Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 6 - Locations

PDF generated at 20:54 on Friday 3rd November, 2023

```
using System;
using System.IO;
using Iteration5;
namespace Iteration5
{
    public class Location : Game_Object, IHaveInventory

    {
        Inventory _inv = new Inventory();

        public Location(string name, string desc) : base(new string[] { "room" }, name, desc)
        {

        }
        public Game_Object Locate(string id)
        {

            if (AreYou(id) == true)
            {
                return this;
            }
            else
            {
                return _inv.Fetch(id);
            }
        }
        public override string FullDescription
        {
            get
            {
                return ("You are in the " + Name + "\n" + base.FullDescription + ".\nIn this room you can see:\n" + _inv.ItemList);
            }
        }
        public Inventory Inv
        {
            get
            {
                return _inv;
            }
        }
    }
}

```

```
using System;
using System.ComponentModel;
using System.Xml.Linq;
using Iteration5;
using NUnit;
namespace Iteration5
{
    [TestFixture()]
    public class LocationTest
    {
        Player p;
        Location l;
        Item gem;

        [SetUp()]
        public void SetUp()
        {
            p = new Player("Fred", "the mighty programmer");
            l = new Location("room", "a big room");
            gem = new Item(new string[] { "gem" }, "gem", "a bright red gem");

            p.Loc = l;
            l.Inv.Put(gem);
        }

        [Test()]
        public void Locations_identify_themselves()
        {
            Assert.AreSame(l, l.Locate("room"));
        }
        [Test()]
        public void Locations_locate_items()
        {
            Assert.AreSame(gem, l.Locate("gem"));
        }

        [Test()]
        public void FullDescriptionTest()
        {

            Assert.AreEqual("You are in the room\na big room.\nIn this room you can
↪   see:\n\ta gem (gem)\n", l.FullDescription);

        }
    }
}
```

```
using System;
namespace Iteration5
{
    public class Player : Game_Object, IHaveInventory
    {
        Inventory _inventory = new Inventory();
        Location _loc;

        public Player(string name, string desc) : base(new string[] { "me",
↪  "inventory", "inv" }, name, desc)
        {

        }

        public Game_Object Locate(string id)
        {
            if (AreYou(id) == true)
            {
                return this;
            }
            else if (_loc != null)
            {
                return _loc.Locate(id);
            }
            else if (_loc.Inv.Fetch(id) != null)
            {
                return _loc.Inv.Fetch(id);


            }
            return _inventory.Fetch(id);
        }
        public Inventory Inv
        {
            get
            {
                return _inventory;
            }
        }
        public Location Loc
        {
            get
            {
                return _loc;
            }
            set
            {
                _loc = value;
            }
        }
        public override string FullDescription
        {
            get
```

```
53              {
54                  return ("You are " + Name + " " + base.FullDescription + "." + "\nYou
    ↪  are carrying\n" + Inv.ItemList);
55              }
56          }
57
58      }
59  }
60
```

```
using System;
using System.Xml.Linq;

namespace Iteration5
{
    public class PlayerTest
    {

        Item sword, computer;
        Player pl;
        Location l;
        Item box;

        [SetUp()]
        public void Constructor_PlayerTest()
        {

            pl = new Player("Fred", "the mighty programmer");

            sword = new Item(new string[] { "sword" }, "sword", "bronze");
            computer = new Item(new string[] { "pc" }, "computer", "small");
            l = new Location("room", "a small room");
            box = new Item(new string[] { "box" }, "big box", "brown");

            pl.Inv.Put(sword);
            pl.Inv.Put(computer);

            l.Inv.Put(box);
            pl.Loc = l;

        }
        [Test()]
        public void Test_Player_is_Identifiable()
        {
            Assert.IsTrue(pl.AreYou("me"));
        }
        [Test()]
        public void Test_Player_Locates_Items()
        {
            Assert.AreEqual(sword, pl.Locate("sword"));
            Assert.IsTrue(pl.Inv.HasItem("sword"));


            Assert.AreEqual(computer, pl.Locate("pc"));
            Assert.IsTrue(pl.Inv.HasItem("pc"));

        }
        [Test()]
        public void Test_Player_Locates_Itself()
        {
            Assert.AreEqual(pl, pl.Locate("me"));
            Assert.AreEqual(pl, pl.Locate("inventory"));

