

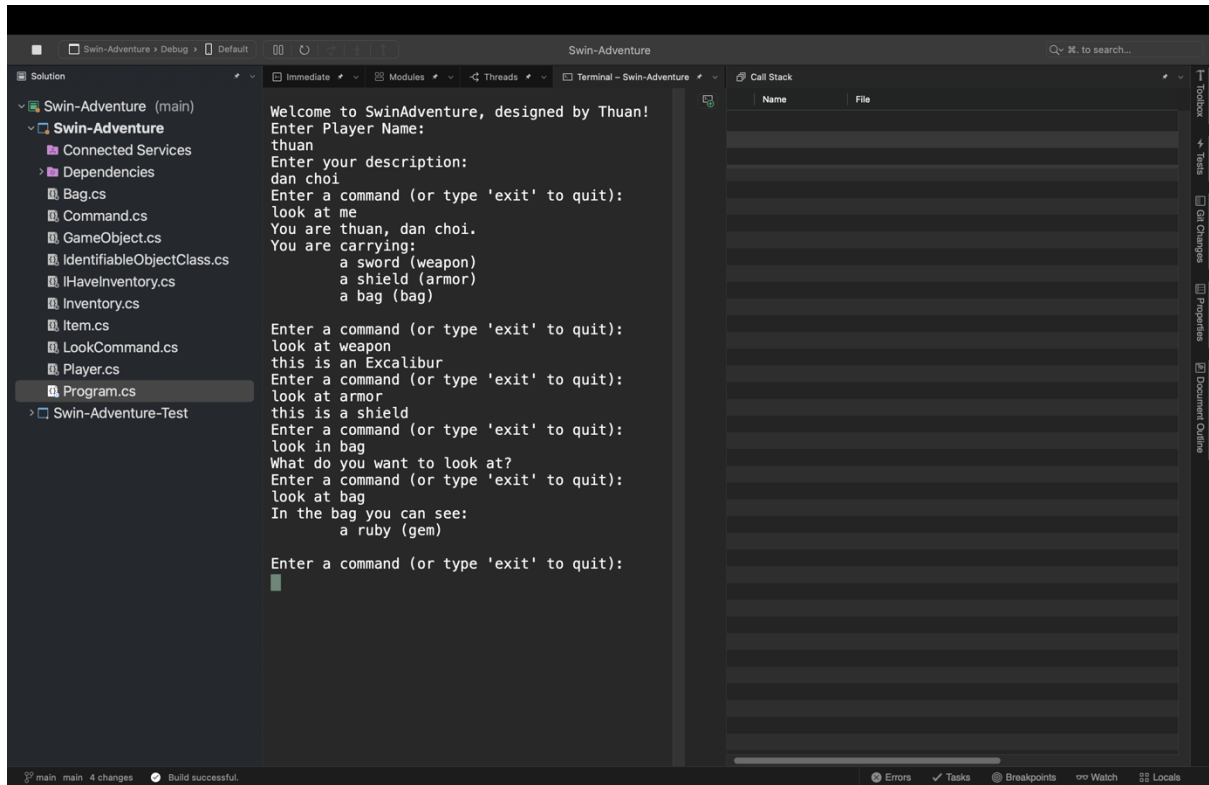
COS 20007

Task 7.1

Duc Thuan Tran

104330455

- ▼ Swin-Adventure
 - ▼ Swin-Adventure-Test
 - ▼ IdentifiableObjectTest
 - ▼ Tests
 - TestAddID
 - TestAreYou
 - TestCaseSensitive
 - TestFirstID
 - TestFirstIdWithNoIDs
 - TestNotAreYou
 - ▼ SwinAdventureTest
 - ▼ InventoryTest
 - TestFetchItem
 - TestFindItem
 - TestItemList
 - TestNoItemFind
 - TestTakeItem
 - ▼ ItemTest
 - TestFullDescription
 - TestItemIsIdentifiable
 - TestShortDescription
 - ▼ PlayerTest
 - TestPlayerFullDescription
 - TestPlayerIsIdentifiable
 - TestPlayerLocateItems
 - TestPlayerLocateItself
 - TestPlayerLocateNothing
 - ▼ TestLookCommand
 - TestInvalidLook
 - TestLookAtGem
 - TestLookAtGemInBag
 - TestLookAtGemInMe
 - TestLookAtGemInNoBag
 - TestLookAtMe
 - TestLookAtNoGemInBag
 - TestLookAtUnk
 - ▼ Swin_Adventure
 - ▼ BagTest
 - TestBagFullDescription
 - TestBagInBag
 - TestBagLocatesItems
 - TestBagLocatesitself
 - TestBagLocatesnothing



