COS20007 - Object Oriented Programming

5.2P - Case Study -- Interation 3: Bags

Student name: Nguyen Duc Manh
ID: 105547489

Iteration 3's new code

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public class Bags : Item
10     {
11         Inventory _inventory;
12
13         public Bags(string[] idents, string name, string desc) : base
             (idents, 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             } return null;
28         }
29
30         public override string FullDescription
31         {
32             get { return $"In the {Name} you can see:\n{_inventory.ItemList
                 ()}" ; }
33         }
34
35         public Inventory Inventory
36         {
37             get { return _inventory; }
38         }
39     }
40 }
41
```

```
1  namespace SwinAdventure;
2
3  public class BagsTest
4  {
5      Item _item1;
6      Item _item2;
7      Bags _bag1;
8      Bags _bag2;
9
10     [SetUp]
11     public void Setup()
12     {
13         _item1 = new Item(["Ram"], "a Ram", "an NVIDIA Ram");
14         _item2 = new Item(["CPU"], "a CPU", "an Intel CPU");
15         _bag1 = new Bags(["Bag1"], "bag test 1", "This bag is huge");
16         _bag2 = new Bags(["Bag2"], "bag test 2", "This bag is small");
17         _bag1.Inventory.Put(_item1);
18         _bag1.Inventory.Put(_item2);
19         //
20     }
21
22     [Test]
23     public void BagLocatesItemTest()
24     {
25         Assert.That(_bag1.Inventory.HasItem("Ram"));
26         Assert.That(_bag1.Inventory.HasItem("CPU"));
27         Assert.That(_bag1.Locate("Ram"), Is.EqualTo(_item1));
28         Assert.That(_bag1.Locate("CPU"), Is.EqualTo(_item2));
29     }
30
31     [Test]
32     public void BagLocatesItselfTest()
33     {
34         Assert.That(_bag1.Locate("Bag1"), Is.EqualTo(_bag1));
35         Assert.That(_bag2.Locate("Bag2"), Is.EqualTo(_bag2));
36     }
37
38     [Test]
39     public void BagLocatesNothingTest()
40     {
41         Assert.That(_bag1.Locate("abc"), Is.Null);
42         Assert.That(_bag2.Locate("xyz"), Is.Null);
43     }
44
45     [Test]
46     public void BagFullDescriptionTest()
47     {
48         Assert.That(_bag1.FullDescription, Is.EqualTo("In the bag test 1
           you can see:\na Ram (ram)\na CPU (cpu)\n"));
```

```
49      }
50
51      [Test]
52      public void BagInBagTest()
53      {
54          Item _item3 = new Item(["Mouse"], "a Mouse", "a wireless mouse");
55          _bag1.Inventory.Put(_bag2);
56          _bag2.Inventory.Put(_item3);
57
58          Assert.That(_bag1.Locate("Bag2"), Is.EqualTo(_bag2)); //Can locate
              bag2 in bag1's inventoyr
59          Assert.That(_bag1.Locate("Ram"), Is.EqualTo(_item1)); //bag1 still
              can locates other items
60          Assert.That(_bag1.Locate("Mouse"), Is.Null); //bag1 can't search
              for bag2's item
61      }
62  }
63
```

Earlier code

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public abstract class GameObject : IdenObj
10     {
11         string _description;
12         string _name;
13
14         public GameObject(string[] idents, string name, string desc) : base
             (idents)
15         {
16             _name = name;
17             _description = desc;
18         }
19
20         public string Name { get { return _name; } }
21
22         public string ShortDescription { get { return $"{_name}
             ({FirstId})"; } }
23
24         public virtual string FullDescription { get { return
             _description; } }
25     }
26 }
27
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public class IdenObj
10     {
11         //fields
12         private List<string> _identifiers;
13         string _myStudentID = "7489";
14
15         //constructor
16         public IdenObj(string[] idents)
17         {
18             _identifiers = new List<string>();
19             if (idents != null)
20             {
21                 for (int i = 0; i < idents.Length; i++)
22                 {
23                     _identifiers.Add(idents[i].ToLower());
24                 }
25             }
26         }
27
28         //methods
29         public bool AreYou(string id)
30         {
31             return _identifiers.Contains(id.ToLower());
32         }
33
34         public string FirstId
35         {
36             get
37             {
38                 if( _identifiers.Count == 0)
39                 {
40                     return "";
41                 } else { return _identifiers.First(); }
42             }
43         }
44
45         public void AddIdentifier(string id)
46         {
47             _identifiers.Add(id.ToLower());
48         }
49
```

```
50          public void PrivilegeEscalation(string pin)
51          {
52              if(pin.Length == 4)
53              {
54                  if(pin == _myStudentID) //105547489
55                  {
56                      _identifiers[0] = _myStudentID;
57                  }
58              }
59              else
60              {
61                  return;
62              }
63
64          }
65      }
66  }
67
```

