Swinburne University of Technology

COS20007 Object Oriented Programming

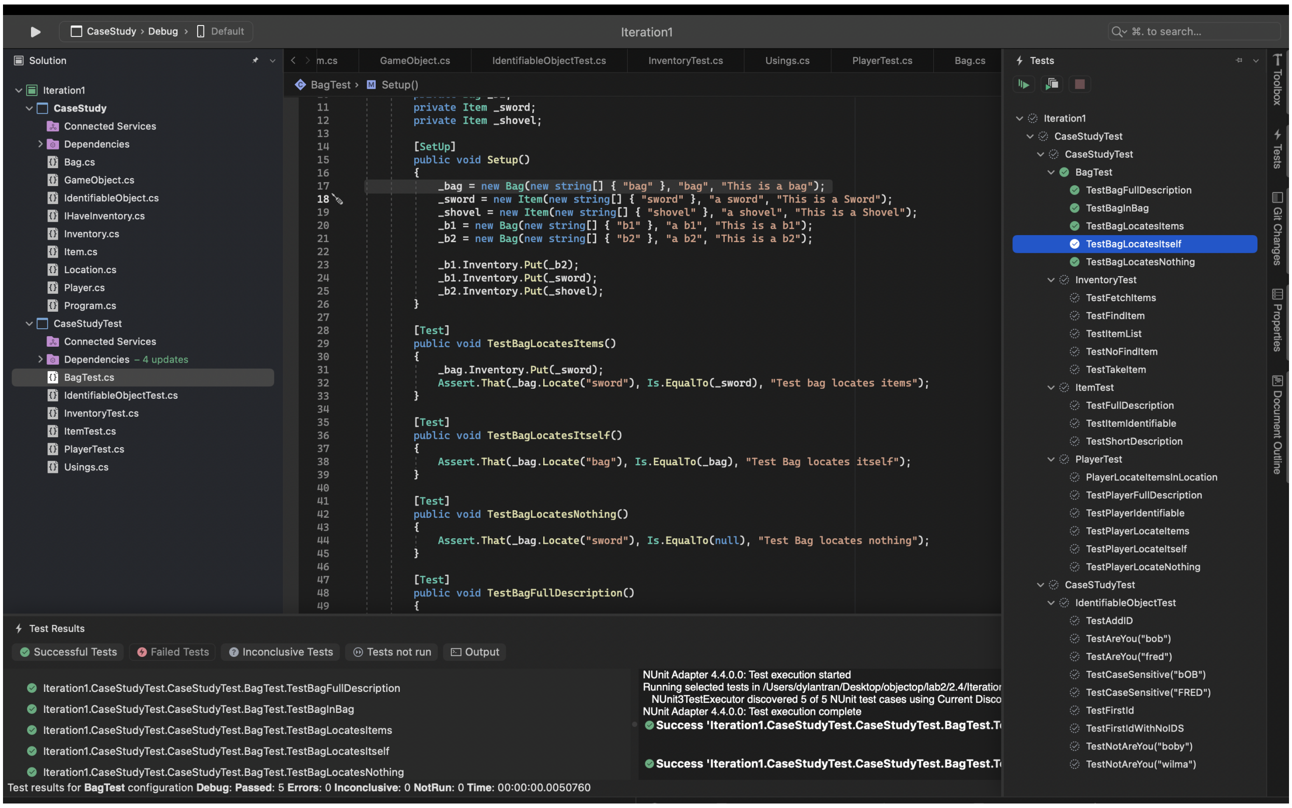
Case Study - Iteration 3 - Bags

PDF generated at 19:43 on Wednesday 4th October, 2023

Produced by OOP Submission Formatter

```
using System;
namespace CaseStudy
{
        public class Bag : Item, IHaveInventory
        {
            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;
                }
                else if (_inventory.HasItem(id))
                {
                    return _inventory.Fetch(id);
                }
                else
                {
                    return null;
                }
            }

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

            public Inventory Inventory
            {
                get
                {
                    return _inventory;
                }
            }

        }
}

