

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 3 - Bags

PDF generated at 14:14 on Monday 25th September, 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
10     // Define a class called Bag that inherits from Item class
11
12     public class Bag : Item
13     {
14
15         // Private field to store the inventory of the bag or bag inventory
16
17         private Inventory _inventory;
18
19         // Constructor for the "Bag" class
20
21         public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
22         {
23
24             // Initialise the inventory when a new bag is created
25
26             _inventory = new Inventory();
27         }
28
29
30         // Method to locate a game object based on its id
31
32         public GameObject Locate(string id)
33         {
34
35             // check if the bag itself has the specified id
36
37             if (AreYou(id))
38             {
39
40                 // return the bag if the id matches the bag's id
41
42                 return this;
43             }
44             else if (_inventory.HasItem(id))
45             {
46
47                 // return the item if the id matches the item's id
48
49                 return _inventory.Fetch(id);
50             }
51
52             // null if object is not found in the bag or the itself
53
54         }
```

```
54         else return null;
55     }
56
57
58     // override the fulldiscription propert from the base class
59
60     public override string FullDescription
61     {
62         get
63         {
64
65             // provide the description of the bag's contents by concatenating the item in its
66             ↪ inventory
67
68             return $"In the bag you can see:\n" + _inventory.ItemList;
69         }
70
71         // Public property to access the inventory of the bag
72
73         public Inventory Inventory
74         {
75             get
76             {
77                 return _inventory;
78             }
79         }
80
81     }
82 }
```

```
1  using Iteration2;
2  using NUnit.Framework;
3  using System;
4
5  namespace Bag_Test
6  {
7      [TestFixture]
8      public class TestBag
9      {
10         Bag bag;
11         Bag cat;
12         Item stool;
13         Item talwar;
14
15         [SetUp]
16         public void Setup()
17         {
18             bag = new Bag(new string[] { "bag" }, "bag", "This is a expensive bag");
19             cat = new Bag(new string[] { "cat" }, "cat", "This is a cat");
20
21             stool = new Item(new string[] { "stool" }, "50cm stool", "This is a
↪ stool");
22             talwar = new Item(new string[] { "talwar" }, "a strong talwar", "This is
↪ a kings talwar ");
23
24
25             bag.Inventory.Put(cat);
26             bag.Inventory.Put(stool);
27             cat.Inventory.Put(talwar);
28         }
29
30         [Test]
31         public void TestLocateItems()
32         {
33
34             Assert.That(bag.Locate("stool"), Is.SameAs(stool));
35             Assert.That(bag.Inventory.HasItem("stool"), Is.True);
36         }
37
38         [Test]
39         public void TestLocateItself()
40         {
41             Assert.That(bag.Locate("bag"), Is.SameAs(bag));
42         }
43
44         [Test]
45         public void TestLocateNothing()
46         {
47             Assert.That(bag.Locate("// put something \\ "), Is.SameAs(null));
48         }
49
50         [Test]
51         public void TestFullDescription()
```

```
52     {
53         Assert.That(bag.FullDescription,
54             Is.EqualTo("In the bag you can see:\n" + bag.Inventory.ItemList));
55     }
56
57     [Test]
58     public void TestBagInBag()
59     {
60         Assert.That(bag.Locate("cat"), Is.SameAs(cat));
61         Assert.That(bag.Locate("stool"), Is.SameAs(stool));
62         Assert.That(bag.Locate("talwar"), Is.SameAs(null));
63     }
64 }
65 }
```