Folder Swin-Adventure

11 printable files

(file list disabled)

Swin-Adventure/Bag.cs

```
1 using System;
2 namespace Swin_Adventure
3 {
4     public class Bag : Item, IHaveInventory
5     {
6         private Inventory _inventory;
7
8         public Bag(string[] ids, string name, string description) : base(ids, name,
description)
9         {
10             _inventory = new Inventory();
11         }
12
13         public GameObject Locate(string id)
14         {
15             if (this.AreYou(id))
16             {
17                 return this;
18             }
19             return _inventory.Fetch(id);
20         }
21
22         public override string FullDescription
23         {
24             get { return $"In the {Name} you can see:\n" + _inventory.ItemList; }
25         }
26
27         public Inventory Inventory
28         {
29             get { return _inventory; }
30         }
31     }
32 }
33
34
```

Swin-Adventure/Command.cs

```
1 using System;
2 namespace Swin_Adventure
3 {
4     public abstract class Command : IdentifiableObject
5     {
6         public Command(string[] ids) : base(ids)
7         {
8         }
9         public abstract string Execute(Player player, string[] text);
10    }
```

```
11 | }  
12 |  
13 |
```

Swin-Adventure/GameObject.cs

```
1  using System;  
2  using System.Xml.Linq;  
3  
4  namespace Swin_Adventure  
5  {  
6      public class GameObject : IdentifiableObject  
7      {  
8          private string _description;  
9          private string _name;  
10  
11         public GameObject(string[] ids, string name, string description) :  
base(ids)  
12         {  
13             _description = description;  
14             _name = name;  
15         }  
16  
17         public string Name  
18         {  
19             get { return _name.ToLower(); }  
20         }  
21  
22         public string ShortDescription  
23         {  
24             get { return $"a {_name.ToLower()} ({FirstId.ToLower()})"; }  
25         }  
26  
27         public virtual string FullDescription  
28         {  
29             get { return _description; }  
30         }  
31     }  
32 }  
33  
34
```

Swin-Adventure/IHaveInventory.cs

```
1  using System;  
2  namespace Swin_Adventure  
3  {  
4      public interface IHaveInventory  
5      {  
6          GameObject Locate(string id);  
7          string Name { get; }  
8      }  
9  }  
10  
11
```

Swin-Adventure/IdentifiableObjectClass.cs

```
1 using System;
2 namespace Swin_Adventure
3 {
4     public class IdentifiableObject
5     {
6         private List<string> _identifiers;
7
8         public IdentifiableObject(string[] ids)
9         {
10             _identifiers = new List<string>(ids);
11             _identifiers.AddRange(ids);
12         }
13
14         public bool AreYou(string id)
15         {
16             return _identifiers.Contains(id.ToLower());
17         }
18
19         public string FirstId
20         {
21             get
22             {
23                 if (_identifiers.Count == 0)
24                 {
25                     return "";
26                 }
27                 return _identifiers[0];
28             }
29         }
30
31         public void AddIdentifier(string id)
32         {
33             _identifiers.Add(id.ToLower());
34         }
35     }
36 }
37
38
```

