

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



```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using Case_Study;
5
6  namespace Case_Study
7  {
8      public abstract class GameObject : IdentifiableObject
9      {
10         private string _description;
11         private string _name;
12
13         public GameObject(string[] ids, string name, string desc) : base(ids)
14         {
15             _description = desc;
16             _name = name;
17         }
18
19         public string Name
20         {
21             get
22             {
23                 return _name;
24             }
25         }
26
27         public string ShortDescription
28         {
29             get
30             {
31                 return _name + " " + FirstId();
32             }
33         }
34
35         public virtual string FullDescription
36         {
37             get
38             {
39                 return _description;
40             }
41         }
42     }
43 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace Case_Study
6  {
7      public class Player : GameObject
8      {
9          private Inventory _inventory;
10         public Player(string name, string desc) : base(new string[] { "me",
11             ↪ "inventory" }, name, desc)
12         {
13             _inventory = new Inventory();
14         }
15
16         public GameObject Locate(string id)
17         {
18             if (this.AreYou(id))
19             {
20                 return this;
21             }
22             else if (this._inventory.HasItem(id))
23             {
24                 return this._inventory.Fetch(id);
25             }
26             return null;
27         }
28
29         public override string FullDescription
30         {
31             get
32             {
33                 return _inventory.ItemList;
34             }
35         }
36
37         public Inventory Inventory
38         {
39             get
40             {
41                 return _inventory;
42             }
43         }
44     }
45 }
```

```
1  /*
2   * File: NUnitTests.cs
3   * Author: Kevin Pham
4   * Date: 15/03/2021
5   * Unit: CDS20007 Object Oriented Programming
6   * Institution: Swinburne University of Technology
7   */
8
9  using System;
10 using System.Collections.Generic;
11 using System.Linq;
12 using System.Text;
13 using System.Threading.Tasks;
14 using NUnit.Framework;
15 using Case_Study;
16
17 namespace NUnitTests
18 {
19     [TestFixture]
20     public class PlayerTest
21     {
22         public Inventory inventory;
23         public Item shovel;
24         public Player player;
25
26         [SetUp]
27         public void Setup()
28         {
29             inventory = new Inventory();
30             shovel = new Item(new string[] { "shovel", "spade" }, "a shovel", "This
31                 ↪ is a mighty fine...");
32             player = new Player("hero", "me");
33         }
34
35         [Test]
36         public void TestPlayerIsIdentifiable()
37         {
38             Assert.IsTrue(player.AreYou("me"));
39         }
40
41         [Test]
42         public void TestPlayerLocatesItems()
43         {
44             inventory.Put(shovel);
45             player.Locate("shovel");
46             Assert.AreEqual("\ta shovel shovel\n", inventory.ItemList);
47             Assert.IsTrue(inventory.HasItem("shovel"));
48         }
49
50         [Test]
51         public void TestPlayerLocatesItself()
52         {
53             Assert.AreEqual(player.Locate("me"), player);
54         }
55     }
56 }
```

```
53     }
54
55     [Test]
56     public void TestPlayerLocatesNothing()
57     {
58         player.AreYou("shield");
59         Assert.IsNull(inventory.Fetch("shield"));
60     }
61
62     [Test]
63     public void TestPlayerFullDescription()
64     {
65         inventory.Put(shovel);
66         StringAssert.Contains("You are carrying: " + player.FullDescription,
67                               ↪ "You are carrying: " + inventory.ItemList);
67     }
68 }
69 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace Case_Study
6 {
7     public class Item : GameObject
8     {
9         public Item(string[] idents, string name, string desc) : base(idents, name,
10             ↪ desc)
11         {
12         }
13     }
14 }
```

```
1  using NUnit.Framework;
2  using Case_Study;
3
4  namespace NUnit_Test_Item
5  {
6      public class ItemTests
7      {
8          public Item items;
9
10         [SetUp]
11         public void Setup()
12         {
13             items = new Item(new string[] { "shovel", "spade" }, "a shovel", "This
14                 ↪ is a might fine...");
15         }
16
17         [Test]
18         public void ItemIsIdentifiable()
19         {
20             Assert.IsTrue(items.AreYou("shovel"));
21         }
22
23         [Test]
24         public void TestShortDescription()
25         {
26             Assert.AreEqual(items.ShortDescription, "a shovel shovel");
27         }
28
29         [Test]
30         public void TestFullDescription()
31         {
32             Assert.AreEqual(items.FullDescription, "This is a might fine...");
33         }
34     }
```



































```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace Case_Study
6  {
7      public class Inventory
8      {
9          private List<Item> _items = new List<Item>();
10         public Inventory() { }
11         public bool HasItem(string id)
12         {
13             foreach (Item item in _items)
14             {
15                 if (item.AreYou(id))
16                 {
17                     return true;
18                 }
19             }
20             return false;
21         }
22
23         public void Put(Item itm)
24         {
25             _items.Add(itm);
26         }
27
28         public Item Take(string id)
29         {
30             Item inStore = null;
31             foreach (Item item in _items)
32             {
33                 if (item.AreYou(id))
34                 {
35                     inStore = item;
36                 }
37             }
38             _items.Remove(inStore);
39             return inStore;
40         }
41
42         public Item Fetch(string id)
43         {
44             foreach (Item item in _items)
45             {
46                 if (item.AreYou(id))
47                 {
48                     return item;
49                 }
50             }
51             return null;
52         }
53     }
```



```
54     public string ItemList
55     {
56         get
57         {
58             string ListItem = "";
59             foreach (Item item in _items)
60             {
61                 ListItem = "\t" + item.ShortDescription + "\n";
62             }
63             if (ListItem == null)
64             {
65                 return "There is no such item in the list";
66             }
67             return ListItem;
68         }
69     }
70 }
71 }
```

```
1  using NUnit.Framework;
2  using Case_Study;
3
4
5  namespace NUnit_Inventory_Tests
6  {
7      public class InventoryTests
8      {
9          public Inventory items;
10         public Item shovel;
11
12         [SetUp]
13         public void Setup()
14         {
15             items = new Inventory();
16             shovel = new Item(new string[] { "shovel"}, "a shovel", "This is a
17                 ↪ mighty fine...");
18
19         [Test]
20         public void TestFindItem()
21         {
22             items.Put(shovel);
23             Assert.IsTrue(items.HasItem("shovel"));
24         }
25
26
27         [Test]
28         public void TestNoItemFound()
29         {
30             items.Put(shovel);
31             Assert.IsFalse(items.HasItem("sword"));
32         }
33
34         [Test]
35         public void TestFetchItem()
36         {
37             items.Put(shovel);
38             Assert.AreEqual(shovel, items.Fetch("shovel"));
39             Assert.IsTrue(items.HasItem("shovel"));
40         }
41
42         [Test]
43         public void TestTakeItem()
44         {
45             items.Put(shovel);
46             Assert.AreEqual(shovel, items.Take("shovel"));
47             Assert.IsFalse(items.HasItem("shovel"));
48         }
49
50         [Test]
51         public void TestItemList()
52         {
```

```
53         items.Put(shovel);
54         Assert.AreEqual("\t" + "a shovel shovel" + "\n", items.ItemList);
55     }
56 }
57 }
```

<div>     <div>  13  13  0 </div>      </div>			
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		
 ItemsIsIdentifiable	8 ms		
 TestFullDescription	2 ms		
 TestShortDescription	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		