Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 2 - Players Items and Inventory

PDF generated at 12:32 on Thursday 7th September, 2023

```
using CaseStudy;
using System;

namespace CaseStudy
{
    public abstract class GameObject : IdentifiableObject
    {
        private string _description, _name;

        public GameObject(string[] ids, string name, string desc) : base(ids)
        {
            _name = name;
            _description = desc;
        }

        public string Name
        {
            get
            {
                return _name;
            }
        }

        public virtual string FullDescription
        {
            get
            {
                return _description;
            }
        }

        public string ShortDescription
        {
            get
            {
                return $"{_name} ({FirstID()})";
            }
        }

    }
}
```

```
namespace CaseStudy
{
    public class Player : GameObject, IHaveInventory
    {
        private Inventory _inventory;
        private Location _location;

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

        public GameObject Locate(string id)
        {
            if (AreYou(id))
            {
                return this;
            }
            GameObject obj = _inventory.Fetch(id);
            if (obj != null)
            {
                return obj;
            }
            if (_location != null)
            {
                obj = _location.Locate(id);
                return obj;
            }
            else
            {
                return null;
            }
        }

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

        public Inventory Inventory
        {
            get
            {
                return _inventory;
            }
        }

        public Location Location
```

```
52          {
53              get
54              {
55                  return _location;
56              }
57              set
58              {
59                  _location = value;
60              }
61          }
62      }
63  }
```

```
using CaseStudy;

namespace CaseStudyTest
{
    public class PlayerTest
    {
        private Player _player;
        private Item _shovel;
        private Item _sword;
        private Location _loc;

        [SetUp]
        public void Setup()
        {
            _player = new Player("Vu Duc Tran", "Swinburne Student");
            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a shovel");
            _sword = new Item(new string[] { "sword" }, "a sword", "This is a Sword");
            _loc = new Location("School", "Swinburne University");

            _player.Inventory.Put(_sword);
            _player.Location = _loc;
            _loc.Items.Put(_shovel);
        }

        [Test]
        public void TestPlayerIdentifiable()
        {
            Assert.That(_player.AreYou("me"), Is.EqualTo(true), "Test player identifiable");
            Assert.That(_player.AreYou("inventory"), Is.EqualTo(true), "Test player identifiable");
        }

        [Test]
        public void TestPlayerLocateItems()
        {
            Assert.That(_player.Locate("sword"), Is.EqualTo(_sword), "Test player locate items");
        }

        [Test]
        public void TestPlayerLocateItself()
        {
            Assert.That(_player.Locate("me"), Is.EqualTo(_player), "Test player locate it self");
            Assert.That(_player.Locate("inventory"), Is.EqualTo(_player), "Test player locate it self");
        }

        [Test]
        public void TestPlayerLocateNothing()
```

```
47          {
48              Assert.That(_player.Locate("mirror"), Is.EqualTo(null), "Test player
    ↪  locate nothing");
49          }
50
51          [Test]
52          public void TestPlayerFullDescription()
53          {
54              Assert.That(_player.FullDescription, Is.EqualTo($"You are Vu Duc Tran,
    ↪  Swinburne Student.\n You are carrying:\n {_player.Inventory.ItemList}"), "Test
    ↪  player full descrition");
55          }
56
57          [Test]