Swin-Adventure/Inventory.cs

```
1 using System;
2 namespace Swin_Adventure
3 {
4     public class Inventory
5     {
6         private List<Item> _items;
7         public Inventory()
8         {
9             _items = new List<Item>();
10        }
11
12        public bool HasItem(string id)
13        {
14            foreach (Item itm in _items)
15            {
16                if (itm.AreYou(id))
17            }
18        }
19    }
20 }
21
22
```

```
17         {
18             return true;
19         }
20     }
21     return false;
22 }
23
24 public void Put(Item itm)
25 {
26     _items.Add(itm);
27 }
28
29 public Item Take(string id)
30 {
31     Item itm = Fetch(id);
32
33     if (itm != null)
34     {
35         _items.Remove(itm);
36     }
37
38     return itm;
39 }
40
41 public Item Fetch(string id)
42 {
43     foreach (Item itm in _items)
44     {
45         if (itm.AreYou(id))
46         {
47             return itm;
48         }
49     }
50     return null;
51 }
52
53 public string ItemList
54 {
55     get
56     {
57         string list = "";
58         foreach (Item item in _items)
59         {
60             list += "\t" + "a " + item.Name + " (" + item.FirstId + ")\n";
61         }
62         return list;
63     }
64 }
65 }
66 }
67
68
```

Swin-Adventure/Item.cs

```
1 using System;
2
3 namespace Swin_Adventure
4 {
```

```
5     public class Item : GameObject
6     {
7         public Item(string[] idents, string name, string description) :
base(idents, name, description)
8         {
9
10        }
11    }
12 }
13
14
```

Swin-Adventure/LookCommand.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using Swin_Adventure;
5
6 namespace Swin_Adventure
7 {
8     public class LookCommand : Command
9     {
10        public LookCommand() : base(new string[] { "look" }) { }
11
12        public override string Execute(Player player, string[] text)
13        {
14            IHaveInventory container = null;
15            string itemId;
16
17            if (text.Length != 3 && text.Length != 5)
18            {
19                return "I don't know how to look like that";
20            }
21            else
22            {
23                if (text[0] != "look")
24                {
25                    return "Error in look input";
26                }
27                if (text[1] != "at")
28                {
29                    return "What do you want to look at?";
30                }
31                if (text.Length == 5 && text[3] != "in")
32                {
33                    return "What do you want to look in?";
34                }
35
36                switch (text.Length)
37                {
38                    case 3:
39                        container = player;
40                        break;
41
42                    case 5:
43                        container = FetchContainer(player, text[4]);
44
45                        if (container == null)
```



```

46         {
47             return $"I can't find the {text[4]}";
48         }
49         break;
50     }
51     itemId = text[2];
52     return LookAtIn(itemId, container);
53 }
54 }
55
56 private IHaveInventory FetchContainer(Player player, string containerId)
57 {
58     return player.Locate(containerId) as IHaveInventory;
59 }
60
61 private string LookAtIn(string thingId, IHaveInventory container)
62 {
63     GameObject locatedObject = container.Locate(thingId);
64
65     if (locatedObject != null)
66     {
67         return locatedObject.FullDescription;
68     }
69     else
70     {
71         return $"I can't find the {thingId}";
72     }
73 }
74 }
75 }
76

```

Swin-Adventure/Player.cs

```

1  using System;
2  namespace Swin_Adventure
3  {
4      public class Player : GameObject, IHaveInventory
5      {
6          private Inventory _inventory;
7
8          public Player(string name, string description) : base(new string[] { "me",
9  "inventory" }, name, description)
10         {
11             _inventory = new Inventory();
12         }
13
14         public GameObject Locate(string id)
15         {
16             if(AreYou(id))
17             {
18                 return this;
19             }
20             return _inventory.Fetch(id);
21         }
22
23         public override string FullDescription
24         {
25             get
26

```

```

25         {
26             return "You are " + Name + ", " + base.FullDescription + ".\n"
27                 + "You are carrying:\n" + Inventory.ItemList;
28         }
29     }
30
31     public Inventory Inventory
32     {
33         get{ return _inventory; }
34     }
35 }
36 }
37
38

```

Swin-Adventure/Program.cs

