

# Bag.cs

C:\Users\PC\Desktop\COS20007\Week\_5\5.2P\Iteration1\Bag.cs

1

```
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 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 if (_inventory.HasItem(id))
25            {
26                return _inventory.Fetch(id);
27            }
28            return null;
29        }
30
31        public override string FullDescription
32        {
33            get
34            {
35                string description = null;
36                description += $"In the {this.Name} you can see:\n";
37                description += _inventory.ItemList;
38                return description;
39            }
40        }
41
42        public Inventory Inventory
43        {
44            get => _inventory;
45        }
46    }
47 }
48
```

# BagUnitTests.cs

...esktop\COS20007\Week\_5\5.2P\BagUnitTests\UnitTest1.cs

1

```
1 using Iteration1;
2 using System.Numerics;
3
4 namespace BagUnitTests
5 {
6     public class Tests
7     {
8         Bag bag1;
9         Bag bag2;
10        Item sword;
11        Item ak47;
12        Item machine_gun;
13
14        [SetUp]
15        public void Setup()
16        {
17            bag1 = new Bag(new string[] { "bag1", "1" }, "Bag 1", "This is the 1st bag of the player!");
18            sword = new Item(new string[] { "sword", "melee" }, "bronze sword", "Melee weapon. High damage.");
19            ak47 = new Item(new string[] { "ak47", "gun" }, "ak47", "Gun. High Damage.");
20            bag1.Inventory.Put(sword);
21            bag1.Inventory.Put(ak47);
22        }
23
24        [Test]
25        public void TestBagLocatesItems()
26        {
27            Assert.That(bag1.Locate("sword"), Is.EqualTo(sword));
28            Assert.That(bag1.Inventory.HasItem("sword"), Is.EqualTo(true));
29            Assert.Pass();
30        }
31
32        [Test]
33        public void TestBagLocatesItself()
34        {
35            Assert.That(bag1.Locate("bag1"), Is.EqualTo(bag1));
36            Assert.Pass();
37        }
38
39        [Test]
40        public void TestBagLocatesNothing()
41        {
42            Assert.That(bag1.Locate("machine gun"), Is.EqualTo(null));
43            Assert.Pass();
44        }
45
46        [Test]
```

```
47     public void TestBagFullDescription()
48     {
49         Assert.That(bag1.FullDescription, Is.EqualTo($"In the
           {bag1.Name} you can see:\n    a bronze sword (sword)\n    a
           ak47 (ak47)\n"));
50         Assert.Pass();
51     }
52
53     [Test]
54     public void TestBagInBag()
55     {
56         bag2 = new Bag(new string[] { "bag2", "2" }, "Bag 2", "This is
           the 2nd part of the player!");
57         machine_gun = new Item(new string[] { "machine gun", "super
           gun" }, "machine gun", "this is a weapon having lots of
           bullets");
58         bag2.Inventory.Put(machine_gun);
59
60         bag1.Inventory.Put(bag2);
61         Assert.That(bag1.Locate("bag2"), Is.EqualTo(bag2));
62         Assert.That(bag1.Locate("ak47"), Is.EqualTo(ak47));
63         Assert.That(bag1.Locate("machine gun"), Is.EqualTo(null));
64         Assert.Pass();
65     }
66 }
67 }
68 }
```

# Output Testing

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the `BagUnitTests` class with a `TestBagFullDescription()` method. The code includes using statements for `Iteration1` and `System.Numerics`, and a namespace `BagUnitTests` containing a `Tests` class with several static methods.
- Test Explorer:** Displays the test run results. The test run finished with 24 tests passed, 0 failed, and 0 skipped, running in 188 ms. The test results are summarized in the following table:

| Test                       | Duration | Traits | Error Message |
|----------------------------|----------|--------|---------------|
| TestIdentifiableObject (6) | 51 ms    |        |               |
| PlayerUnitTests (5)        | 51 ms    |        |               |
| ItemUnitTests (3)          | 50 ms    |        |               |
| InventoryUnitTests (5)     | 50 ms    |        |               |
| BagUnitTests (5)           | 17 ms    |        |               |
| BagUnitTests (5)           | 17 ms    |        |               |
| Tests (5)                  | 17 ms    |        |               |
| TestBagLocatesNothing      | < 1 ms   |        |               |
| TestBagLocatesItself       | < 1 ms   |        |               |
| TestBagLocatesItems        | 2 ms     |        |               |
| TestBagInBag               | < 1 ms   |        |               |
| TestBagFullDescription     | 15 ms    |        |               |

The **Group Summary** for `BagUnitTests` shows 5 tests in the group, a total duration of 17 ms, and 5 passed outcomes.

The **Test Explorer** also shows the test results for the `TestBagFullDescription()` method, which passed.

The **Solution Explorer** on the right shows the project structure, including `BagUnitTests`, `Iteration1`, and `GameObjects`.

The **Output Window** at the bottom shows the test results for the `TestBagFullDescription()` method, including the assertion message: `Assert.That(bag1.FullDescription, Is.EqualTo($"In the {bag1.Name} you can see:\n a bronze sword (sword)\n a ak47 (ak47)\n"));`