COS20007 - Object Oriented Programming

4.2P - Case Study - Iteration 2: Custom Players, Items, and Inventory

Student name: Nguyen Duc Manh
ID: 105547489

SWINBURNE UNIVERSITY OF TECHNOLOGY

Part 1: GameObject, Player, Item, Inventory

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

namespace SwinAdventure
{
    public abstract class GameObject : IdenObj
    {
        string _description;
        string _name;

        public GameObject(string[] idents, string name, string desc) : base
          (idents)
        {
            _name = name;
            _description = desc;
        }

        public string Name { get { return _name; } }

        public string ShortDescription { get { return $"{_name}
          ({FirstId})"; } }

        public virtual string FullDescription { get { return _description; } }
    }
}
```

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

namespace SwinAdventure
{
    public class Player : GameObject
    {
        Inventory _inventory;

        public Player (string name, string desc) : base(new string[] {"me",
          "inventory"}, 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 $"{Name}, {base.ShortDescription}.You are carrying:
                  {_inventory.ItemList()}";
            }
        }

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

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

namespace SwinAdventure
{
    public class Item : GameObject
    {
        public Item(string[] idents, string name, string desc) : base(idents,
          name, desc)
        {
            //not yet
        }
    }
}
```

```
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)
        {
```

```
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    return item;
                }
            }
            return null;
        }

        public string ItemList()
        {
            string listitm = "";
            foreach (Item item in _items)
            {
                listitm = listitm + item.ShortDescription + "\n";
            }
            return listitm;
        }
    }
}
```

Part 2: Test Code

```
namespace SwinAdventure
{
    public class InventoryTest
    {
        Item _item;
        Inventory _inventory;

        [SetUp]
        public void Setup()
        {
            _item = new(new String[] { "HDMI" }, "HDMI cord", "can connect to large screen");
            _inventory = new Inventory();
        }

        [Test]
        public void FindItemTest()
        {
            _inventory.Put(_item);
            Assert.That(_inventory.HasItem(_item.FirstId), Is.True);
        }

        [Test]
        public void NoItemFindTest()
        {
            Assert.That(_inventory.HasItem("Mouse"), Is.False);
        }

        [Test]
        public void FetchItemTest()
        {
            _inventory.Put(_item);
            Assert.That(_inventory.Fetch(_item.FirstId), Is.EqualTo(_item));
        }

        [Test]
        public void TakeItemTest()
        {
            _inventory.Put(_item);
            _inventory.Take(_item.FirstId);
            Assert.That(_inventory.HasItem(_item.FirstId), Is.False);
        }

        [Test]
        public void TestItemList()
        {
            _inventory.Put(_item);
            Assert.That(_inventory.ItemList, Is.EqualTo("HDMI cord (hdmi)\n"));
        }
```

```
    }
}
```

```
using Microsoft.VisualStudio.TestPlatform.ObjectModel;

namespace SwinAdventure
{
    public class ItemTest
    {
        Item laptop;

        [SetUp]
        public void Setup()
        {
            laptop = new Item(new string[] { "laptop" }, "a laptop", "This is a Swinburne laptop");
        }

        [Test]
        public void TestItemIdentifiable()
        {
            var areyou2 = laptop.AreYou("laptop");
            Assert.IsTrue(areyou2);
        }

        [Test]
        public void TestShortDescription()
        {
            Assert.That(laptop.ShortDescription, Is.EqualTo("a laptop (laptop)"));
        }

        [Test]
        public void TestFullDescription()
        {
            Assert.That(laptop.FullDescription, Is.EqualTo("This is a Swinburne laptop"));
        }

        [Test]
        public void PrivilegeEscalationTest()
        {
            var firstID = new string[] { "sword", "blade" };
            var item = new Item(firstID, "Sword", "A sharp blade");
            item.PrivilegeEscalation("7489");

            Assert.That(item.FirstId, Is.EqualTo("7489"));
        }

    }
}
```

```
namespace SwinAdventure
{
    public class PlayerTest
    {
        Item _item;
        Inventory _inventory;
        Player _swinburneStudent;

        [SetUp]
        public void Setup()
        {
            _item = new(new String[] { "sword" }, "diamond sword", "can destroy enemies");
            _inventory = new Inventory();
            _swinburneStudent = new Player("Duc Manh", "OOP Student");
        }

        [Test]
        public void PlayerIsIdentifiableTest()
        {
            Assert.Multiple(() =>
            {
                Assert.That(_swinburneStudent.AreYou("me"), Is.True);
                Assert.That(_item.AreYou("sword"), Is.True);
            });
        }

        [Test]
        public void PlayerLocatesItemsTest()
        {
            _swinburneStudent.Inventory.Put(_item);
            Assert.That(_swinburneStudent.Locate(_item.FirstId), Is.EqualTo(_item));
        }

        [Test]
        public void PlayerLocatesItselfTesr()
        {
            Assert.That(_swinburneStudent, Is.EqualTo(_swinburneStudent.Locate("me")));
            Assert.That(_swinburneStudent, Is.EqualTo(_swinburneStudent.Locate("inventory")));
        }

        [Test]
        public void PlayerLocatesNothingTest()
        {
            Assert.That(_swinburneStudent.Locate("shield"), Is.EqualTo(null));
        }
```

```
        [Test]
        public void PlayerFullDescriptionTest()
        {
            string expectedOutput = "Duc Manh, Duc Manh (me).You are carrying: ↵
              diamond sword (sword)\n";
            _swinburneStudent.Inventory.Put(_item);
            Assert.That(expectedOutput, Is.EqualTo                            ↵
              (_swinburneStudent.FullDescription));
        }
    }
}
```

