
4.2P - Case Study - Iteration 2 - Players Items and Inventory

PDF generated at 00:04 on Monday 3rd April, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7
8  namespace Iteration1
9  {
10     public abstract class GameObject : IdentifiableObject
11     {
12         string _description;
13         string _name;
14
15         public GameObject(string[] ids, string name, string desc) : base(ids)
16         {
17             _description = desc;
18             _name = name;
19         }
20
21         public string Name
22         {
23             get
24             {
25                 return _name;
26             }
27         }
28
29         public string ShortDescription
30         {
31             get
32             {
33                 return $"a {Name} ({FirstId})";
34             }
35         }
36
37         public virtual string FullDescription
38         {
39             get
40             {
41                 return _description;
42             }
43         }
44     }
45 }
46
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration1
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 {Name} {base.FullDescription} You are carrying:
↪ {_inventory.ItemList}";
32             }
33         }
34
35         public Inventory Inventory
36         {
37             get
38             {
39                 return _inventory;
40             }
41         }
42     }
43 }
```

```
1  using Iteration1;
2
3  namespace PlayerUnitTest
4  {
5      public class PlayerTests
6      {
7          private Player _TestPlayer;
8          private Item _TestSword;
9          private Item _TestBall;
10
11         [SetUp]
12         public void Setup()
13         {
14             _TestPlayer = new Player("bob", "the cool guy.");
15             _TestSword = new Item(new string[] { "sword" }, "bronze sword", "This is
↵ a bronze sword");
16             _TestBall = new Item(new string[] { "ball" }, "tennis ball", "This is a
↵ tennis ball");
17             _TestPlayer.Inventory.Put(_TestSword);
18             _TestPlayer.Inventory.Put(_TestBall);
19         }
20
21         [Test]
22         public void TestPlayerIsIdentifiable()
23         {
24             Assert.IsTrue(_TestPlayer.AreYou("me") &&
↵ _TestPlayer.AreYou("inventory"));
25         }
26         [Test]
27         public void TestPlayerLocatesItems()
28         {
29             Assert.AreEqual(_TestPlayer.Locate("ball"), _TestBall);
30         }
31
32         [Test]
33         public void TestPlayerLocateItself()
34         {
35             Assert.AreEqual(_TestPlayer.Locate("me"), _TestPlayer);
36         }
37
38         [Test]
39         public void PlayerLocatesNothing()
40         {
41             Assert.AreEqual(_TestPlayer.Locate("bike"), null);
42         }
43         [Test]
44         public void PlayerFullDescription()
45         {
46             Assert.AreEqual(_TestPlayer.FullDescription, $"You are bob the cool guy.
↵ You are carrying: {_TestPlayer.Inventory.ItemList}");
47         }
48     }
49 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7
8  namespace Iteration1
9  {
10     public class Item : GameObject
11     {
12         public Item(string[] idents, string name, string desc) : base(idents, name,
↵      desc)
13         {
14
15         }
16     }
17 }
```

```
1  using Iteration1;
2  namespace ItemUnitTest
3  {
4      public class ItemTests
5      {
6          private Item _TestItem;
7
8          [SetUp]
9          public void Setup()
10         {
11             _TestItem = new Item(new string[]{"sword"}, "bronze sword", "This is a
↪  bronze sword");
12         }
13
14         [Test]
15         public void TestItemIsIdentifiable()
16         {
17             Assert.IsTrue(_TestItem.AreYou("sword"));
18         }
19         [Test]
20         public void TestShortDescription()
21         {
22             Assert.AreEqual(_TestItem.ShortDescription, "a bronze sword (sword)");
23         }
24         [Test]
25         public void TestFullDescription()
26         {
27             Assert.AreEqual(_TestItem.FullDescription, "This is a bronze sword");
28         }
29     }
30 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Xml.Linq;
7
8  namespace Iteration1
9  {
10     public class Inventory
11     {
12         private List<Item> _items;
13
14         public Inventory()
15         {
16             _items = new List<Item>();
17         }
18
19         public bool HasItem(string id)
20         {
21             foreach (Item item in _items)
22             {
23                 if (item.AreYou(id))
24                 {
25                     return true;
26                 }
27             }
28             return false;
29         }
30
31         public void Put(Item itm)
32         {
33             _items.Add(itm);
34         }
35
36         public Item Take(string id)
37         {
38             foreach (Item itm in _items)
39             {
40                 if (itm.AreYou(id))
41                 {
42                     _items.Remove(itm);
43                     return itm;
44                 }
45             }
46             return null;
47         }
48
49         public Item Fetch(string id)
50         {
51             foreach (Item item in _items)
52             {
53                 if (item.AreYou(id))
```

```
54         {
55             return item;
56         }
57     }
58     return null;
59 }
60
61 public string ItemList
62 {
63     get
64     {
65         string itemlist = "";
66
67         foreach (Item item in _items)
68         {
69             itemlist += $"{item.ShortDescription}\n";
70         }
71         return itemlist ;
72     }
73 }
74 }
75 }
```



```
1  using Iteration1;
2  namespace InventoryUnitTest
3  {
4      public class InventoryTest
5      {
6          private Item _TestSword;
7          private Inventory _TestInventory;
8          private Item _TestBall;
9
10
11         [SetUp]
12         public void Setup()
13         {
14             _TestSword = new Item(new string[] { "sword" }, "bronze sword", "This is
↪ a bronze sword");
15             _TestBall = new Item(new string[] { "ball" }, "tennis ball", "This is a
↪ tennis ball");
16             _TestInventory = new Inventory();
17             _TestInventory.Put(_TestSword);
18             _TestInventory.Put(_TestBall);
19         }
20
21         [Test]
22         public void TestFindItem()
23         {
24             Assert.NotNull(_TestInventory);
25             Assert.IsTrue(_TestInventory.HasItem("sword"));
26             Assert.IsTrue(_TestInventory.HasItem("ball"));
27         }
28
29         [Test]
30         public void NoItemFind()
31         {
32             Assert.IsFalse(_TestInventory.HasItem("chicken"));
33         }
34
35         [Test]
36         public void FetchItem()
37         {
38             Assert.AreEqual(_TestInventory.Fetch("sword"), _TestSword);
39             Assert.AreEqual(_TestInventory.Fetch("ball"), _TestBall);
40         }
41
42         [Test]
43         public void TakeItem()
44         {
45             Assert.AreEqual(_TestInventory.Take("sword"), _TestSword);
46         }
47     }
48 }
```

