TestLookAtGemInBag
TestLookAtGemInMe
TestLookA...mInNoBag
TestLookAtMe
TestLookA...GemInBag
TestLookAtUnk
Properties
Property Filter
Build
Build action: Compile
Misc
Name: Player.cs
Type: C# so...code
Errors
Build Output
Breakpoints
Test Results
Package Console
Successful Tests
Failed Tests
Inconclusive Tests
Tests not run
Output
Filter
Iteration4.TestIteration4.SwinAdventure.lookCommandTest.InvalidLook
Iteration4.TestIteration4.SwinAdven....lookCommandTest.TestLookAtGem
Iteration4.TestIteration4.SwinAdven...ommandTest.TestLookAtGemInBag
Iteration4.TestIteration4.SwinAdven...CommandTest.TestLookAtGemInMe
Iteration4.TestIteration4.SwinAdven...mandTest.TestLookAtGemInNoBag
Iteration4.TestIteration4.SwinAdven...e.lookCommandTest.TestLookAtMe
Iteration4.TestIteration4.SwinAdven...mandTest.TestLookAtNoGemInBag
Iteration4.TestIteration4.SwinAdven....lookCommandTest.TestLookAtUnk
NUnit Adapter 4.3.0.0: Test execution started
Running all tests in /Users/andyibrahim/Desktop/TestIteration4/bin/Debug/net7.0/TestIteration4.dll
NUnit3TestExecutor discovered 8 of 8 NUnit test cases using Current Discovery mode, Non-Explicit run
NUnit Adapter 4.3.0.0: Test execution complete
Success 'Iteration4.TestIteration4.SwinAdventure.lookCommandTest.InvalidLook'
Success 'Iteration4.TestIteration4.SwinAdventure.lookCommandTest.TestLookAtGem'
Success 'Iteration4.TestIteration4.SwinAdventure.lookCommandTest.TestLookAtGemInBag'
Test results for Iteration4 configuration Debug: Passed: 8 Errors: 0 Inconclusive: 0 NotRun: 0 Time: 00:00:00.0062280
0 10 Build: 0 errors, 10 warnings
Terminal