58          public void PlayerLocateItemsInLocation()
59          {
60              Assert.That(_player.Location.FullDescription, Is.EqualTo("You are at:
    ↪  School (location)\n\nItems at this location:\na shovel (shovel)\n"));
61          }
62      }
63  }
```

```
using System;

namespace CaseStudy
{
    public class Item : GameObject
    {
        public Item(string[] idents, string name, string desc) : base(idents, name,
↪   desc)
        {
        }
    }
}
```

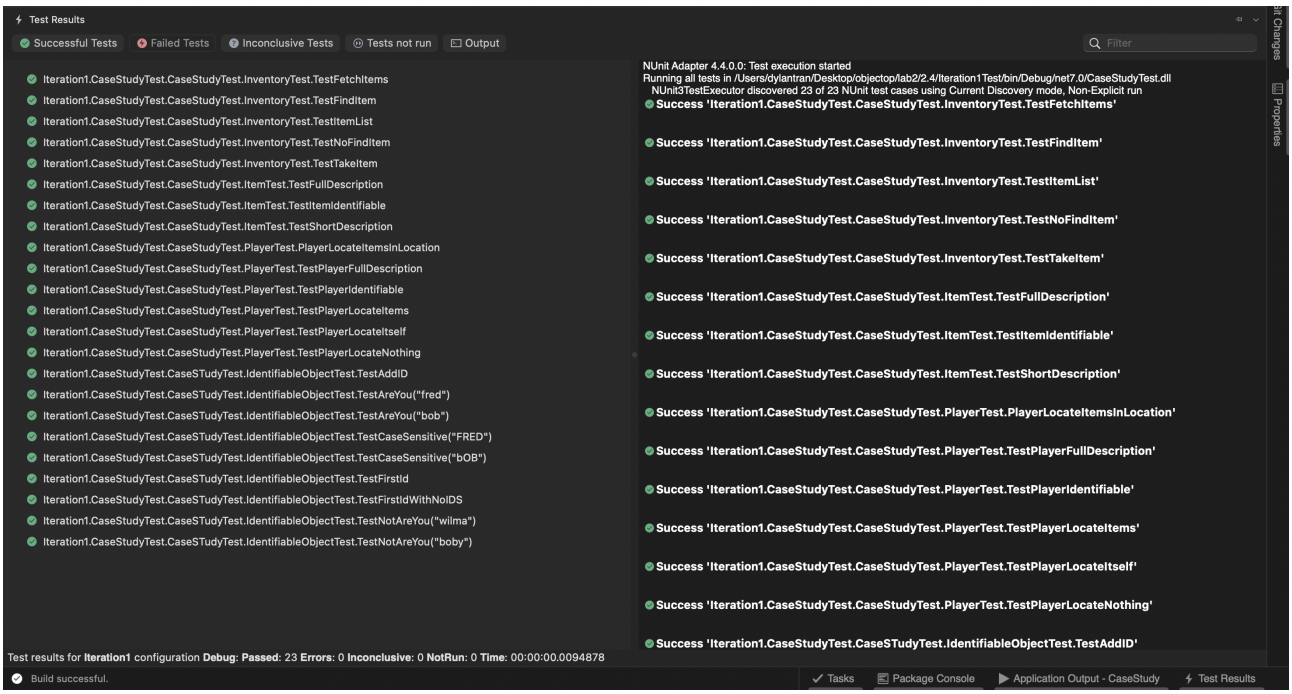
```
using System;
using NUnit.Framework;
using CaseStudy;

namespace CaseStudyTest
{
    public class ItemTest
    {
        private Item _shovel;
        private Item _sword;

        [SetUp]
        public void Setup()
        {
            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a shovel");
            _sword = new Item(new string[] { "sword" }, "a sword", "This is a sword");
        }

        [Test]
        public void TestItemIdentifiable()
        {
            Assert.That(_shovel.AreYou("sword"), Is.EqualTo(false), "Test not correct Identifiable");
            Assert.That(_sword.AreYou("sword"), Is.EqualTo(true), "Test Identifiable");
        }

        [Test]
        public void TestShortDescription()
        {
            Assert.That(_shovel.ShortDescription, Is.EqualTo("a shovel (shovel)"), "Test Short Description");
            Assert.That(_sword.ShortDescription, Is.Not.EqualTo("a shovel (shovel)"), "Test not correct Short Description");
        }

        [Test]
        public void TestFullDescription()
        {
            Assert.That(_shovel.FullDescription, Is.EqualTo("This is a shovel"), "Test Full Description");
            Assert.That(_sword.FullDescription, Is.Not.EqualTo("This is a shovel"), "Test not correct Full Description");

        }
    }
}
```

```
using System;

namespace CaseStudy
{
    public class Inventory
    {
        private List<Item> _items;

        public Inventory()
        {
            _items = new List<Item>();
        }

        public bool HasItem(string id)
        {
            foreach (Item itm in _items)
            {
                if (itm.AreYou(id))
                {
                    return true;
                }

            }
            return false;
        }

        public void Put(Item itm)
        {
            _items.Add(itm);
        }

        public Item Take(string id)
        {
            Item takeitem = this.Fetch(id);
            _items.Remove(takeitem);
            return takeitem;
        }

        public Item Fetch(string id)
        {
            foreach (Item itm in _items)
            {
                if (itm.AreYou(id))
                {
                    return itm;
                }
            }
            return null;
        }

        public string ItemList
        {
            get
```

```
54              {
55                  string listitm = "";
56                  foreach (Item i in _items)
57                  {
58                      listitm = listitm + i.ShortDescription + "\n";
59                  }
60                  return listitm;
61              }
62          }
63      }
64  }
```

```
using System;
using CaseStudy;
using NUnit.Framework;

namespace CaseStudyTest
{
    public class InventoryTest
    {
        private Item _shovel;
        private Item _sword;
        private Inventory _ive;
        [SetUp]
        public void Setup()
        {
            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a shovel");
            _sword = new Item(new string[] { "sword" }, "a sword", "This is a sword");
            _ive = new Inventory();
            _ive.Put(_shovel);
            _ive.Put(_sword);
        }

        [Test]
        public void TestFindItem()
        {
            Assert.That(_ive.HasItem(_shovel.FirstID()), Is.EqualTo(true), "Test Find Item");
        }

        [Test]
        public void TestNoFindItem()
        {
            Assert.That(_ive.HasItem("gem"), Is.EqualTo(false), "Test No Find Item");
        }