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

namespace SwinAdventure
{
    public class Inventory
    {
        List<Item> _items;

        public Inventory()
        {
            _items = new List<Item>();
        }

        public bool HasItem(string id)
        {
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    return true;
                }
            }
            return false;
        }

        public void Put(Item itm)
        {
            _items.Add(itm);
        }

        public Item Take(string id)
        {
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    _items.Remove(item);
                    return item;
                }
            }
            return null;
        }

        public Item Fetch(string id)
        {
```

```
50              foreach (Item item in _items)
51              {
52                  if (item.AreYou(id))
53                  {
54                      return item;
55                  }
56              }
57              return null;
58          }
59
60          public string ItemList()
61          {
62              string listitm = "";
63              foreach (Item item in _items)
64              {
65                  listitm = listitm + item.ShortDescription + "\n";
66              }
67              return listitm;
68          }
69      }
70  }
71
```

```
1  namespace SwinAdventure
2  {
3      public class InventoryTest
4      {
5          Item _item;
6          Inventory _inventory;
7
8          [SetUp]
9          public void Setup()
10         {
11             _item = new(["HDMI"], "HDMI cord", "can connect to large
                 screen");
12             _inventory = new Inventory();
13         }
14
15         [Test]
16         public void FindItemTest()
17         {
18             _inventory.Put(_item);
19             Assert.That(_inventory.HasItem(_item.FirstId), Is.True);
20         }
21
22         [Test]
23         public void NoItemFindTest()
24         {
25             Assert.That(_inventory.HasItem("Mouse"), Is.False);
26         }
27
28         [Test]
29         public void FetchItemTest()
30         {
31             _inventory.Put(_item);
32             Assert.That(_inventory.Fetch(_item.FirstId), Is.EqualTo
                 (_item));
33         }
34
35         [Test]
36         public void TakeItemTest()
37         {
38             _inventory.Put(_item);
39             _inventory.Take(_item.FirstId);
40             Assert.That(_inventory.HasItem(_item.FirstId), Is.False);
41         }
42
43         [Test]
44         public void TestItemList()
45         {
46             _inventory.Put(_item);
47             Assert.That(_inventory.ItemList, Is.EqualTo("HDMI cord (hdmi)
```