```

```
54          }
55          [Test()]
56          public void Test_Player_Locates_Nothing()
57          {
58              Assert.AreEqual(null, pl.Locate("food"));
59
60              Assert.AreEqual(null, pl.Locate("boat"));
61          }
62          [Test()]
63          public void Test_Player_full_Description()
64          {
65              Assert.AreEqual("You are Fred the mighty programmer.\n" + "You are
    ↪  carrying\n" + "\ta sword (sword)\n\ta computer (pc)\n", pl.FullDescription);
66          }
67          [Test()]
68          public void Players_locate_items_in_location()
69          {
70              Assert.AreSame(box, pl.Locate("box"));
71          }
72          [Test()]
73          public void Player_identify_location()
74          {
75              Assert.AreSame(l, pl.Locate("room"));
76          }
77      }
78  }
79
```

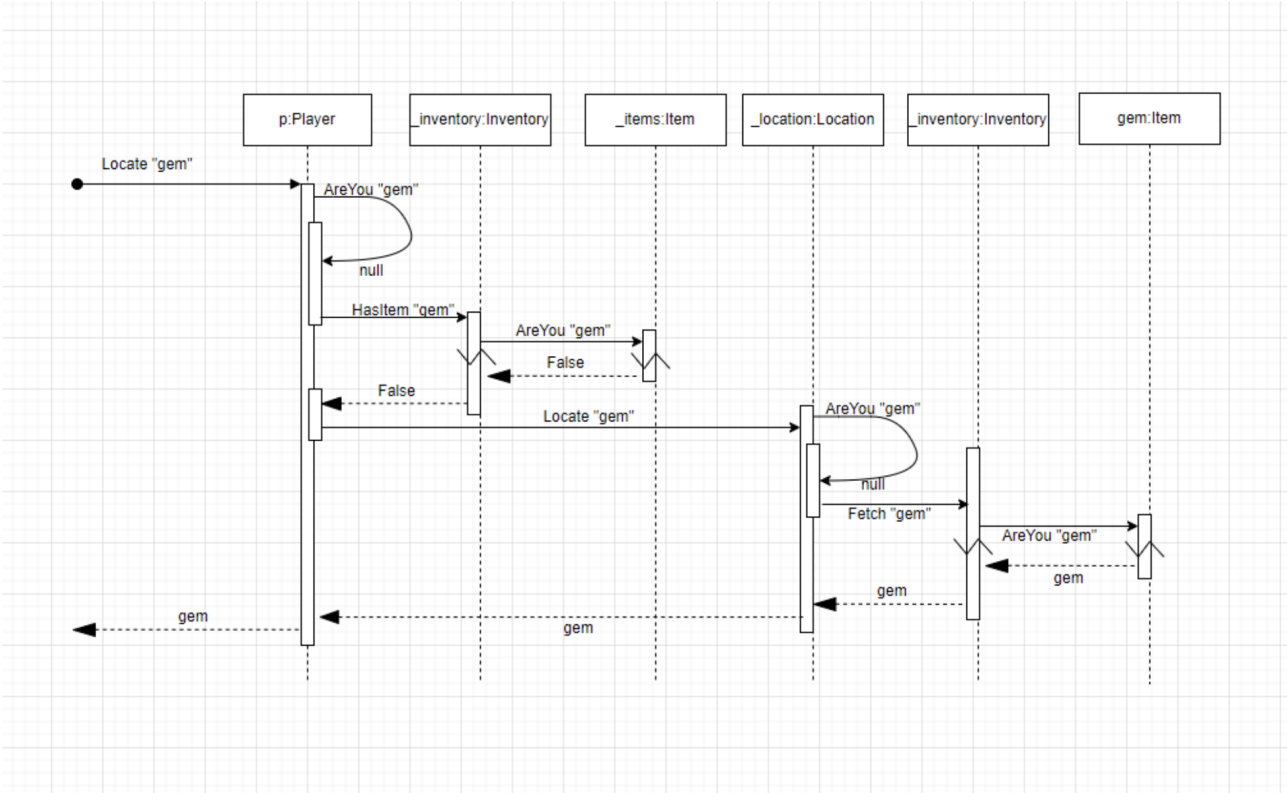
```
using System;
using System.ComponentModel;
using System.Numerics;

namespace Iteration5
{
    public class LookCommand : Command
    {



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

        }

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

            if (text.Length != 3 && text.Length != 5 && text.Length != 1)
            {
                return ("I don't know how to look like that.");
            }
            else
            {
                if (text[0] != "look")
                {
                    return ("Error in look input");
                }

                if (text.Length != 1 && 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 == 1)
                {
                    if (text[0] == "look")
                    {
                        container = p;
                        thingId = "room";
                    }

                }
                if (text.Length == 3)
                {
```