        [Test]
        public void TestFetchItems()
        {
            Assert.That(_ive.Fetch("shovel"), Is.EqualTo(_shovel), "Test Correct Item Returned");
            Assert.That(_ive.HasItem("shovel"), Is.EqualTo(true), "Test That Item Remains In The Inventory");
        }

        [Test]
        public void TestTakeItem()
        {
            Assert.That(_ive.Take("shovel"), Is.EqualTo(_shovel), "Test Take Item");
            Assert.That(_ive.HasItem("shovel"), Is.EqualTo(false), "Test Take Item");
        }

        [Test]
```

```
49          public void TestItemList()
50          {
51              Assert.That(_ive.ItemList, Is.EqualTo("a shovel (shovel)\n" + "a sword
    ↪  (sword)\n"), "Test Item List");
52          }
53
54      }
55  }
```


Test Results
Successful Tests
Failed Tests
Inconclusive Tests
Tests not run
Output
Filter
Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestFetchItems
Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestFindItem
Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestItemList
Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestNoFindItem
Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestTakeItem
Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestFullDescription
Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestItemIdentifiable
Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestShortDescription
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.PlayerLocateItemsInLocation
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerFullDescription
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerIdentifiable
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateItems
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateItself
Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateNothing
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestAddID
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestAreYou("fred")
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestAreYou("bob")
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestCaseSensitive("FRED")
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestCaseSensitive("bOB")
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestFirstId
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestFirstIdWithNoIDS
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestNotAreYou("wilma")
Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestNotAreYou("boby")
NUnit Adapter 4.4.0.0: Test execution started
Running all tests in /Users/dylantran/Desktop/objectop/lab2/2.4/Iteration1Test/bin/Debug/net7.0/CaseStudyTest.dll
NUnit3TestExecutor discovered 23 of 23 NUnit test cases using Current Discovery mode, Non-Explicit run
Success 'Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestFetchItems'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestFindItem'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestItemList'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestNoFindItem'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.InventoryTest.TestTakeItem'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestFullDescription'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestItemIdentifiable'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.ItemTest.TestShortDescription'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.PlayerLocateItemsInLocation'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerFullDescription'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerIdentifiable'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateItems'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateItself'
Success 'Iteration1.CaseStudyTest.CaseStudyTest.PlayerTest.TestPlayerLocateNothing'
Success 'Iteration1.CaseStudyTest.CaseSTudyTest.IdentifiableObjectTest.TestAddID'
Test results for Iteration1 configuration Debug: Passed: 23 Errors: 0 Inconclusive: 0 NotRun: 0 Time: 00:00:00.0094878
Build successful.
Tasks
Package Console
Application Output - CaseStudy
Test Results
Git Changes
Properties