## 5.2. Case Iteration 3: Bag

## Bag Class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class Bag : Item
    {
        private Inventory _inventory = new Inventory();

        public Bag(string[] idents, string name, string desc) : base(idents,
name, desc)
        {
        }

        public Item? Locate(string id)
        {
            if (AreYou(id))
            {
                return this;
            }
           else
                return _inventory.Fetch(id);
        }

        public override string FullDescription
        {
            get
            {
                return "In the " + Name + " you can see:\n" +
_inventory.ItemList;
            }
        }

        public Inventory Inventory
        {
            get
            {
                return _inventory;
            }
        }
    }
}
```

## Test file

```
using SwinAdventure;

namespace TestQueue
{
    public class Tests
    {
        Item item1 = new Item(new string[] { "sword" }, "sword", "a sword");
        Item item2 = new Item(new string[] { "shield" }, "shield", "a shield");
```

```
        Item item3 = new Item(new string[] { "shiba" }, "shiba", "a shiba");

        [SetUp]
        public void Setup()
        {
        }



         //Test the Bag class
        [Test]
        public void BagLocate()
        {
            Bag backpack = new Bag(new string[] { "backpack" }, "backpack", "a
backpack");
            backpack.Inventory.Put(item1);
            backpack.Inventory.Put(item2);
            backpack.Inventory.Put(item3);

            //ask to return item and item stays in backpack
            Assert.That(item3, Is.EqualTo(backpack.Locate("shiba")));
            Assert.IsTrue(backpack.Inventory.HasItem("shiba"));

        }

        [Test]
        public void BagLocatesItself()
        {
            Bag backpack = new Bag(new string[] { "backpack" }, "backpack", "a
backpack");
            Assert.That(backpack, Is.EqualTo(backpack.Locate("backpack")));
        }

        [Test]
        public void BagLocateNothing()
        {
            Bag backpack = new Bag(new string[] { "backpack" }, "backpack", "a
backpack");
            Assert.That(backpack.Locate("sword"), Is.Null);
        }

        [Test]
        public void BagFullDescription()
        {
            Bag backpack = new Bag(new string[] { "backpack" }, "backpack", "a
backpack");
            backpack.Inventory.Put(item1);
            backpack.Inventory.Put(item2);
            backpack.Inventory.Put(item3);

            //the list string below is the expected output, consisting of every
item in the following format: name ( first id)
            Assert.That(backpack.FullDescription, Is.EqualTo("In the backpack you
can see:\nsword (sword)\nshield (shield)\nshiba (shiba)\n"));
        }

        [Test]
        public void BagInBag()
        {
            Bag backpack = new Bag(new string[] { "backpack" }, "backpack", "a
backpack");
            Bag satchel = new Bag(new string[] { "satchel" }, "satchel", "a
satchel");
```

```
            backpack.Inventory.Put(satchel);

            Assert.That(satchel, Is.EqualTo(backpack.Locate("satchel")));
        }
    }
}
```