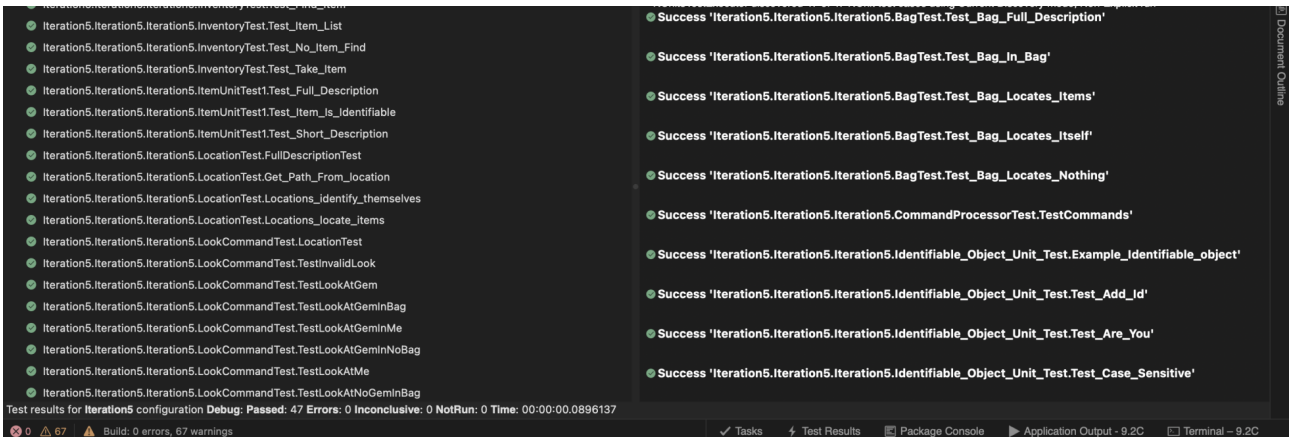
```
54                      container = p;
55                      thingId = text[2];
56                  }
57                  if (text.Length == 5)
58                  {
59                      container = FetchContainer(p, text[4]);
60                      if (container == null)
61                      {
62                          return ("I cannot find the " + text[4]);
63                      }
64                      thingId = text[2];
65
66                  }
67
68              }
69
70
71              return LookAtIn(thingId, container);
72
73          }
74          private IHaveInventory FetchContainer(Player p, string containerId)
75          {
76              return (IHaveInventory)p.Locate(containerId);
77          }
78          private string LookAtIn(string thingId, IHaveInventory container)
79          {
80              if (container.Locate(thingId) == null)
81              {
82                  return ("I cannot find the " + thingId + " in the " +
    ↪  container.Name);
83              }
84              else
85              {
86                  return container.Locate(thingId).FullDescription;
87              }
88
89          }
90
91      }
92  }
93
```

