Swinburne University of Technology

COS20007 Object Oriented Programming

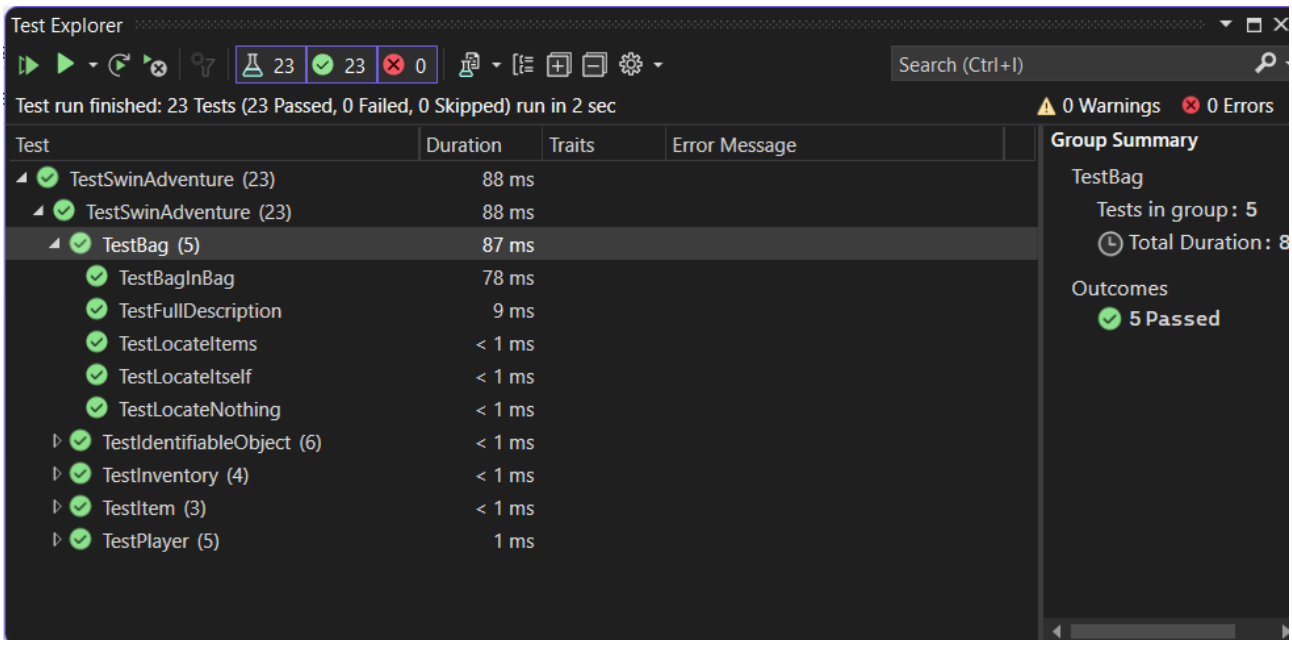
5.2P - Case Study - Iteration 3 - Bags

PDF generated at 07:36 on Saturday 25th March, 2023

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

namespace SwinAdventure
{
    public class Bag : Item
    {
        //local variables
        private Inventory _inventory;

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

        //methods
        public GameObject Locate(string id)
        {
            if (AreYou(id))
            {
                return this;
            }
            else if (_inventory.HasItem(id))
            {
                return _inventory.Fetch(id);
            }
            else return null;
        }

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

        public Inventory Inventory => _inventory;
    }
}
```

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

namespace TestSwinAdventure
{
    [TestFixture]
    public class TestBag
    {
        //initializing 2 bags and 2 items
        Bag bag;
        Bag purse;
        Item sword;
        Item bat;

        [SetUp]
        public void Setup()
        {
            bag = new Bag(new string[] { "bag" }, "bag", "This is a good bag");
            purse = new Bag(new string[] { "purse" }, "purse", "This is a purse");

            sword = new Item(new string[] { "Sword" }, "a bronze sword", "This is a
↪ bronze sword");
            bat = new Item(new string[] { "Bat" }, "a hard bat", "This is a hard
↪ bat");

            // adding a bag inside another bag, and 1 items in each
            bag.Inventory.Put(purse);
            bag.Inventory.Put(sword);
            purse.Inventory.Put(bat);
        }

        [Test]
        public void TestLocateItems()
        {
            //test if it can locate an item inside it and it remains there
            Assert.That(bag.Locate("sword"), Is.SameAs(sword));
            Assert.That(bag.Inventory.HasItem("sword"), Is.True);
        }

        [Test]
        public void TestLocateItself()
        {
            Assert.That(bag.Locate("bag"), Is.SameAs(bag));
        }

        [Test]
        public void TestLocateNothing()
        {
```

```
52              Assert.That(bag.Locate("apron"), Is.SameAs(null));
53          }
54
55          [Test]
56          public void TestFullDescription()
57          {
58              Assert.That(bag.FullDescription,
59                  Is.EqualTo("In the bag you can see:\n" + bag.Inventory.ItemList));
60          }
61
62          [Test]
63          public void TestBagInBag()
64          {
65              //test is the bag can locate another bag inside it
66              Assert.That(bag.Locate("purse"), Is.SameAs(purse));
67              Assert.That(bag.Locate("sword"), Is.SameAs(sword));
68
69              //test if it cannot locate an item inside the bag its storing
70              Assert.That(bag.Locate("bat"), Is.SameAs(null));
71          }
72      }
73  }
```


Test Explorer
23
23
0
Search (Ctrl+I)
Test run finished: 23 Tests (23 Passed, 0 Failed, 0 Skipped) run in 2 sec
0 Warnings
0 Errors
Test
Duration
Traits
Error Message
TestSwinAdventure (23) 88 ms
TestSwinAdventure (23) 88 ms
TestBag (5) 87 ms
TestBagInBag 78 ms
TestFullDescription 9 ms
TestLocateItems < 1 ms
TestLocateItself < 1 ms
TestLocateNothing < 1 ms
TestIdentifiableObject (6) < 1 ms
TestInventory (4) < 1 ms
TestItem (3) < 1 ms
TestPlayer (5) 1 ms
Group Summary
TestBag
Tests in group: 5
Total Duration: 8
Outcomes
5 Passed