

```
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 IdentifiableObject
10    {
11        private List<string> _identifiers;
12
13        public IdentifiableObject(string[] idents)
14        {
15            _identifiers = new List<string>();
16            foreach (string id in idents)
17            {
18                _identifiers.Add(id.ToLower());
19            }
20        }
21
22        public bool AreYou(string id)
23        {
24            return _identifiers.Contains(id.ToLower());
25        }
26
27        public string FirstId
28        {
29            get
30            {
31                if (_identifiers.Count > 0)
32                {
33                    return _identifiers[0];
34                }
35                else
36                {
37                    return "";
38                }
39            }
40        }
41
42        public void AddIdentifier(string id)
43        {
44            _identifiers.Add(id.ToLower());
45        }
46    }
47 }
48
```

```
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 GameObject : IdentifiableObject
10    {
11        private string _name;
12        private string _description;
13        private string[] _ids;
14
15        public GameObject(string[] ids, string name, string desc) : base  ➤
16            (ids)
17        {
18            _name = name;
19            _description = desc;
20            _ids = ids;
21        }
22
23        public string Name
24        {
25            get
26            {
27                return _name;
28            }
29        }
30
31        public string ShortDescription
32        {
33            get
34            {
35                return _name + " (" + _ids[0] + ")";
36            }
37        }
38
39        public virtual string FullDescription
40        {
41            get
42            {
43                return "The " + _name + " (" + _ids[0] + ")" + " " +  ➤
44                    _description;
45            }
46        }
47    }
```

```
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
12            (idents, name, desc)
13        {
14        }
15    }
16 }
17
```

```
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 Inventory
10    {
11        List<Item> _items;
12
13        public Inventory()
14        {
15            _items = new List<Item>();
16        }
17
18        public bool HasItem(string id)
19        {
20            foreach (Item i in _items)
21            {
22                if(i.AreYou(id))
23                {
24                    return true;
25                }
26            }
27            return false;
28        }
29
30        public void Put(Item itm)
31        {
32            _items.Add(itm);
33        }
34
35        public Item? Take(string id)
36        {
37            Item? t = null;
38            foreach (Item i in _items)
39            {
40                if(i.AreYou(id))
41                {
42                    t = i;
43                    _items.Remove(i);
44                    return t;
45                }
46            }
47            return t;
48        }
49    }
```

```
50     public Item? Fetch(string id)
51     {
52         Item? t = null;
53         foreach (Item i in _items)
54         {
55             if (i.AreYou(id))
56             {
57                 return i;
58             }
59         }
60         return t;
61     }
62
63     public string ItemList
64     {
65         get
66         {
67             string list = "";
68             foreach (Item i in _items)
69             {
70                 list += i.ShortDescription + "\n";
71             }
72             return list;
73         }
74     }
75 }
76 }
77
```

```
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        private Inventory _inventory;
12        string _name;
13        string _desc;
14
15        public Player(string name, string desc) : base(new string[] { "me", ↵
16            "inventory" }, name, desc)
17        {
18            _inventory = new Inventory();
19            _name = name;
20            _desc = desc;
21        }
22
23        public GameObject? Locate(string id)
24        {
25            if (AreYou(id))
26            {
27                return this;
28            }
29            return _inventory.Fetch(id);
30        }
31
32        public override string FullDescription
33        {
34            get
35            {
36                return "You are " + _name + ", " + _desc + ". You are ↵
37                    carrying: " + _inventory.ItemList;
38            }
39        }
40
41        public Inventory Inventory
42        {
43            get
44            {
45                return _inventory;
46            }
47        }
48    }
49 }
```

```
1 using SwinAdventure;
2
3 namespace ItemTest
4 {
5     public class ItemTest
6     {
7         Item BronzeSword = new Item(new string[] { "sword", "weapon" },
8             "Bronze Sword", "A simple bronze sword.");
9         Item BronzeAxe = new Item(new string[] { "axe", "weapon" }, "Bronze
10             Axe", "A simple bronze axe.");
11         [SetUp]
12         public void Setup()
13         {
14
15         }
16         [Test]
17         public void TestItemIsIdentifiable()
18         {
19             bool SwordIsIdentifiable = BronzeSword.AreYou("sword");
20             bool AxeIsIdentifiable = BronzeAxe.AreYou("sword");
21
22             Assert.IsTrue(SwordIsIdentifiable);
