

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 2 - Players Items and Inventory

PDF generated at 00:20 on Thursday 28th September, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ass24
8  {
9      public abstract class GameObject : IdentifiableObject
10     {
11         private string _description;
12         private string _name;
13
14         public GameObject(string[] id, string name, string desc) :base(id)
15         {
16             _description = desc;
17             _name = name;
18         }
19         public string Name { get { return _name; } }
20         public string ShortDescription { get {
21             string _shortdesc = _name + " " + "(" + FirstId+");";
22             return _shortdesc; } } //name and FirstId
23         public virtual string FullDescription { get { return _description; } }
24     }
25 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Xml.Linq;
7
8  namespace Ass24
9  {
10     public class Player :GameObject
11     {
12         private Inventory _inventory;
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
21             if (this.AreYou(id)) { return this; }
22             else
23             {
24                 return this._inventory.Fetch(id);
25             }
26
27         }
28
29         public override string FullDescription { get {
30             string fulldesc = Name + FullDescription + _inventory;
31             return fulldesc; } }
32
33         public Inventory Inventory { get { return _inventory; } }
34     }
35 }
```

```
1 using Ass24;
2 using System.Numerics;
3 //was not running, solution: run from solution
4
5 namespace PlayerTest
6 {
7     public class TestPlayer
8     {
9         Ass24.Player _player;
10        [SetUp]
11        public void SetUp() {
12            _player = new Player("id1", "id2");
13        }
14        [Test]
15        public void PlayerIdentifierTest() { Assert.IsTrue(_player.AreYou("me")); }
16        [Test]
17        public void PlayerLocaterTest() { Assert.IsTrue(_player.Locate("me") ==
↵ _player); }
18        [Test]
19        public void PlayerItemLocaterTest()
20        {
21            Item TestItem = new Item(new string[] { "sheild", "hard" }, "hard
↵ sheild", "a hardsheild");
22            _player.Inventory.Put(TestItem);
23            Assert.IsTrue(_player.Locate("sheild") == TestItem);
24        }
25
26        [Test]
27        public void PlayerItemNotlocated() { Assert.IsFalse(_player.Locate("nope") ==
↵ _player); }
28
29        [Test]
30        public void PlayerFullDescription() { Assert.IsTrue(_player.FullDescription
↵ == ""); }
31
32    }
33 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ass24
8  {
9      public class Item:GameObject
10     {
11         public Item(string[] idents,string name, string desc) :base(idents, name,
↵ desc)
12         {
13             foreach(var ident in idents)
14             {
15                 this.AddIdentifier(ident);
16             }
17         }
18     }
19 }
20 }
```

```
1
2 using Ass24;
3
4
5 namespace Test42
6 {
7     public class Tests
8     {
9         Ass24.Item _identifiable;
10        [SetUp]
11        public void Setup()
12        {
13            _identifiable = new Item(new string[] { "id1", "id2" }, "Lily", "a tired
↪ programmer");
14        }
15
16        //item unit tests
17
18        [Test]
19        public void Test1()
20        {
21            _identifiable.AreYou("id1");
22            Assert.IsTrue(_identifiable.AreYou("id1"));
23        }
24
25        [Test]
26        public void Test2()
27        {
28            string result = _identifiable.ShortDescription;
29            Assert.AreEqual("Lily (id1)", result);
30        }
31
32        [Test]
33        public void Test3()
34        {
35            string result = _identifiable.FullDescription;
36            Assert.AreEqual("a tired programmer", result);
37        }
38    }
39 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ass24
8  {
9      public class Inventory
10     {
11         private List<Item> _items;
12         public Inventory()
13         {
14             _items = new List<Item>();
15         }
16         public bool HasItem(string id)
17         {
18             foreach (Item item in _items)
19             {
20                 if (item.AreYou(id))
21                     return true;
22             }
23             return false;
24         }
25
26         public void Put(Item itm) { _items.Add(itm);}
27         public Item Take(string id) {
28
29             foreach (Item item in _items)
30             {
31                 if (item.AreYou(id))
32                     _items.Remove(item);
33                 return item;
34             }
35
36             return null;
37         }
38
39         public Item Fetch(string id) {
40             foreach (Item item in _items)
41             {
42                 if (item.AreYou(id))
43                     return item;
44             }
45
46             return null;
47         }
48
49         public string ItemList { get {
50             string result = "";
51             foreach (Item item in _items)
52             {
53                 result = item.ShortDescription;
```

```
54         }  
55  
56         return result; } }  
57  
58  
59     }  
60 }
```



```
1  using Ass24;
2
3  namespace InventoryUnitTest
4  {
5      public class Tests
6      {
7          Ass24.Inventory _inventory;
8          Ass24.Item _item;
9          [SetUp]
10         public void Setup()
11         {
12             _item = new Item(new string[] { "id1", "id2" }, "Lily", "a tired
↪ programmer");
13             _inventory = new Ass24.Inventory();
14             _inventory.Put(_item);
15         }
16
17         [Test]
18         public void FindItem()
19         {
20             bool result = _inventory.HasItem("id1");
21             Assert.IsTrue(result);
22         }
23
24         [Test]
25         public void NoItemFind()
26         {
27             bool result = _inventory.HasItem("nope");
28             Assert.IsFalse(result);
29         }
30         [Test]
31         public void FetchItem()
32         {
33             Item result = _inventory.Fetch("id1");
34             Assert.AreEqual(_item, result);
35         }
36         [Test]
37         public void TakeItem()
38         {
39             Item result = _inventory.Take("id1");
40             Assert.AreEqual(_item, result);
41         }
42         [Test]
43         public void ItemList()
44         {
45             string result = _inventory.ItemList;
46             Assert.AreEqual(_item.ShortDescription, result);
47         }
48     }
49 }
```

