

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## Case Study - Iteration 3 - Bags

---

PDF generated at 20:33 on Sunday 1<sup>st</sup> October, 2023

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Xml.Linq;
6
7  namespace SwinAdventure;
8
9  public class Bag : Item
10 {
11     private Inventory _inventory;
12
13     public Bag(string[] ids, string name, string desc) : base(ids, 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         else
25         {
26             return _inventory.Fetch(id);
27         }
28     }
29
30     public override string FullDescription
31     {
32         get
33         {
34             return $"In the {Name}, you can see:\n{_inventory.ItemList}";
35         }
36     }
37
38     public Inventory Inventory
39     {
40         get { return _inventory; }
41     }
42 }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Linq;
5  using NUnit.Framework;
6  using SwinAdventure;
7
8  namespace SwinAdventureTests
9  {
10     [TestFixture]
11     public class BagTests
12     {
13         Bag _bag2;
14         Bag _bag;
15         Item _sword = new Item(new string[] { "sword" }, "sword", "This is a sword");
16         Item _spade = new Item(new string[] { "spade" }, "spade", "This is a spade");
17         Item _shovel = new Item(new string[] { "shovel" }, "shovel", "This is a
18         ↵ shovel");
19         Item _diamond = new Item(new string[] { "diamond" }, "diamond", "This is a
20         ↵ diamond");
21
22         [SetUp]
23         public void Setup()
24         {
25             _bag = new Bag(new string[] { "bag" }, "Lorraine's bag", "This is a
26             ↵ bag");
27             _bag2 = new Bag(new string[] { "bag2" }, "a bag2", "This is a bag2");
28             _bag.Inventory.Put(_sword); _bag.Inventory.Put(_spade);
29             _bag2.Inventory.Put(_shovel); _bag2.Inventory.Put(_diamond);
30         }
31
32         [Test]
33         public void TestBagLocatesItems()
34         {
35             Assert.IsTrue(_bag.Inventory.HasItem("sword")); //Check if have item
36             Assert.IsTrue(_bag.Inventory.HasItem("spade"));
37
38             Assert.IsTrue(_bag.Locate(_sword.FirstID) == _sword); //Check if locate
39             ↵ item
40             Assert.IsTrue(_bag.Locate(_spade.FirstID) == _spade);
41         }
42
43         [Test]
44         public void TestBagLocatesItself()
45         {
46             Assert.IsTrue(_bag.Locate(_bag.FirstID) == _bag);
47             Assert.IsTrue(_bag.Locate("bag") == _bag);
48         }
49
50         [Test]
51         public void TestBagLocatesNothing()
52         {
53             Assert.That(_bag.Locate("nothing"), Is.EqualTo(null));
54         }
55     }
56 }
```

```
50         }
51
52     [Test]
53     public void TestBagFullDescription()
54     {
55         string expectedFullDescription = "In the Lorraine's bag, you can
56         ← see:\n\ta sword (sword)\n\ta spade (spade)";
57         Assert.That(_bag.FullDescription, Is.EqualTo(expectedFullDescription));
58     }
59
60     [Test]
61     public void TestBagInBag()
62     {
63         _bag.Inventory.Put(_bag2);
64         Assert.IsTrue(_bag.Locate(_bag2.FirstID) == _bag2);
65         Assert.IsTrue(_bag.Locate(_sword.FirstID) == _sword);
66         Assert.IsFalse(_bag.Locate(_diamond.FirstID) == _diamond);
67     }
68 }
```