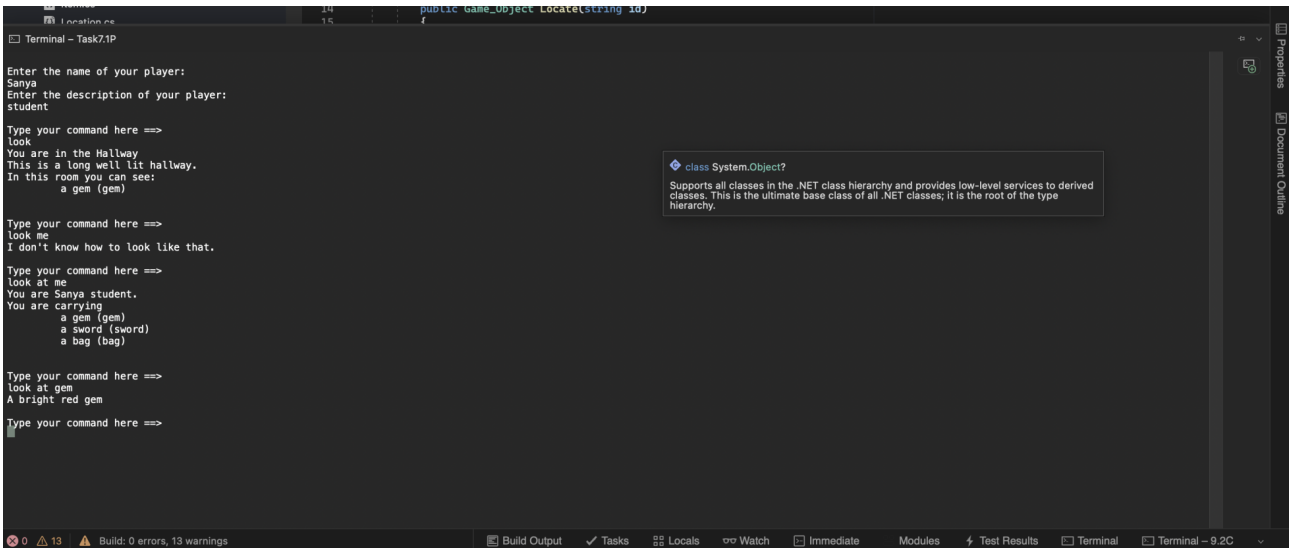
```
using System;
using System.ComponentModel;
using System.Xml.Linq;
using Iteration5;
using NUnit;
namespace Iteration5
{
    [TestFixture()]
    public class LookCommandTest
    {
        LookCommand l;
        Player p;
        Item gem;
        Bag b;
        Location lc;

        [SetUp()]
        public void Setup()
        {
            l = new LookCommand();
            p = new Player("Fred", "the mighty programmer");
            b = new Bag(new string[] { "bag" }, "leather bag", "small brown");
            gem = new Item(new string[] { "gem" }, "gem", "A bright red");
            lc = new Location("room", "a small room");
            p.Inv.Put(gem);
            p.Loc = lc;
        }
        [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);

```

```
50
51          }
52
53          [Test()]
54          public void TestLookAtGemInBag()
55          {
56              b.Inv.Put(gem);
57              p.Inv.Put(b);
58
59              Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪  "bag" }), gem.FullDescription);
60          }
61
62          [Test()]
63          public void TestLookAtGemInNoBag()
64          {
65              Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪  "bag" }), "I cannot find the bag");
66          }
67          [Test()]
68          public void TestLookAtNoGemInBag()
69          {
70              p.Inv.Put(b);
71
72              Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "gem", "in",
    ↪  "bag" }), "I cannot find the gem in the leather bag");
73          }
74          [Test()]
75          public void TestInvalidLook()
76          {
77              Assert.AreEqual(l.Execute(p, new string[] { "look", "around" }), "I don't
    ↪  know how to look like that.");
78              Assert.AreEqual(l.Execute(p, new string[] { "Hello", "Sanya", "Baweja"
    ↪  }), "Error in look input");
79              Assert.AreEqual(l.Execute(p, new string[] { "look", "at", "a", "at", "b"
    ↪  }), "What do you want to look in?");
80
81          }
82          [Test()]
83          public void LocationTest()
84
85          {
86              Assert.AreEqual(l.Execute(p, new string[] { "look" }),
    ↪  lc.FullDescription);
87          }
88
89          }
90  }
```

