

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 2 - Players Items and Inventory

PDF generated at 15:26 on Thursday 24th August, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public abstract class GameObject : IdentifiableObject
10     {
11         private string _description;
12         private string _name;
13
14         public GameObject(string[] ids, string name, string desc) : base(ids)
15         {
16             _description = desc;
17             _name = name;
18         }
19
20         public string Name
21         {
22             get
23             {
24                 return _name;
25             }
26         }
27
28         public string ShortDescription
29         {
30             get
31             {
32                 return _name + " " + "(" + FirstId + ")";
33             }
34         }
35
36         public virtual string FullDescription
37         {
38             get
39             {
40                 return _description;
41             }
42         }
43     }
44 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public class Player : GameObject
10     {
11         private Inventory _inventory;
12
13         public Player(string name, string desc) : base(new string[] { "me",
14             "inventory" }, name, desc)
15         {
16             _inventory = new Inventory();
17         }
18
19         public GameObject Locate(string id)
20         {
21             if (AreYou(id))
22             {
23                 return this;
24             }
25             return _inventory.Fetch(id);
26         }
27
28         public override string FullDescription
29         {
30             get
31             {
32                 return $"You are {this.Name} {base.FullDescription}\nYou are
33             carrying:\n{n{_inventory.ItemList}}";
34             }
35         }
36
37         public Inventory Inventory
38         {
39             get
40             {
41                 return _inventory;
42             }
43         }
44     }
45 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using NUnit.Framework;
7  using Iteration2;
8
9  namespace IterationTest2
10 {
11     [TestFixture()]
12     public class PlayerTest
13     {
14         private Player _player;
15         private Item _item;
16
17         [SetUp()]
18         public void SetUp()
19         {
20             _player = new Player("Siam", "a Gamer");
21             _item = new Item(new string[] { "shovel", "spade" }, "a shovel", "This
22             ↵ might be fine...");  
             _player.Inventory.Put(_item);
23         }
24
25         [Test()]
26         public void IdentifiablePlayer()
27         {
28             Assert.IsTrue(_player.AreYou("me"));
29         }
30
31         [Test()]
32         public void LocateItems()
33         {
34             Assert.AreEqual(_item, _player.Locate("shovel"));
35             Assert.IsTrue(_player.Inventory.HasItem("shovel"));
36         }
37
38         [Test()]
39         public void LocateItself()
40         {
41             Assert.AreEqual(_player, _player.Locate("me"));
42             Assert.AreEqual(_player, _player.Locate("inventory"));
43         }
44
45         [Test()]
46         public void LocateNothing()
47         {
48             Assert.AreEqual(null, _player.Locate("abc"));
49         }
50
51         [Test()]
52         public void FullDesc()
```

```
53         {
54             Assert.AreEqual("You are Siam a Gamer\nYou are carrying:\n\ta shovel
55             (shovel)\n", _player.FullDescription);
56         }
57     }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public class Item : GameObject
10     {
11         public Item(string[] idents, string name, string desc) : base(idents, name,
12             desc)
13         {
14         }
15     }
16 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using NUnit.Framework;
7  using Iteration2;
8
9
10 namespace IterationTest2
11 {
12
13     [TestFixture()]
14     public class ItemTest
15     {
16         private Item _item;
17
18         [SetUp()]
19
20         public void SetUp()
21         {
22             _item = new Item(new string[] { "shovel", "spade" }, "a shovel", "This
23             ↪ might be fine...");
24         }
25
26         [Test()]
27
28         public void ItemIdentifiable()
29         {
30             Assert.IsTrue(_item.AreYou("shovel"));
31             Assert.IsTrue(_item.AreYou("spade"));
32         }
33
34         [Test()]
35
36         public void ShortDesc()
37         {
38             Assert.AreEqual("a shovel (shovel)", _item.ShortDescription);
39         }
40
41         [Test()]
42
43         public void FullDesc()
44         {
45             Assert.AreEqual("This might be fine...", _item.FullDescription);
46         }
47     }
}
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public class Inventory
10     {
11         private List<Item> _items = new List<Item>();
12
13         public Inventory() { }
14
15         public bool HasItem(string id)
16         {
17             foreach (Item item in _items)
18             {
19                 if (itm.AreYou(id))
20                     return true;
21             }
22             return false;
23         }
24
25         public void Put(Item item)
26         {
27             _items.Add(item);
28         }
29
30         public Item Take(string id)
31         {
32
33             foreach (Item item in _items)
34             {
35                 if (itm.AreYou(id))
36                 {
37                     _items.Remove(item);
38                     return item;
39                 }
40             }
41             return null;
42         }
43
44         public Item Fetch(string id)
45         {
46             foreach (Item item in _items)
47             {
48                 if (itm.AreYou(id))
49                 {
50                     return item;
51                 }
52             }
53             return null;
54 }
```

