

GameObject.cs

...Desktop\COS20007\Week_4\4.2P\Iteration1\GameObject.cs

1

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.Design;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Iteration1
9 {
10     public abstract class GameObject : IdentifiableObject
11     {
12         private string _description;
13         private string _name;
14
15         public GameObject(string[] ids, string name, string description) : ↗
16             base(ids)
17         {
18             _description = description;
19             _name = name;
20         }
21
22         public string Name
23         {
24             get => _name;
25         }
26
27         public string ShortDescription
28         {
29             get => $"a {Name} ({FirstId()})";
30         }
31
32         public virtual string FullDescription
33         {
34             get => _description;
35         }
36     }
37 }
```

IdentifiableObject.cs

...COS20007\Week_4\4.2P\Iteration1\IdentifiableObject.cs

1

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Iteration1
8 {
9     public class IdentifiableObject
10    {
11        private List<string> _identifiers = new List<string>();
12
13        public IdentifiableObject(string[] idents)
14        {
15            for (int i = 0; i < idents.Length; i++)
16            {
17                _identifiers.Add(idents[i].ToLower());
18            }
19        }
20        public bool AreYou(string id)
21        {
22            if (_identifiers.Contains(id.ToLower()))
23            {
24                return true;
25            }
26            return false;
27        }
28        public string FirstId()
29        {
30            if (_identifiers.Any())
31            {
32                return _identifiers.First();
33            }
34            return "";
35        }
36        public void AddIdentifier(string id)
37        {
38            _identifiers.Add(id.ToLower());
39        }
40    }
41 }
42
```

Inventory.cs

...\Desktop\COS20007\Week_4\4.2P\Iteration1\Inventory.cs

1

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.IO.Pipes;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace Iteration1
10 {
11     public class Inventory
12     {
13         private List<Item> _items;
14
15         public Inventory()
16         {
17             _items = new List<Item>();
18         }
19
20         public bool HasItem(string id)
21         {
22             foreach (Item itm in _items)
23             {
24                 if (itm.AreYou(id))
25                 {
26                     return true;
27                 }
28             }
29             return false;
30         }
31
32         public void Put(Item itm)
33         {
34             _items.Add(itm);
35         }
36
37         public Item Fetch(string id)
38         {
39             foreach (Item itm in _items)
40             {
41                 if (itm.AreYou(id))
42                 {
43                     return itm;
44                 }
45             }
46             return null;
47         }
48
49         public Item Take(string id)
```

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

Item.cs

C:\Users\PC\Desktop\COS20007\Week_4\4.2P\Iteration1\Item.cs

1

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Iteration1
8 {
9     public class Item : GameObject
10    {
11        public Item(string[] idents, string name, string description) :
12            base(idents, name, description)
13        {
14        }
15    }
16 }
```

Player.cs

...\PC\Desktop\COS20007\Week_4\4.2P\Iteration1\Player.cs

1

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Reflection.Metadata.Ecma335;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Iteration1
9 {
10     public class Player : GameObject
11     {
12         private Inventory _inventory;
13
14         public Player(string name, string desc) : base(new string[] { "me", ↵
15             "inventory" }, name, desc)
16         {
17             _inventory = new Inventory();
18         }
19
20         public GameObject Locate(string id)
21         {
22             if (AreYou(id))
23             {
24                 return this;
25             }
26             else if (_inventory.HasItem(id))
27             {
28                 return _inventory.Fetch(id);
29             }
30             return null;
31         }
32
33         public override string FullDescription
34         {
35             get
36             {
37                 string fulldesc = "";
38                 fulldesc += $"You are {Name}, {base.FullDescription}\n";
39                 fulldesc += "You are carrying\n";
40                 fulldesc += $"{_inventory.ItemList}";
41                 return fulldesc;
42             }
43         }
44
45         public Inventory Inventory
46         {
47             get => _inventory;
48         }
49     }
50 }
```

49 }

50

Program.cs

...PC\Desktop\COS20007\Week_4\4.2P\Iteration1\Program.cs

1

```
1 namespace Iteration1
2 {
3     internal class Program
4     {
5         static void Main(string[] args)
6         {
7             Console.WriteLine("Hello, World!");
8         }
9     }
10 }
11
```


InventoryUnitTests.cs

...\COS20007\Week_4\4.2P\InventoryUnitTests\UnitTest1.cs

1

```
1 using Iteration1;
2
3 namespace InventoryUnitTests
4 {
5     public class Tests
6     {
7         Inventory ivt;
8         Item sword;
9         Item ak47;
10
11         [SetUp]
12         public void Setup()
13         {
14             ivt = new Inventory();
15             sword = new Item(new string[] { "sword", "melee" }, "bronze ↗
                sword", "Melee weapon. High damage.");
16             ak47 = new Item(new string[] { "ak47", "gun" }, "ak47", "Gun. ↗
                High Damage.");
17             ivt.Put(sword);
18             ivt.Put(ak47);
19         }
20
21         [Test]
22         public void TestFindItem()
23         {
24             Assert.That(ivt.HasItem("sword"), Is.EqualTo(true));
25             Assert.Pass();
26         }
27
28         [Test]
29         public void TestNoItemFind()
30         {
31             Assert.That(ivt.HasItem("machine gun"), Is.EqualTo(false));
32             Assert.Pass();
33         }
34
35         [Test]
36         public void TestFetchItem()
37         {
38             Assert.That(ivt.Fetch("sword"), Is.EqualTo(sword));
39             Assert.That(ivt.HasItem("sword"), Is.EqualTo(true));
40             Assert.Pass();
41         }
42
43         [Test]
44         public void TestTakeItem()
45         {
46             Assert.That(ivt.Take("ak47"), Is.EqualTo(ak47));
47             Assert.That(ivt.HasItem("ak47"), Is.EqualTo(false));
```

