

```
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        Inventory _inventory = new Inventory();
12
13        public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
14        {
15        }
16
17        public GameObject Locate(string id)
18        {
19            if (AreYou(id))
20            {
21                return this;
22            }
23            return _inventory.Fetch(id);
24        }
25
26        public string FullDescription
27        {
28            get
29            {
30                return "In the " + Name + " you can see:\n" +
31                    _inventory.ItemList;
32            }
33        }
34
35        public Inventory Inventory
36        {
37            get
38            {
39                return _inventory;
40            }
41        }
42    }
43 }
```



```
1 using SwinAdventure;
2
3 namespace BagTest
4 {
5     internal class BagTest
6     {
7         Bag bag = new Bag(new string[] { "bag" }, "bag", "a bag");
8         Item BronzeSword = new Item(new string[] { "sword", "weapon" },
9             "Bronze Sword", "A simple bronze sword.");
10        Item BronzeAxe = new Item(new string[] { "axe", "weapon" }, "Bronze
11            Axe", "A simple bronze axe.");
12    [SetUp]
13    public void Setup()
14    {
15        bag = new Bag(new string[] { "bag" }, "bag", "a bag");
16    }
17
18    [Test]
19    public void TestBagLocate()
20    {
21        bag.Inventory.Put(BronzeSword);
22
23        GameObject obj = bag.Locate("sword");
24
25        Assert.AreEqual(BronzeSword, obj);
26    }
27
28    [Test]
29    public void TestBagLocateItself()
30    {
31        GameObject obj = bag.Locate("bag");
32
33        Assert.AreEqual(bag, obj);
34    }
35
36    [Test]
37    public void TestBagLocateNothing()
38    {
39        GameObject obj = bag.Locate("sword");
40
41        Assert.IsNull(obj);
42    }
43
44    [Test]
45    public void TestBagFullDescription()
46    {
47        bag.Inventory.Put(BronzeSword);
48        bag.Inventory.Put(BronzeAxe);
```













```
48         string desc = bag.FullDescription;
49
50         Assert.AreEqual("In the bag you can see:\nBronze Sword (sword) \nBronze Axe (axe)\n", desc);
51     }
52
53     [Test]
54     public void BagInBag()
55     {
56         Bag b1 = new Bag(new string[] { "bag" }, "bag1", "a bag");
57         Bag b2 = new Bag(new string[] { "bag" }, "bag2", "a bag");
58
59         b1.Inventory.Put(b2);
60         b1.Inventory.Put(BronzeSword);
61         b2.Inventory.Put(BronzeAxe);
62
63         GameObject obj1 = b1.Locate("bag");
64         GameObject obj2 = b1.Locate("sword");
65         Assert.IsNotNull(obj1);
66         Assert.IsNotNull(obj2);
67
68         GameObject obj3 = b1.Locate("axe");
69         Assert.IsNull(obj3);
70     }
71 }
72 }
73
```

Test Explorer



Search (Ctrl+I)


Test run finished: 24 Tests (24 Passed, 0 Failed, 0  0 Warnings  0 Errors

Test	Duration	T...	Error Message
▲  Test (24)	10 ms		
▲  BagTest (5)	10 ms		
▲  BagTest (5)	10 ms		
 BagInBag	7 ms		
 TestBagFullDescription	3 ms		
 TestBagLocate	< 1 ms		
 TestBagLocateItself	< 1 ms		
 TestBagLocateNothing	< 1 ms		
▸  IdentifiableObjectTest (6)	< 1 ms		
▸  InventoryTest (5)	< 1 ms		
▸  ItemTest (3)	< 1 ms		
▸  PlayerTest (5)	< 1 ms		


Group Summary

Test

Tests in group: 24

 Total Duration: 10 ms

Outcomes

 24 Passed