```
                \n"));
48          }
49      }
50  }
51
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public class Item : GameObject
10     {
11         public Item(string[] idents, string name, string desc) : base
             (idents, name, desc)
12         {
13             //not yet
14         }
15     }
16 }
17
```

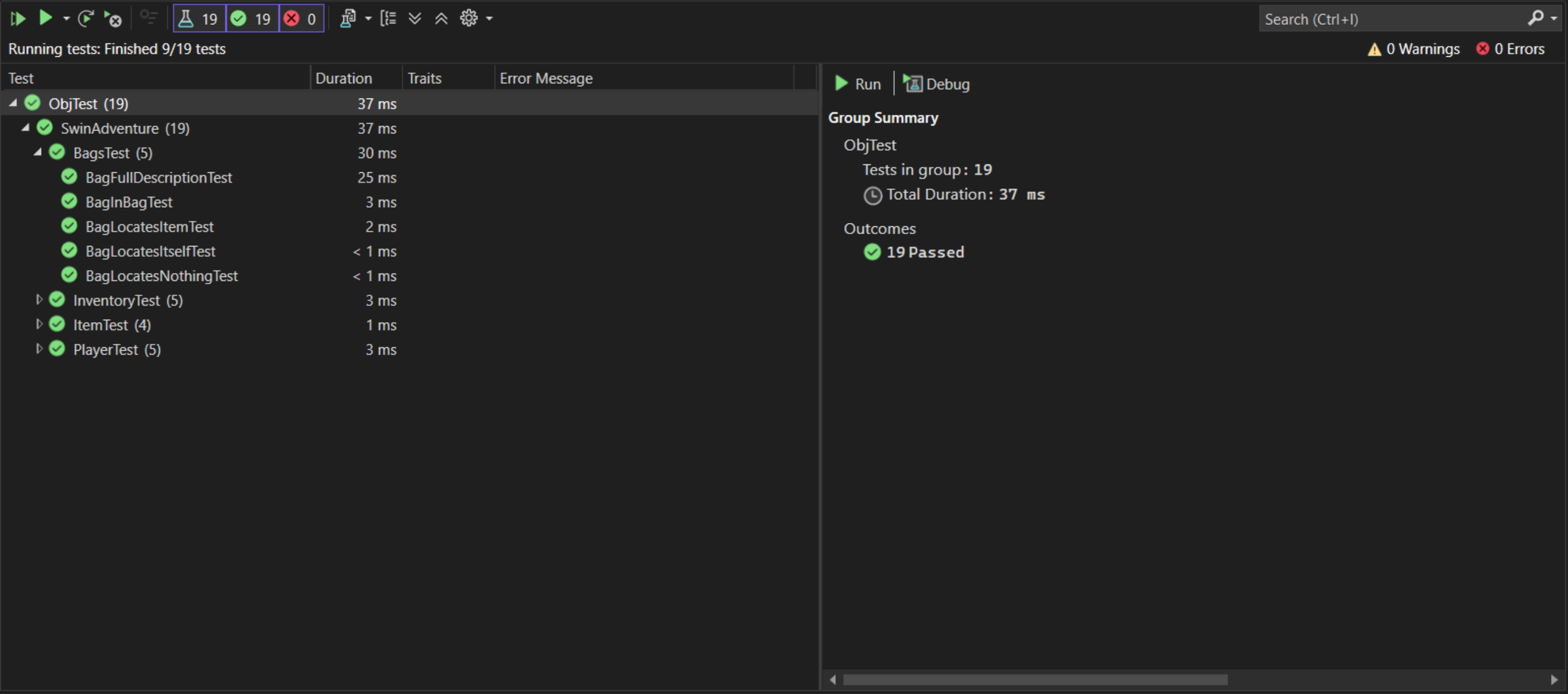
```
1  using Microsoft.VisualStudio.TestPlatform.ObjectModel;
2
3  namespace SwinAdventure
4  {
5      public class ItemTest
6      {
7          Item laptop;
8
9          [SetUp]
10         public void Setup()
11         {
12             laptop = new Item(new string[] { "laptop" }, "a laptop", "This
               is a Swinburne laptop");
13         }
14
15         [Test]
16         public void TestItemIdentifiable()
17         {
18             var areyou2 = laptop.AreYou("laptop");
19             Assert.IsTrue(areyou2);
20         }
21
22         [Test]
23         public void TestShortDescription()
24         {
25             Assert.That(laptop.ShortDescription, Is.EqualTo("a laptop
               (laptop)"));
26         }
27
28         [Test]
29         public void TestFullDescription()
30         {
31             Assert.That(laptop.FullDescription, Is.EqualTo("This is a
               Swinburne laptop"));
32         }
33
34         [Test]
35         public void PrivilegeEscalationTest()
36         {
37             var firstID = new string[] { "sword", "blade" };
38             var item = new Item(firstID, "Sword", "A sharp blade");
39             item.PrivilegeEscalation("7489");
40
41             Assert.That(item.FirstId, Is.EqualTo("7489"));
42         }
43
44     }
45 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public class Player : GameObject
10     {
11         Inventory _inventory;
12
13         public Player (string name, string desc) : base(new string[] {"
             me", "inventory"}, 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             return _inventory.Fetch(id);
25         }
26
27         public override string FullDescription
28         {
29             get
30             {
31                 return $"{Name}, {base.ShortDescription}.You are carrying:
                     {_inventory.ItemList()}";
32             }
33         }
34
35         public Inventory Inventory { get { return _inventory; } }
36     }
37 }
38
```

