

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## Case Study - Iteration 2 - Players Items and Inventory

---

PDF generated at 15:26 on Thursday 24<sup>th</sup> 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",
↪ "inventory" }, name, desc)
14         {
15             _inventory = new Inventory();
16         }
17
18         public GameObject Locate(string id)
19         {
20             if (AreYou(id))
21             {
22                 return this;
23             }
24             return _inventory.Fetch(id);
25         }
26
27         public override string FullDescription
28         {
29             get
30             {
31                 return $"You are {this.Name} {base.FullDescription}\nYou are
↪ carrying:\n{_inventory.ItemList}";
32             }
33         }
34
35         public Inventory Inventory
36         {
37             get
38             {
39                 return _inventory;
40             }
41         }
42     }
43 }
```

```
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
↪ might be fine...");
22             _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
↪ (shovel)\n", _player.FullDescription);
55         }
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,
↵ desc)
12         {
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
↪ might be fine...");
23         }
24
25         [Test()]
26
27         public void ItemIdentifiable()
28         {
29             Assert.IsTrue(_item.AreYou("shovel"));
30             Assert.IsTrue(_item.AreYou("spade"));
31         }
32
33         [Test()]
34
35         public void ShortDesc()
36         {
37             Assert.AreEqual("a shovel (shovel)", _item.ShortDescription);
38         }
39
40         [Test()]
41
42         public void FullDesc()
43         {
44             Assert.AreEqual("This might be fine...", _item.FullDescription);
45         }
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 itm in _items)
18             {
19                 if (itm.AreYou(id))
20                     return true;
21             }
22             return false;
23         }
24
25         public void Put(Item itm)
26         {
27             _items.Add(itm);
28         }
29
30         public Item Take(string id)
31         {
32
33             foreach (Item itm in _items)
34             {
35                 if (itm.AreYou(id))
36                 {
37                     _items.Remove(itm);
38                     return itm;
39                 }
40             }
41             return null;
42         }
43
44         public Item Fetch(string id)
45         {
46             foreach (Item itm in _items)
47             {
48                 if (itm.AreYou(id))
49                 {
50                     return itm;
51                 }
52             }
53             return null;
```

```
54     }
55
56     public string ItemList
57     {
58         get
59         {
60             string itemList = "";
61
62             foreach (Item itm in _items)
63             {
64                 itemList += "\t" + itm.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
↪ might be fine...");
23             _inventory.Put(_item);
24         }
25
26         [Test()]
27
28         public void TestPutItem()
29         {
30             Assert.IsTrue(_inventory.HasItem("shovel"));
31         }
32
33         [Test()]
34
35         public void TestNoItem()
36         {
37             Assert.IsFalse(_inventory.HasItem("gun"));
38         }
39
40         [Test()]
41
42         public void TestFetchItem()
43         {
44             Assert.AreEqual(_item, _inventory.Fetch("shovel"));
45             Assert.IsTrue(_inventory.HasItem("shovel"));
46         }
47
48         [Test()]
49
50         public void TestTakeItem()
51         {
52             Assert.AreEqual(_item, _inventory.Take("shovel"));
```

```
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 }
```