Part 3: Earlier code and Test Output

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

namespace SwinAdventure
{
    public class IdenObj
    {
        //fields
        private List<string> _identifiers;
        string _myStudentID = "7489";

        //constructor
        public IdenObj(string[] idents)
        {
            _identifiers = new List<string>();
            if (idents != null)
            {
                for (int i = 0; i < idents.Length; i++)
                {
                    _identifiers.Add(idents[i].ToLower());
                }
            }
        }

        //methods
        public bool AreYou(string id)
        {
            return _identifiers.Contains(id.ToLower());
        }

        public string FirstId
        {
            get
            {
                if( _identifiers.Count == 0)
                {
                    return "";
                } else { return _identifiers.First(); }
            }
        }

        public void AddIdentifier(string id)
        {
            _identifiers.Add(id.ToLower());
        }
```

```
        public void PrivilegeEscalation(string pin)
        {
            if(pin.Length == 4)
            {
                if(pin == _myStudentID) //105547489
                {
                    _identifiers[0] = _myStudentID;
                }
            }
            else
            {
                return;
            }

        }
    }
}
```

```
namespace SwinAdventure
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("I'm making Indentifiable Object");
        }
    }
}
```

Test Explorer

14 14 0

Search (Ctrl+I)

Test run finished: 14 Tests (14 Passed, 0 Failed, 0 Skipped) run in 460 ms

0 Warnings 0 Errors

Test	Duration	Traits	Error Message
ObjTest (14)	21 ms		
SwinAdventure (14)	21 ms		
InventoryTest (5)	19 ms		
FetchItemTest	11 ms		
FindItemTest	2 ms		
NoItemFindTest	< 1 ms		
TakeItemTest	< 1 ms		
TestItemList	6 ms		
ItemTest (4)	1 ms		
PrivilegeEscalationTest	1 ms		
TestFullDescription	< 1 ms		
TestItemIdentifiable	< 1 ms		
TestShortDescription	< 1 ms		
PlayerTest (5)	1 ms		
PlayerFullDescriptionTest	< 1 ms		
PlayerIsIdentifiableTest	1 ms		
PlayerLocatesItemsTest	< 1 ms		
PlayerLocatesItselfTesr	< 1 ms		
PlayerLocatesNothingTest	< 1 ms		

Run Debug

Group Summary

ObjTest

Tests in group: 14

Total Duration: 21 ms

Outcomes

14 Passed