```
54     }
55
56     public string ItemList
57     {
58         get
59         {
60             string itemList = "";
61
62             foreach (Item item in _items)
63             {
64                 itemList += "\t" + item.ShortDescription + "\n";
65             }
66             if (itemList == "")
67             {
68                 itemList = "\tnothing\n";
69             }
70             return itemList;
71         }
72     }
73 }
74 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using NUnit.Framework;
7  using Iteration2;
8
9  namespace IterationTest2
10 {
11     [TestFixture()]
12     public class InventoryTest
13     {
14         private Inventory _inventory;
15         private Item _item;
16
17         [SetUp()]
18
19         public void SetUp()
20         {
21             _inventory = new Inventory();
22             _item = new Item(new string[] { "shovel", "spade" }, "a shovel", "This
23 ← might be fine...");
24             _inventory.Put(_item);
25         }
26
27         [Test()]
28
29         public void TestPutItem()
30         {
31             Assert.IsTrue(_inventory.HasItem("shovel"));
32         }
33
34         [Test()]
35
36         public void TestNoItem()
37         {
38             Assert.IsFalse(_inventory.HasItem("gun"));
39         }
40
41         [Test()]
42
43         public void TestFetchItem()
44         {
45             Assert.AreEqual(_item, _inventory.Fetch("shovel"));
46             Assert.IsTrue(_inventory.HasItem("shovel"));
47         }
48
49         [Test()]
50
51         public void TestTakeItem()
52         {
53             Assert.AreEqual(_item, _inventory.Take("shovel"));
54         }
55     }
56 }
```

```
53             Assert.IsFalse(_inventory.HasItem("shovel"));
54         }
55
56     [Test()]
57
58     public void TestFullList()
59     {
60         Assert.AreEqual("\ta shovel (shovel)\n", _inventory.ItemList);
61     }
62 }
63 }
```

The screenshot shows the Visual Studio Test Explorer window. The left pane displays the code for the `GameObject` class. The right pane shows the results of a test run named `IterationTest2`. The summary indicates 19 Passed, 0 Failed, and 0 Skipped tests, completed in 207 ms. The Group Summary table shows the total duration for the group is 24 ms. The Outcomes section shows 19 Passed. The Solution Explorer on the right lists the files in the `IterationTest2` project.

Test	Duration	Traits	Error Message
IterationTest2	24 ms		
IterationTest2	24 ms		
IdentifiableObject	24 ms		
TestA...	< 1 ms		
TestB...	< 1 ms		
TestC...	< 1 ms		
TestD...	< 1 ms		
TestE...	< 1 ms		
TestF...	< 1 ms		
TestG...	< 1 ms		
TestH...	< 1 ms		
TestI...	< 1 ms		
TestJ...	< 1 ms		
TestK...	< 1 ms		
TestL...	< 1 ms		
TestM...	< 1 ms		
TestN...	< 1 ms		
TestO...	< 1 ms		
TestP...	< 1 ms		
TestQ...	< 1 ms		
TestR...	< 1 ms		
TestS...	< 1 ms		
TestT...	< 1 ms		
Inventory	< 1 ms		
TestF...	< 1 ms		
TestG...	< 1 ms		
TestH...	< 1 ms		
TestI...	< 1 ms		
TestJ...	< 1 ms		
TestK...	< 1 ms		
TestL...	< 1 ms		
TestM...	< 1 ms		
TestN...	< 1 ms		
TestO...	< 1 ms		
TestP...	< 1 ms		
TestQ...	< 1 ms		
TestR...	< 1 ms		
TestS...	< 1 ms		
TestT...	< 1 ms		
Item	< 1 ms		
FullDe...	< 1 ms		
ItemL...	< 1 ms		
Short...	< 1 ms		
Player	< 1 ms		
FullDe...	< 1 ms		
Identifi...	< 1 ms		
Locati...	< 1 ms		
Locati...	< 1 ms		
Locati...	< 1 ms		

Group Summary
IterationTest2
Tests in group: 19
Total Duration: 24 ms

Outcomes
19 Passed

Search (Ctrl-I) 0 Warnings 0 Errors