```

1  using System;
2  using Swin_Adventure;
3
4  namespace Swin_Adventure;
5
6  class Program
7  {
8      static void Main(string[] args)
9      {
10         Console.WriteLine("Welcome to SwinAdventure, designed by Thuan!");
11
12         Console.WriteLine("Enter Player Name: ");
13         string playerName = Console.ReadLine();
14
15         Console.WriteLine("Enter your description: ");
16         string playerDescription = Console.ReadLine();
17
18         Player player = new Player(playerName, playerDescription);
19
20         Item item1 = new Item(new string[] { "weapon" }, "sword", "this is an
Excalibur");
21         Item item2 = new Item(new string[] { "armor" }, "shield", "this is a
shield");
22         player.Inventory.Put(item1);
23         player.Inventory.Put(item2);
24
25         Bag bag = new Bag(new string[] { "bag" }, "bag", "This is a bag.");
26         player.Inventory.Put(bag);
27
28         Item itemInBag = new Item(new string[] { "gem" }, "ruby", "This is a
beautiful gem");
29         bag.Inventory.Put(itemInBag);
30
31         bool exitRequested = false;
32
33         while (!exitRequested)
34         {
35             Console.WriteLine("Enter a command (or type 'exit' to quit):");
36             string input = Console.ReadLine();
37             string[] inputArray = input.Split(' ');
38
39             if (inputArray.Length > 0)

```

```
40     {
41         string command = inputArray[0].ToLower();
42
43         if (command == "exit" || command == "quit")
44         {
45             exitRequested = true;
46
47         }
48         else
49         {
50
51             LookCommand lookCommand = new LookCommand();
52             string result = lookCommand.Execute(player, inputArray);
53
54             Console.WriteLine(result);
55         }
56     }
57 }
58 }
59 }
```

Swin-Adventure/Swin-Adventure.csproj

```
1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <PropertyGroup>
4          <OutputType>Exe</OutputType>
5          <TargetFramework>net7.0</TargetFramework>
6          <RootNamespace>Swin_Adventure</RootNamespace>
7          <ImplicitUsings>enable</ImplicitUsings>
8          <Nullable>enable</Nullable>
9      </PropertyGroup>
10
11  </Project>
12
```

Folder IdentifiableObjectTest

8 printable files

(file list disabled)

IdentifiableObjectTest/BagTest.cs

```
1 using System;
2 namespace Swin_Adventure
3 {
4     [TestFixture]
5     public class BagTest
6     {
7         private Bag _bagTest1;
8         private Bag _bagTest2;
9         private Item _weaponTest;
10        private Item _armorTest;
11
12        [SetUp]
13        public void Setup()
14        {
15            _bagTest1 = new Bag(new string[] { "bag1" }, "backpack", "It's
16            spacious");
17            _bagTest2 = new Bag(new string[] { "bag2" }, "suitcase", "It's compact"
18            );
19            _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is an
20            Excalibur");
21            _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
22            shield");
23
24            _bagTest1.Inventory.Put(_bagTest2);
25            _bagTest1.Inventory.Put(_weaponTest);
26            _bagTest2.Inventory.Put(_armorTest);
27        }
28
29        [Test]
30        public void TestBagLocatesItems()
31        {
32            Assert.AreSame(_weaponTest, _bagTest1.Locate("weapon"));
33        }
34
35        [Test]
36        public void TestBagLocatesitself()
37        {
38            Assert.AreSame(_bagTest1, _bagTest1.Locate("bag1"));
39        }
40
41        [Test]
42        public void TestBagLocatesnothing()
43        {
44            Assert.IsNull(_bagTest1.Locate("bag3"));
45        }
46
47        [Test]
48        public void TestBagFullDescription()
49        {
50        }
```

```

46         Assert.AreEqual("In the backpack you can see:\n\ta suitcase (bag2)\n\ta
sword (weapon)\n", _bagTest1.FullDescription);
47     }
48
49     [Test]
50     public void TestBaginBag()
51     {
52         Assert.AreSame(_bagTest2, _bagTest1.Locate("bag2"));
53         Assert.AreSame(_weaponTest, _bagTest1.Locate("weapon"));
54         Assert.IsNull(_bagTest1.Locate("armor"));
55     }
56 }
57 }
58
59

```

IdentifiableObjectTest/IdentifiableObjectTest.cs

