

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 3 - Bags

PDF generated at 22:34 on Tuesday 19th 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 SwinAdventure
8  {
9      public class Bag : Item
10     {
11         private Inventory _inventory;
12         public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
13         {
14             _inventory = new Inventory();
15         }
16         public GameObject Locate(string id)
17         {
18             if (AreYou(id))
19             {
20                 return this;
21             }
22             else if (_inventory.HasItem(id))
23             {
24                 return _inventory.Fetch(id);
25             }
26             return null;
27         }
28         public override string FullDescription
29         {
30             get
31             {
32                 return $"{this.Name}, containing:\n" + _inventory.ItemList;
33             }
34         }
35         public Inventory Inventory { get { return _inventory; } }
36     }
37 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using SwinAdventure;
7
8  namespace SwinAdventureTests
9  {
10     public class BagTest
11     {
12         private Bag _bagObject;
13         private Bag _secondBagObject;
14         private Item _shovel;
15         private Item _sword;
16         private Item _bow;
17         [SetUp]
18         public void Setup()
19         {
20             _bagObject = new Bag(new[] { "bag" }, "a bag", "This is a bag");
21             _secondBagObject = new Bag(new[] { "second bag" }, "a second bag", "This
↪ is a second bag");
22             _shovel = new Item(new[] { "shovel" }, "a shovel", "This is a shovel");
23             _sword = new Item(new[] { "sword" }, "a sword", "This is a Sword");
24             _bow = new Item(new[] { "bow" }, "a bow", "This is a bow");
25             _secondBagObject.Inventory.Put(_bow);
26             _bagObject.Inventory.Put(_sword);
27             _bagObject.Inventory.Put(_shovel);
28         }
29         [Test]
30         public void TestBagLocatesItems()
31         {
32
33             Assert.IsTrue(_bagObject.Inventory.HasItem("sword")); //Check if have
↪ item
34             Assert.IsTrue(_bagObject.Inventory.HasItem("shovel"));
35
36             Assert.IsTrue(_bagObject.Locate(_sword.FirstID) == _sword); //Check if
↪ locate item
37             Assert.IsTrue(_bagObject.Locate(_shovel.FirstID) == _shovel);
38
39         }
40
41         [Test]
42         public void TestBagLocatesItself()
43         {
44             Assert.IsTrue(_bagObject.Locate(_bagObject.FirstID) == _bagObject);
45             Assert.IsTrue(_bagObject.Locate("bag") == _bagObject);
46         }
47         [Test]
48         public void TestBagLocatesNothing()
49         {
50             Assert.IsTrue(_bagObject.Locate(_bow.FirstID) == null);
```

```
51     }
52     [Test]
53     public void TestBagFullDescription()
54     {
55         Assert.AreEqual(_bagObject.FullDescription, "a bag, containing:\na sword
↪ (sword)\na shovel (shovel)\n");
56     }
57     [Test]
58     public void TestBagInBag()
59     {
60         _bagObject.Inventory.Put(_secondBagObject);
61         Assert.IsTrue(_bagObject.Locate(_secondBagObject.FirstID) ==
↪ _secondBagObject);
62         Assert.IsTrue(_bagObject.Locate(_sword.FirstID) == _sword);
63         Assert.IsFalse(_bagObject.Locate(_bow.FirstID) == _bow);
64
65         Assert.AreEqual(_bagObject.FullDescription, "a bag, containing:\na sword
↪ (sword)\na shovel (shovel)\na second bag (second bag)\n");
66         Assert.AreEqual(_secondBagObject.FullDescription, "a second bag,
↪ containing:\na bow (bow)\n");
67     }
68 }
69 }
70 }
```