```

```
using System;
using CaseStudy;

namespace CaseStudyTest
{
    public class BagTest
    {
        private Bag _bag;
        private Bag _b1;
        private Bag _b2;
        private Item _sword;
        private Item _shovel;

        [SetUp]
        public void Setup()
        {
            _bag = new Bag(new string[] { "bag" }, "bag", "This is a bag");
            _sword = new Item(new string[] { "sword" }, "a sword", "This is a Sword");
            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a Shovel");
            _b1 = new Bag(new string[] { "b1" }, "a b1", "This is a b1");
            _b2 = new Bag(new string[] { "b2" }, "a b2", "This is a b2");

            _b1.Inventory.Put(_b2);
            _b1.Inventory.Put(_sword);
            _b2.Inventory.Put(_shovel);
        }

        [Test]
        public void TestBagLocatesItems()
        {
            _bag.Inventory.Put(_sword);
            Assert.That(_bag.Locate("sword"), Is.EqualTo(_sword), "Test bag locates items");
        }

        [Test]
        public void TestBagLocatesItself()
        {
            Assert.That(_bag.Locate("bag"), Is.EqualTo(_bag), "Test Bag locates itself");
        }

        [Test]
        public void TestBagLocatesNothing()
        {
            Assert.That(_bag.Locate("sword"), Is.EqualTo(null), "Test Bag locates nothing");
        }

        [Test]
        public void TestBagFullDescription()
        {
            Assert.That(_bag.FullDescription, Is.EqualTo("In the bag you can see:\n" + _bag.Inventory.ItemList), "Test Bag Full Description");
        }

        [Test]
        public void TestBagInBag()
        {
            Assert.That(_b1.Locate("b2"), Is.EqualTo(_b2), "Test b1 can locate b2");

            Assert.That(_b1.Locate("sword"), Is.EqualTo(_sword), "Test b1 can locate other items in b1");

            Assert.That(_b1.Locate("shovel"), Is.EqualTo(null), "Test b1 can not locate items in b2");
        }
    }
}


```



```
        private Item _sword;
        private Item _shovel;

        [SetUp]
        public void Setup()
        {
            _bag = new Bag(new string[] { "bag" }, "bag", "This is a bag");
            _sword = new Item(new string[] { "sword" }, "a sword", "This is a Sword");
            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a Shovel");
            _b1 = new Bag(new string[] { "b1" }, "a b1", "This is a b1");
            _b2 = new Bag(new string[] { "b2" }, "a b2", "This is a b2");

            _b1.Inventory.Put(_b2);
            _b1.Inventory.Put(_sword);
            _b2.Inventory.Put(_shovel);
        }

        [Test]
        public void TestBagLocatesItems()
        {
            _bag.Inventory.Put(_sword);
            Assert.That(_bag.Locate("sword"), Is.EqualTo(_sword), "Test bag locates items");
        }

        [Test]
        public void TestBagLocatesItself()
        {
            Assert.That(_bag.Locate("bag"), Is.EqualTo(_bag), "Test Bag locates itself");
        }

        [Test]
        public void TestBagLocatesNothing()
        {
            Assert.That(_bag.Locate("sword"), Is.EqualTo(null), "Test Bag locates nothing");
        }

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

```
Iteration1.CaseStudyTest.CaseStudyTest.BagTest.TestBagFullDescription
Iteration1.CaseStudyTest.CaseStudyTest.BagTest.TestBagInBag
Iteration1.CaseStudyTest.CaseStudyTest.BagTest.TestBagLocatesItems
Iteration1.CaseStudyTest.CaseStudyTest.BagTest.TestBagLocatesItself
Iteration1.CaseStudyTest.CaseStudyTest.BagTest.TestBagLocatesNothing
Test results for BagTest configuration Debug: Passed: 5 Errors: 0 Inconclusive: 0 NotRun: 0 Time: 00:00:00.0050760
```

```
NUnit Adapter 4.4.0.0: Test execution started
Running selected tests in /Users/dylantran/Desktop/objectop/lab2/2.4/Iteratior
  NUnit3TestExecutor discovered 5 of 5 NUnit test cases using Current Disco
NUnit Adapter 4.4.0.0: Test execution complete
Success 'Iteration1.CaseStudyTest.CaseStudyTest.BagTest.Te

Success 'Iteration1.CaseStudyTest.CaseStudyTest.BagTest.Te
```