```
48         Assert.Pass();
49     }
50
51     [Test]
52     public void TestItemList()
53     {
54         Assert.That(ivt.ItemList, Is.EqualTo("    a bronze sword
55         (sword)\n    a ak47 (ak47)\n"));
56         Assert.Pass();
57     }
58 }
```

ItemUnitTests

...sktop\COS20007\Week_4\4.2P\ItemUnitTests\UnitTest1.cs

1

```
1 using Iteration1;
2
3 namespace ItemUnitTests
4 {
5     public class Tests
6     {
7         Item itm;
8         [SetUp]
9         public void Setup()
10        {
11            itm = new Item(new string[] { "sword", "great"}, "bronze sword",
12                           "Melee weapon. High damage.");
13
14            [Test]
15            public void TestItemIsIdentifiable()
16            {
17                Assert.That(itm.AreYou("sword"), Is.EqualTo(true));
18                Assert.Pass();
19            }
20
21            [Test]
22            public void TestShortDescription()
23            {
24                Assert.That(itm.ShortDescription, Is.EqualTo("a bronze sword
25                           (sword)"));
26                Assert.Pass();
27            }
28
29            [Test]
30            public void TestFullDescription()
31            {
32                Assert.That(itm.FullDescription, Is.EqualTo("Melee weapon. High
33                           damage. "));
34                Assert.Pass();
35            }
36        }
37    }
38 }
```

PlayerUnitTests

...top\COS20007\Week_4\4.2P\PlayerUnitTests\UnitTest1.cs

1

```
1 using Iteration1;
2
3 namespace PlayerUnitTests
4 {
5     public class Tests
6     {
7         Player player;
8         Item sword;
9         Item ak47;
10
11         [SetUp]
12         public void Setup()
13         {
14             player = new Player("Chien", "A boy with high curiosity.");
15             sword = new Item(new string[] { "sword", "melee" }, "bronze sword", "Melee weapon. High damage.");
16             ak47 = new Item(new string[] { "ak47", "gun" }, "ak47", "Gun. High Damage.");
17             player.Inventory.Put(sword);
18             player.Inventory.Put(ak47);
19         }
20
21         [Test]
22         public void TestPlayerIdentifiable()
23         {
24             Assert.That(player.AreYou("inventory"), Is.EqualTo(true));
25             Assert.Pass();
26         }
27
28         [Test]
29         public void TestPlayerLocatesItem()
30         {
31             Assert.That(player.Locate("sword"), Is.EqualTo(sword));
32             Assert.That(player.Inventory.HasItem("sword"), Is.EqualTo(true));
33             Assert.Pass();
34         }
35
36         [Test]
37         public void TestPlayerLocatesItself()
38         {
39             Assert.That(player.Locate("inventory"), Is.EqualTo(player));
40             Assert.That(player.Locate("me"), Is.EqualTo(player));
41             Assert.Pass();
42         }
43
44         [Test]
45         public void TestPlayerLocatesNothing()
46         {
```

```
47         Assert.That(player.Locate("machine gun"), Is.EqualTo(null));
48         Assert.Pass();
49     }
50
51     [Test]
52     public void TestPlayerFullDescription()
53     {
54         Assert.That(player.FullDescription, Is.EqualTo("You are Chien,
55             A boy with high curiosity.\nYou are carrying\n    a bronze
56             sword (sword)\n    a ak47 (ak47)\n"));
57         Assert.Pass();
58     }
59 }
```

TestIdentifiableObjects

...20007\Week_4\4.2P\TestIdentifiableObject\UnitTest1.cs

1

```
1 using Iteration1;
2
3 namespace TestIdentifiableObject
4 {
5     public class Tests
6     {
7         private IdentifiableObject myObject;
8
9         [SetUp]
10        public void SetUp()
11        {
12            myObject = new IdentifiableObject(new string[] { "fred",
13                "bob" });
14
15        [Test]
16        public void TestAreYou()
17        {
18            Assert.That(myObject.AreYou("bob"), Is.EqualTo(true));
19            Assert.That(myObject.AreYou("fred"), Is.EqualTo(true));
20
21            Assert.Pass();
22        }
23
24        [Test]
25        public void TestNotAreYou()
26        {
27            Assert.That(myObject.AreYou("wilma"), Is.EqualTo(false));
28            Assert.That(myObject.AreYou("boby"), Is.EqualTo(false));
29
30            Assert.Pass();
31        }
32
33        [Test]
34        public void TestCaseSentive()
35        {
36            Assert.That(myObject.AreYou("FRED"), Is.EqualTo(true));
37            Assert.That(myObject.AreYou("bOB"), Is.EqualTo(true));
38
39            Assert.Pass();
40        }
41
42        [Test]
43        public void TestFirstId()
44        {
45            Assert.That(myObject.FirstId(), Is.EqualTo("fred"));
46
47            Assert.Pass();
48        }
49    }
50 }
```

```
49
50     [Test]
51     public void TestFirstIdWithoutIDs()
52     {
53         IdentifiableObject myObject = new IdentifiableObject(new string[] { });
54
55         Assert.That(myObject.FirstId(), Is.EqualTo(""));
56
57         Assert.Pass();
58     }
59
60     [Test]
61     public void TestAddId()
62     {
63         myObject.AddIdentifier("wilma");
64
65         Assert.That(myObject.AreYou("fred"), Is.EqualTo(true));
66         Assert.That(myObject.AreYou("bob"), Is.EqualTo(true));
67         Assert.That(myObject.AreYou("wilma"), Is.EqualTo(true));
68
69         Assert.Pass();
70     }
71 }
72 }
```

Screenshot of Test Passing