```

1  using NUnit.Framework;
2  using Swin_Adventure;
3
4  namespace IdentifiableObjectTest
5  {
6
7      internal class Tests
8      {
9          private IdentifiableObject _test1;
10         private IdentifiableObject _test2;
11         private IdentifiableObject _test3;
12         private IdentifiableObject _test4;
13         private IdentifiableObject _test5;
14         private IdentifiableObject _test6;
15
16         [SetUp]
17         public void Setup()
18         {
19             _test1 = new IdentifiableObject(new string[] { "fred", "bob" });
20             _test2 = new IdentifiableObject(new string[] { "fred", "bob" });
21             _test3 = new IdentifiableObject(new string[] { "fred", "bob" });
22             _test4 = new IdentifiableObject(new string[] { "fred", "bob" });
23             _test5 = new IdentifiableObject(new string[] { });
24             _test6 = new IdentifiableObject(new string[] { "fred", "bob" });
25         }
26
27         [Test]
28         public void TestAreYou()
29         {
30             Assert.IsTrue(_test1.AreYou("fred"));
31             Assert.IsTrue(_test1.AreYou("bob"));
32         }
33
34         [Test]
35         public void TestNotAreYou()
36         {
37             Assert.IsFalse(_test2.AreYou("wilma"));
38             Assert.IsFalse(_test2.AreYou("boby"));
39         }
40
41         [Test]

```

```

42 |     public void TestCaseSensitive()
43 |     {
44 |         Assert.IsTrue(_test3.AreYou("FRED"));
45 |         Assert.IsTrue(_test3.AreYou("b0B"));
46 |     }
47 |
48 |     [Test]
49 |     public void TestFirstID()
50 |     {
51 |         Assert.AreEqual("fred", _test4.FirstId);
52 |     }
53 |
54 |     [Test]
55 |     public void TestFirstIdWithNoIDs()
56 |     {
57 |         Assert.AreEqual("", _test5.FirstId);
58 |     }
59 |
60 |     [Test]
61 |     public void TestAddID()
62 |     {
63 |         _test6.AddIdentifier("wilma");
64 |         Assert.IsTrue(_test6.AreYou("fred"));
65 |         Assert.IsTrue(_test6.AreYou("bob"));
66 |         Assert.IsTrue(_test6.AreYou("wilma"));
67 |     }
68 | }
69 | }
70 |
71 |

```

IdentifiableObjectTest/InventoryTest.cs

```

1 | using System;
2 | using Swin_Adventure;
3 |
4 | namespace SwinAdventureTest
5 | {
6 |     [TestFixture]
7 |     public class InventoryTest
8 |     {
9 |         private Inventory _inventoryTest;
10 |        private Item _weaponTest;
11 |        private Item _armorTest;
12 |
13 |        [SetUp]
14 |        public void Setup()
15 |        {
16 |            _inventoryTest = new Inventory();
17 |            _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is
an Excalibur");
18 |            _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
shield");
19 |
20 |            _inventoryTest.Put(_weaponTest);
21 |            _inventoryTest.Put(_armorTest);
22 |        }
23 |
24 |        [Test]

```

