Swinburne University of Technology

Object Oriented Programming (2021 S1)

Doubtfire Submission

# Pass Task 4.2: Case Study Iteration 2

*Submitted By:*
Kevin Pham
102683681
2021/05/20 11:10

*Tutor:*
Tien Pham

May 20, 2021

Produced by Doubtfire

```
using System;
using System.Collections.Generic;
using System.Text;
using Case_Study;

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

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

        public string Name
        {
            get
            {
                return _name;
            }
        }

        public string ShortDescription
        {
            get
            {
                return _name + " " + FirstId();
            }
        }

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

```
using System;
using System.Collections.Generic;
using System.Text;

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

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

        public override string FullDescription
        {
            get
            {
                return _inventory.ItemList;
            }
        }

        public Inventory Inventory
        {
            get
            {
                return _inventory;
            }
        }

    }
}
```

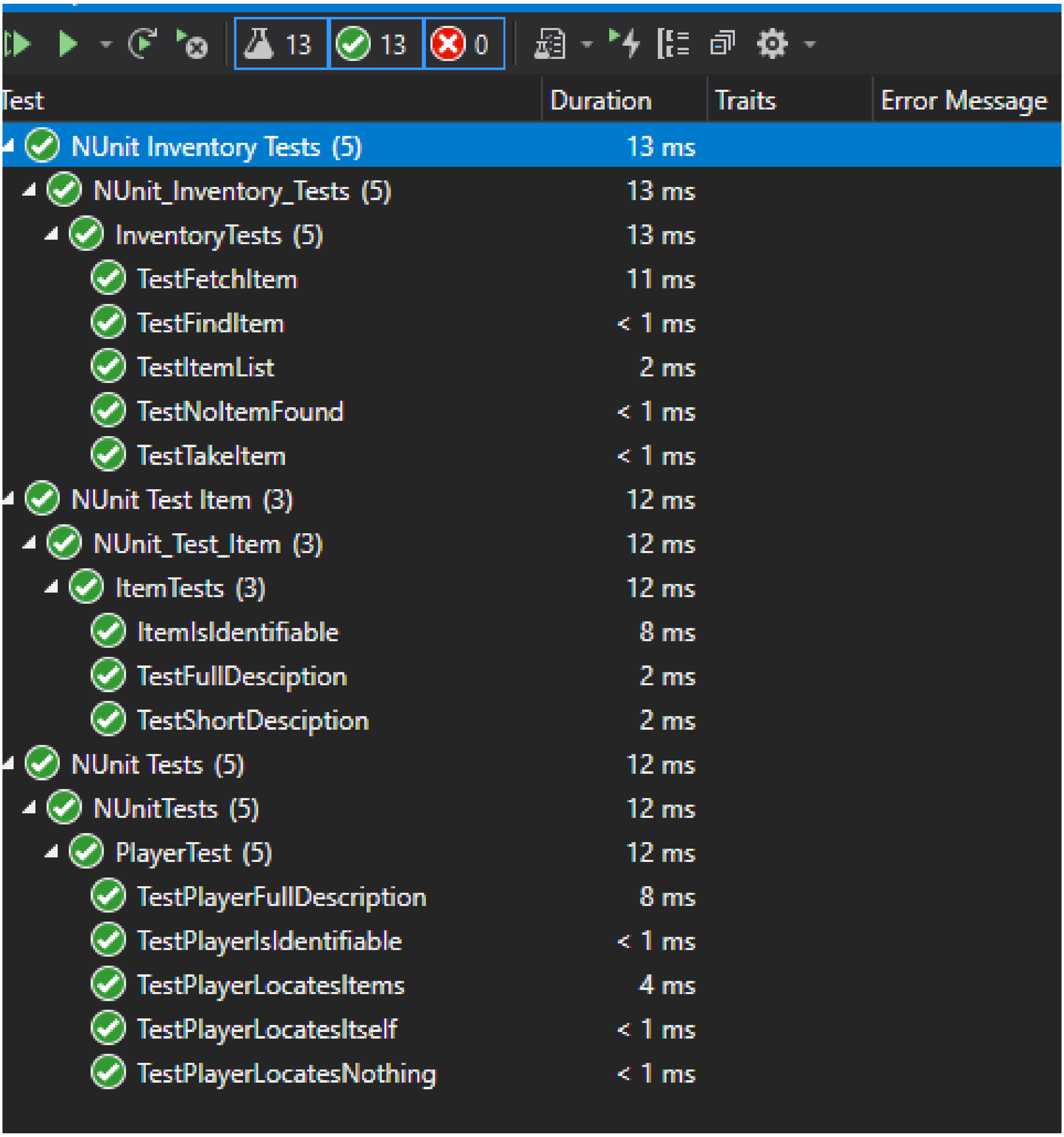
```
/*
 * File: NUnitTests.cs
 * Author: Kevin Pham
 * Date: 15/03/2021
 * Unit: COS20007 Object Oriented Programming
 * Institution: Swinburne University of Technology
 */

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NUnit.Framework;
using Case_Study;

namespace NUnitTests
{
    [TestFixture]
    public class PlayerTest
    {
        public Inventory inventory;
        public Item shovel;
        public Player player;

