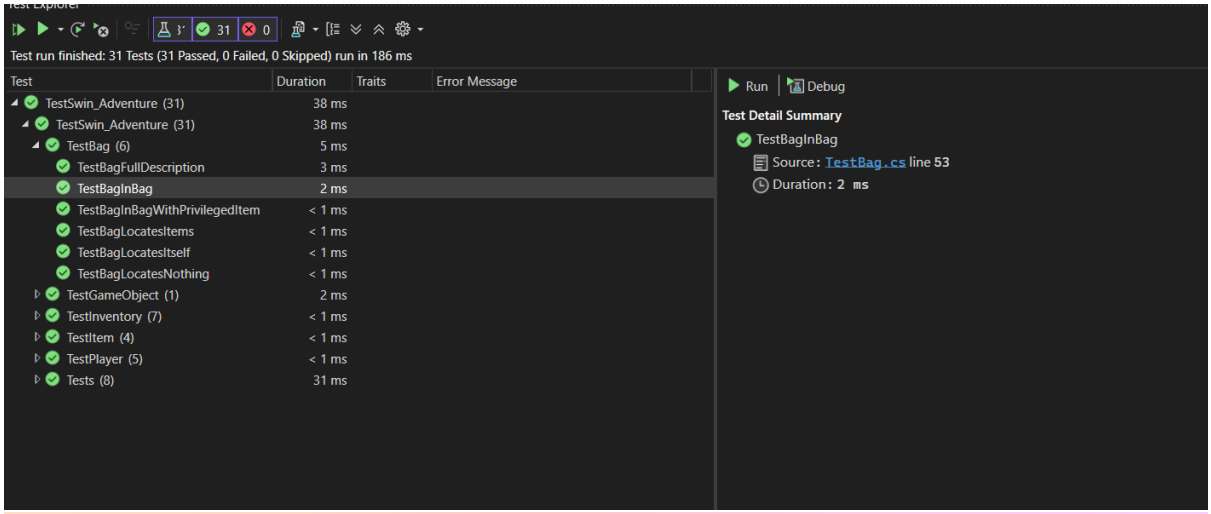


```cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Swin_Adventure
{
    public class Bag : Item
    {
        private Inventory _inventory;

        public Bag(string[] ids, string name, string desc) :
            base(ids, name, desc)
        {
            _inventory = new Inventory();
        }

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

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

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

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NUnit.Framework;
using Swin_Adventure;

namespace TestSwin_Adventure
{
    public class TestBag
    {
        [Test]
        public void TestBagLocatesItems()
        {
            Bag bag = new Bag(new string[] { "bag" }, "leather bag", "A sturdy leather bag");
            Item item = new Item(new string[] { "sword" }, "bronze sword", "A bronze sword");
            bag.Inventory.Put(item);
            Assert.AreEqual(item, bag.Locate("sword"));
            Assert.True(bag.Inventory.HasItem("sword"));
        }

        [Test]
        public void TestBagLocatesItself()
        {
            Bag bag = new Bag(new string[] { "bag", "sack" }, "leather bag", "A sturdy leather bag");
            Assert.AreEqual(bag, bag.Locate("bag"));
            Assert.AreEqual(bag, bag.Locate("sack"));
        }

        [Test]
        public void TestBagLocatesNothing()
        {
            Bag bag = new Bag(new string[] { "bag" }, "leather bag", "A sturdy leather bag");
            Assert.IsNull(bag.Locate("nonexistent"));
        }

        [Test]
        public void TestBagFullDescription()
        {
            Bag bag = new Bag(new string[] { "bag" }, "leather bag", "A sturdy leather bag");
            Item item1 = new Item(new string[] { "sword" }, "bronze sword", "A bronze sword");
            Item item2 = new Item(new string[] { "gem" }, "shiny gem", "A shiny gem");
            bag.Inventory.Put(item1);
            bag.Inventory.Put(item2);
            string desc = bag.FullDescription;
            Assert.IsTrue(desc.Contains("In the leather bag you can see:"));
            Assert.IsTrue(desc.Contains("bronze sword (sword)"));
            Assert.IsTrue(desc.Contains("shiny gem (gem)"));
        }

        [Test]
        public void TestBagInBag()
        {
            Bag b1 = new Bag(new string[] { "b1" }, "bag one", "First bag");
            Bag b2 = new Bag(new string[] { "b2" }, "bag two", "Second bag");
            Item item = new Item(new string[] { "sword" }, "bronze sword", "A bronze sword");
            b2.Inventory.Put(item);
            b1.Inventory.Put(b2);
            Assert.AreEqual(b2, b1.Locate("b2"));
            Assert.IsNull(b1.Locate("sword"));
        }

        [Test]
        public void TestBagInBagWithPrivilegedItem()
        {
            Bag b1 = new Bag(new string[] { "b1" }, "bag one", "First bag");
            Bag b2 = new Bag(new string[] { "b2" }, "bag two", "Second bag");
            Item privileged = new Item(new string[] { "privileged" }, "privileged item", "A privileged item");
            b2.Inventory.Put(privileged);
            b1.Inventory.Put(b2);
            privileged.PrivilegeEscalation("1358"); // Escalate privilege
            Assert.IsNull(b1.Locate("privileged"));
        }
    }
}
```