```
1  namespace SwinAdventure
2  {
3      public class PlayerTest
4      {
5          Item _item;
6          Inventory _inventory;
7          Player _swinburneStudent;
8
9          [SetUp]
10         public void Setup()
11         {
12             _item = new(new String[] { "sword" }, "diamond sword", "can
                   destroy enemies");
13             _inventory = new Inventory();
14             _swinburneStudent = new Player("Duc Manh", "OOP Student");
15         }
16
17         [Test]
18         public void PlayerIsIdentifiableTest()
19         {
20             Assert.Multiple(() =>
21             {
22                 Assert.That(_swinburneStudent.AreYou("me"), Is.True);
23                 Assert.That(_item.AreYou("sword"), Is.True);
24             });
25         }
26
27         [Test]
28         public void PlayerLocatesItemsTest()
29         {
30             _swinburneStudent.Inventory.Put(_item);
31             Assert.That(_swinburneStudent.Locate(_item.FirstId), Is.EqualTo
                   (_item));
32         }
33
34         [Test]
35         public void PlayerLocatesItselfTesr()
36         {
37             Assert.That(_swinburneStudent, Is.EqualTo
                   (_swinburneStudent.Locate("me")));
38             Assert.That(_swinburneStudent, Is.EqualTo
                   (_swinburneStudent.Locate("inventory")));
39         }
40
41         [Test]
42         public void PlayerLocatesNothingTest()
43         {
44             Assert.That(_swinburneStudent.Locate("shield"), Is.EqualTo
                   (null));
```

```
45          }
46
47          [Test]
48          public void PlayerFullDescriptionTest()
49          {
50              string expectedOutput = "Duc Manh, Duc Manh (me).You are
                  carrying: diamond sword (sword)\n";
51              _swinburneStudent.Inventory.Put(_item);
52              Assert.That(expectedOutput, Is.EqualTo
                  (_swinburneStudent.FullDescription));
53          }
54      }
55  }
56
```

```
1  namespace SwinAdventure
2  {
3      public class Program
4      {
5          public static void Main(string[] args)
6          {
7              Console.WriteLine("Hehehe");
8          }
9      }
10 }
```


19
19
0
Search (Ctrl+I)
Running tests: Finished 9/19 tests
0 Warnings
0 Errors
Test
Duration
Traits
Error Message
ObjTest (19) 37 ms
SwinAdventure (19) 37 ms
BagsTest (5) 30 ms
BagFullDescriptionTest 25 ms
BagInBagTest 3 ms
BagLocatesItemTest 2 ms
BagLocatesItselfTest < 1 ms
BagLocatesNothingTest < 1 ms
InventoryTest (5) 3 ms
ItemTest (4) 1 ms
PlayerTest (5) 3 ms
Run
Debug
Group Summary
ObjTest
Tests in group: 19
Total Duration: 37 ms
Outcomes
19 Passed