        [SetUp]
        public void Setup()
        {
            inventory = new Inventory();
            shovel = new Item(new string[] { "shovel", "spade" }, "a shovel", "This is a mighty fine...");
            player = new Player("hero", "me");
        }

        [Test]
        public void TestPlayerIsIdentifiable()
        {
            Assert.IsTrue(player.AreYou("me"));
        }

        [Test]
        public void TestPlayerLocatesItems()
        {
            inventory.Put(shovel);
            player.Locate("shovel");
            Assert.AreEqual("\ta shovel shovel\n", inventory.ItemList);
            Assert.IsTrue(inventory.HasItem("shovel"));
        }

        [Test]
        public void TestPlayerLocatesItself()
        {
            Assert.AreEqual(player.Locate("me"), player);
        }

        [Test]
        public void TestPlayerLocatesNothing()
        {
            player.AreYou("shield");
            Assert.IsNull(inventory.Fetch("shield"));
        }

        [Test]
        public void TestPlayerFullDescription()
        {
            inventory.Put(shovel);
            StringAssert.Contains("You are carrying: " + player.FullDescription, "You are carrying: " + inventory.ItemList);
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

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

        }
    }
}
```

```
using NUnit.Framework;
using Case_Study;

namespace NUnit_Test_Item
{
    public class ItemTests
    {
        public Item items;

        [SetUp]
        public void Setup()
        {
            items = new Item(new string[] { "shovel", "spade" }, "a shovel", "This is a might fine...");
        }

        [Test]
        public void ItemIsIdentifiable()
        {
            Assert.IsTrue(items.AreYou("shovel"));
        }

        [Test]
        public void TestShortDesciption()
        {
            Assert.AreEqual(items.ShortDescription, "a shovel shovel");
        }

        [Test]
        public void TestFullDesciption()
        {
            Assert.AreEqual(items.FullDescription, "This is a might fine...");
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Case_Study
{
    public class Inventory
    {
        private List<Item> _items = new List<Item>();
        public Inventory() { }
        public bool HasItem(string id)
        {
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    return true;
                }
            }
            return false;
        }

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

        public Item Take(string id)
        {
            Item inStore = null;
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    inStore = item;
                }
            }
            _items.Remove(inStore);
            return inStore;
        }

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

        public string ItemList
        {
            get
            {
                string ListItem = "";
                foreach (Item item in _items)
                {
                    ListItem = "\t" + item.ShortDescription + "\n";
                }
                if (ListItem == null)
                {
                    return "There is no such item in the list";
                }
                return ListItem;
            }
        }
    }
}
```

```
using NUnit.Framework;
using Case_Study;


namespace NUnit_Inventory_Tests
{
    public class InventoryTests
    {
        public Inventory items;
        public Item shovel;

        [SetUp]
        public void Setup()
        {
            items = new Inventory();
            shovel = new Item(new string[] { "shovel"}, "a shovel", "This is a mighty fine...");
        }

        [Test]
        public void TestFindItem()
        {
            items.Put(shovel);
            Assert.IsTrue(items.HasItem("shovel"));
        }


        [Test]
        public void TestNoItemFound()
        {
            items.Put(shovel);
            Assert.IsFalse(items.HasItem("sword"));
        }

        [Test]
        public void TestFetchItem()
        {
            items.Put(shovel);
            Assert.AreEqual(shovel, items.Fetch("shovel"));
            Assert.IsTrue(items.HasItem("shovel"));
        }

        [Test]
        public void TestTakeItem()
        {
            items.Put(shovel);
            Assert.AreEqual(shovel, items.Take("shovel"));
            Assert.IsFalse(items.HasItem("shovel"));
        }

        [Test]
        public void TestItemList()
        {
            items.Put(shovel);
            Assert.AreEqual("\t" + "a shovel shovel" + "\n", items.ItemList);
        }
    }
}
```



| Test | Duration | Traits | Error Message |
|---|---|---|---|
| NUnit Inventory Tests (5) | 13 ms | | |
| NUnit_Inventory_Tests (5) | 13 ms | | |
| InventoryTests (5) | 13 ms | | |
| TestFetchItem | 11 ms | | |
| TestFindItem | < 1 ms | | |
| TestItemList | 2 ms | | |
| TestNoItemFound | < 1 ms | | |
| TestTakeItem | < 1 ms | | |
| NUnit Test Item (3) | 12 ms | | |
| NUnit_Test_Item (3) | 12 ms | | |
| ItemTests (3) | 12 ms | | |
| ItemIsIdentifiable | 8 ms | | |
| TestFullDesciption | 2 ms | | |
| TestShortDesciption | 2 ms | | |
| NUnit Tests (5) | 12 ms | | |
| NUnitTests (5) | 12 ms | | |
| PlayerTest (5) | 12 ms | | |
| TestPlayerFullDescription | 8 ms | | |
| TestPlayerIsIdentifiable | < 1 ms | | |
| TestPlayerLocatesItems | 4 ms | | |
| TestPlayerLocatesItself | < 1 ms | | |
| TestPlayerLocatesNothing | < 1 ms | | |