```

25 public void TestFindItem()
26 {
27     Assert.IsTrue(_inventoryTest.HasItem("weapon"));
28     Assert.IsTrue(_inventoryTest.HasItem("armor"));
29 }
30
31 [Test]
32 public void TestNoItemFind()
33 {
34     Assert.IsFalse(_inventoryTest.HasItem("axe"));
35     Assert.IsFalse(_inventoryTest.HasItem("helmet"));
36 }
37
38 [Test]
39 public void TestFetchItem()
40 {
41     Assert.IsTrue(_weaponTest == _inventoryTest.Fetch("weapon"));
42     Assert.IsTrue(_inventoryTest.HasItem("weapon"));
43
44     Assert.IsTrue(_armorTest == _inventoryTest.Fetch("armor"));
45     Assert.IsTrue(_inventoryTest.HasItem("armor"));
46 }
47
48 [Test]
49 public void TestTakeItem()
50 {
51     Assert.IsTrue(_weaponTest == _inventoryTest.Take("weapon"));
52     Assert.IsFalse(_inventoryTest.HasItem("weapon"));
53
54     Assert.IsTrue(_armorTest == _inventoryTest.Take("armor"));
55     Assert.IsFalse(_inventoryTest.HasItem("armor"));
56 }
57
58 [Test]
59 public void TestItemList()
60 {
61     Assert.IsTrue(_inventoryTest.ItemList.Replace("\t", "") == "a sword
62 (weapon)\na shield (armor)\n");
63 }
64 }
65
66

```

IdentifiableObjectTest/ItemTest.cs

```

1 using System;
2 using Swin_Adventure;
3
4 namespace SwinAdventureTest
5 {
6     [TestFixture]
7     public class ItemTest
8     {
9         private Item _itemTest;
10
11
12         [SetUp]
13         public void Setup()

```

```

14     {
15         _itemTest = new Item(new string[] { "weapon" }, "sword", "This is an
Excalibur");
16     }
17
18     [Test]
19     public void TestItemIsIdentifiable()
20     {
21         Assert.IsTrue(_itemTest.AreYou("weapon"));
22     }
23
24     [Test]
25     public void TestShortDescription()
26     {
27         Assert.IsTrue(_itemTest.ShortDescription == "a sword (weapon)");
28     }
29
30     [Test]
31     public void TestFullDescription()
32     {
33         Assert.IsTrue(_itemTest.FullDescription == "This is an Excalibur");
34     }
35 }
36 }
37 }
38
39

```

IdentifiableObjectTest/LookCommandTest.cs

```

1  using NUnit.Framework;
2  using System.Numerics;
3  using Swin_Adventure;
4
5  namespace SwinAdventureTest
6  {
7      [TestFixture]
8      public class TestLookCommand
9      {
10         private LookCommand _lookCommandTest;
11         private Player _playerTest;
12         private Bag _bagTest;
13         private Item _gemTest;
14
15         [SetUp]
16         public void Setup()
17         {
18             _lookCommandTest = new LookCommand();
19             _playerTest = new Player("thuan", "dan choi");
20             _bagTest = new Bag(new string[] { "duffelbag" }, "duffelbag", "it's
small-sized");
21             _gemTest = new Item(new string[] { "gem" }, "gem", "a beautiful gem");
22         }
23
24         [Test]
25         public void TestLookAtMe()
26         {
27             Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
}; "at", "inventory" }), Is.EqualTo("You are thuan, dan choi.\nYou are carrying:\n")

```



```
28     }
29
30     [Test]
31     public void TestLookAtGem()
32     {
33         _playerTest.Inventory.Put(_gemTest);
34
35         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
36 , "at", "gem" }), Is.EqualTo("a beautiful gem"));
37     }
38
39     [Test]
40     public void TestLookAtUnk()
41     {
42         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
43 , "at", "unknown" }), Is.EqualTo("I can't find the unknown"));
44     }
45
46     [Test]
47     public void TestLookAtGemInMe()
48     {
49         _playerTest.Inventory.Put(_gemTest);
50
51         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
52 , "at", "gem", "in", "inventory" }), Is.EqualTo("a beautiful gem"));
53     }
54
55     [Test]
56     public void TestLookAtGemInBag()
57     {
58         _bagTest.Inventory.Put(_gemTest);
59         _playerTest.Inventory.Put(_bagTest);
60
61         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
62 , "at", "gem", "in", "duffelbag" }), Is.EqualTo("a beautiful gem"));
63     }
64
65     [Test]
66     public void TestLookAtGemInNoBag()
67     {
68         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
69 , "at", "gem", "in", "duffelbag" }), Is.EqualTo("I can't find the duffelbag"));
70     }
71
72     [Test]
73     public void TestLookAtNoGemInBag()
74     {
75         _playerTest.Inventory.Put(_bagTest);
76
77         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
78 , "at", "gem", "in", "duffelbag" }), Is.EqualTo("I can't find the gem"));
79     }
80
81     [Test]
82     public void TestInvalidLook()
83     {
84         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
85 , "around" }), Is.EqualTo("I don't know how to look like that"));
86         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "
87 hello" }), Is.EqualTo("I don't know how to look like that"));
88         Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
89 , "at", "a", "at", "b" }), Is.EqualTo("What do you want to look in?"));
```