23             Assert.IsFalse(AxeIsIdentifiable);
24         }
25         [Test]
26         public void TestShortDescription()
27         {
28             string SwordShortDesc = BronzeSword.ShortDescription;
29             string AxeShortDesc = BronzeAxe.ShortDescription;
30
31             Assert.AreEqual("Bronze Sword (sword)", SwordShortDesc);
32             Assert.AreEqual("Bronze Axe (axe)", AxeShortDesc);
33         }
34         [Test]
35         public void TestFullDescription()
36         {
37             string SwordFullDesc = BronzeSword.FullDescription;
38             string AxeFullDesc = BronzeAxe.FullDescription;
39
40             Assert.AreEqual("The Bronze Sword (sword) A simple bronze
41                 sword.", SwordFullDesc);
42             Assert.AreEqual("The Bronze Axe (axe) A simple bronze axe.",
43                 AxeFullDesc);
44         }
45     }
```

```
1 using SwinAdventure;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace InventoryTest
9 {
10     internal class InventoryTest
11     {
12         Item BronzeSword = new Item(new string[] { "sword", "weapon" },
13             "Bronze Sword", "A simple bronze sword.");
14         Item BronzeAxe = new Item(new string[] { "axe", "weapon" }, "Bronze
15             Axe", "A simple bronze axe.");
16         [SetUp]
17         public void Setup()
18         {
19
20         }
21         [Test]
22         public void FindItemTest()
23         {
24             Inventory inv = new Inventory();
25
26             inv.Put(BronzeAxe);
27             inv.Put(BronzeSword);
28
29             bool sword = inv.HasItem("sword");
30             bool axe = inv.HasItem("axe");
31
32             Assert.IsTrue(sword);
33             Assert.IsTrue(axe);
34         }
35         [Test]
36         public void NoItemFindTest()
37         {
38             Inventory inv = new Inventory();
39
40             Assert.IsFalse(inv.HasItem("sword"));
41             Assert.IsFalse(inv.HasItem("axe"));
42         }
43         [Test]
44         public void FetchItemTest()
45         {
46             Inventory inv = new Inventory();
```



```
48
49         inv.Put(BronzeAxe);
50         inv.Put(BronzeSword);
51
52         Item sword = inv.Fetch("sword");
53         Item axe = inv.Fetch("axe");
54
55         Assert.AreEqual(BronzeSword, sword);
56         Assert.AreEqual(BronzeAxe, axe);
57     }
58
59     [Test]
60     public void TakeItemTest()
61     {
62         Inventory inv = new Inventory();
63
64         inv.Put(BronzeAxe);
65         inv.Put(BronzeSword);
66
67         Item sword = inv.Take("sword");
68         Item axe = inv.Take("axe");
69
70         Item SwordRemain = inv.Fetch("sword");
71         Item AxeRemain = inv.Fetch("axe");
72
73         Assert.AreEqual(BronzeSword, sword);
74         Assert.AreEqual(BronzeAxe, axe);
75         Assert.IsTrue(SwordRemain == null);
76         Assert.IsTrue(AxeRemain == null);
77     }
78
79     [Test]
80     public void ItemListTest()
81     {
82         Inventory inv = new Inventory();
83
84         inv.Put(BronzeAxe);
85         inv.Put(BronzeSword);
86
87         string list = inv.ItemList;
88
89         Assert.AreEqual("Bronze Axe (axe)\nBronze Sword (sword)\n",
90             list);
91     }
92 }
93
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using SwinAdventure;
7
8 namespace PlayerTest
9 {
10     internal class PlayerTest
11     {
12         Player _player = new Player("Sen", "luv ur mom");
13         Item BronzeSword = new Item(new string[] { "sword", "weapon" },
14             "Bronze Sword", "A simple bronze sword.");
15         [SetUp]
16         public void Setup()
17         {
18         }
19
20         [Test]
21         public void PlayerisIdentifiableTest()
22         {
23             Assert.IsTrue(_player.AreYou("me"));
24             Assert.IsTrue(_player.AreYou("inventory"));
25         }
26
27         [Test]
28         public void PlayerLocateItemTest()
29         {
30             _player.Inventory.Put(BronzeSword);
31
32             Assert.AreEqual(BronzeSword, _player.Locate("sword"));
33         }
34
35         [Test]
36         public void PlayerLocateItselfTest()
37         {
38             Assert.AreEqual(_player, _player.Locate("me"));
39             Assert.AreEqual(_player, _player.Locate("inventory"));
40         }
41
42         [Test]
43         public void PlayerLocateNothingTest()
44         {
45             Assert.IsNull(_player.Locate("sadau"));
46         }
47
48         [Test]
```

```
49     public void PlayerFullDescriptionTest()
50     {
51         _player.Inventory.Put(BronzeSword);
52         string desc = _player.FullDescription;
53
54         Assert.AreEqual("You are Sen, luv ur mom. You are carrying:
55             Bronze Sword (sword)\n", desc);
56     }
57 }
58
```

Test	Duration	T...	Error Message
▲ Test (19)	59 ms		
▸ IdentifiableObject...	58 ms		
▲ InventoryTest (5)	1 ms		
▲ InventoryTest (5)	1 ms		
FetchItemTest	< 1 ms		
FindItemTest	< 1 ms		
ItemListTest	1 ms		
NoItemFindTest	< 1 ms		
TakeItemTest	< 1 ms		
▲ ItemTest (3)	< 1 ms		
▲ ItemTest (3)	< 1 ms		
TestFullDescri...	< 1 ms		
TestItemsIdent...	< 1 ms		
TestShortDesc...	< 1 ms		
▲ PlayerTest (5)	< 1 ms		
▲ PlayerTest (5)	< 1 ms		
PlayerFullDesc...	< 1 ms		
PlayerIsIdentifi...	< 1 ms		
PlayerLocateIt...	< 1 ms		
PlayerLocateIt...	< 1 ms		
PlayerLocateN...	< 1 ms		