uses
Player
- _loc : Location
+ Loc : Location <<property>>
Location
- _inv : Inventory
+ Location(string name, string desc)
+ Locate(string id) : Game_Object
+ Full Description : string <<readonly, property>>
+ Inv : Inventory <<readonly, property>>
Game_Object
<<implements>>
<<interface>>
IHaveInventory


p:Player
_inventory:Inventory
_items:Item
_location:Location
_inventory:Inventory
gem:Item
Locate "gem"
AreYou "gem"
null
HasItem "gem"
AreYou "gem"
False
False
Locate "gem"
AreYou "gem"
null
Fetch "gem"
AreYou "gem"
gem
gem
gem
gem


Iteration5.Iteration5.Iteration5.InventoryTest.Test_Item_List
Iteration5.Iteration5.Iteration5.InventoryTest.Test_No_Item_Find
Iteration5.Iteration5.Iteration5.InventoryTest.Test_Take_Item
Iteration5.Iteration5.Iteration5.ItemUnitTest1.Test_Full_Description
Iteration5.Iteration5.Iteration5.ItemUnitTest1.Test_Item_Is_Identifiable
Iteration5.Iteration5.Iteration5.ItemUnitTest1.Test_Short_Description
Iteration5.Iteration5.Iteration5.LocationTest.FullDescriptionTest
Iteration5.Iteration5.Iteration5.LocationTest.Get_Path_From_location
Iteration5.Iteration5.Iteration5.LocationTest.Locations_identify_themselves
Iteration5.Iteration5.Iteration5.LocationTest.Locations_locate_items
Iteration5.Iteration5.Iteration5.LookCommandTest.LocationTest
Iteration5.Iteration5.Iteration5.LookCommandTest.TestInvalidLook
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtGem
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtGemInBag
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtGemInMe
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtGemInNoBag
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtMe
Iteration5.Iteration5.Iteration5.LookCommandTest.TestLookAtNoGemInBag
Success 'Iteration5.Iteration5.Iteration5.BagTest.Test_Bag_Full_Description'
Success 'Iteration5.Iteration5.Iteration5.BagTest.Test_Bag_In_Bag'
Success 'Iteration5.Iteration5.Iteration5.BagTest.Test_Bag_Locates_Items'
Success 'Iteration5.Iteration5.Iteration5.BagTest.Test_Bag_Locates_Itself'
Success 'Iteration5.Iteration5.Iteration5.BagTest.Test_Bag_Locates_Nothing'
Success 'Iteration5.Iteration5.Iteration5.CommandProcessorTest.TestCommands'
Success 'Iteration5.Iteration5.Iteration5.Identifiable_Object_Unit_Test.Example_Identifiable_object'
Success 'Iteration5.Iteration5.Iteration5.Identifiable_Object_Unit_Test.Test_Add_Id'
Success 'Iteration5.Iteration5.Iteration5.Identifiable_Object_Unit_Test.Test_Are_You'
Success 'Iteration5.Iteration5.Iteration5.Identifiable_Object_Unit_Test.Test_Case_Sensitive'
Test results for Iteration5 configuration Debug: Passed: 47 Errors: 0 Inconclusive: 0 NotRun: 0 Time: 00:00:00.0896137
0 67
Build: 0 errors, 67 warnings
Tasks
Test Results
Package Console
Application Output - 9.2C
Terminal – 9.2C
Document Outline


public Game_Object Locate(string id)
Terminal – Task7.1P
Enter the name of your player:
Sanya
Enter the description of your player:
student
Type your command here ==>
look
You are in the Hallway
This is a long well lit hallway.
In this room you can see:
a gem (gem)
Type your command here ==>
look me
I don't know how to look like that.
Type your command here ==>
look at me
You are Sanya student.
You are carrying
a gem (gem)
a sword (sword)
a bag (bag)
Type your command here ==>
look at gem
A bright red gem
Type your command here ==>
class System.Object?
Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.
Build: 0 errors, 13 warnings
Build Output
Tasks
Locals
Watch
Immediate
Modules
Test Results
Terminal
Terminal – 9.2C
Properties
Document Outline