```

81     Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "
hello", "at", "a" }), Is.EqualTo("Error in look input"));
82     Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look"
, "by", "a" }), Is.EqualTo("What do you want to look at?"));
83     }
84     }
85 }
86

```

IdentifiableObjectTest/PlayerTest.cs

```

1  using System;
2  using Swin_Adventure;
3
4  namespace SwinAdventureTest
5  {
6      [TestFixture]
7      public class PlayerTest
8      {
9          private Player _playerTest;
10         private Item _weaponTest;
11         private Item _armorTest;
12
13         [SetUp]
14         public void Setup()
15         {
16             _playerTest = new Player("thuan", "dan choi");
17             _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is an
Excalibur");
18             _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
shield");
19
20
21
22             _playerTest.Inventory.Put(_weaponTest);
23             _playerTest.Inventory.Put(_armorTest);
24         }
25
26         [Test]
27         public void TestPlayerIsIdentifiable()
28         {
29             Assert.IsTrue(_playerTest.AreYou("me"));
30             Assert.IsTrue(_playerTest.AreYou("inventory"));
31         }
32
33         [Test]
34         public void TestPlayerLocateItems()
35         {
36             Assert.IsTrue(_playerTest.Locate("weapon") == _weaponTest);
37             Assert.IsTrue(_playerTest.Locate("armor") == _armorTest);
38
39             Assert.IsTrue(_playerTest.Inventory.HasItem("weapon"));
40             Assert.IsTrue(_playerTest.Inventory.HasItem("armor"));
41         }
42
43         [Test]
44         public void TestPlayerLocateItself()
45         {
46             Assert.IsTrue(_playerTest == _playerTest.Locate("me"));
47             Assert.IsTrue(_playerTest == _playerTest.Locate("inventory"));

```

```

48     }
49
50     [Test]
51     public void TestPlayerLocateNothing()
52     {
53         Assert.IsTrue(_playerTest.Locate("helmet") == null);
54     }
55
56     [Test]
57     public void TestPlayerFullDescription()
58     {
59         Assert.IsTrue(_playerTest.FullDescription == "You are thuan, dan
choi.\nYou are carrying:\n\t a sword (weapon)\n\t a shield (armor)\n");
60     }
61 }
62 }
63
64

```

IdentifiableObjectTest/Swin-Adventure-Test.csproj

```

1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <PropertyGroup>
4          <TargetFramework>net7.0</TargetFramework>
5          <ImplicitUsings>enable</ImplicitUsings>
6          <Nullable>enable</Nullable>
7
8          <IsPackable>false</IsPackable>
9          <IsTestProject>true</IsTestProject>
10     </PropertyGroup>
11
12     <ItemGroup>
13         <PackageReference Include="Microsoft.NET.Test.Sdk" Version="17.5.0" />
14         <PackageReference Include="NUnit" Version="3.13.3" />
15         <PackageReference Include="NUnit3TestAdapter" Version="4.4.2" />
16         <PackageReference Include="NUnit.Analyzers" Version="3.6.1" />
17         <PackageReference Include="coverlet.collector" Version="3.2.0" />
18     </ItemGroup>
19
20     <ItemGroup>
21         <ProjectReference Include="..\Swin-Adventure\Swin-Adventure.csproj" />
22     </ItemGroup>
23 </Project>
24

```

IdentifiableObjectTest/Usings.cs

```

1  global using NUnit.Framework;
2

```