```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Hello, World!");
14         }
15
16     }
17 }
18
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public  class GameObject : IdentifiableObject
10     {
11         private string _description;
12         private string _name;
13         public GameObject(string[] ids, string name, string desc) : base
             (ids)
14         {
15             _description = desc;
16             _name = name;
17         }
18         public string name
19         {
20             get { return _name; }
21         }
22         public string ShortDescription
23         {
24             get { return $"{_name} ({FirstID})"; }
25         }
26         public virtual string FullDescription
27         {
28             get { return _description; }
29         }
30     }
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 Iteration2
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 ident in idents)
17             {
18                 _identifiers.Add(ident.ToLower());
19             }
20
21
22         }
23
24         public bool AreYou(string name)
25         {
26             foreach (string idents in _identifiers)
27             {
28                 if (idents.ToLower() == name.ToLower())
29                 {
30                     return true;
31                 }
32             }
33
34             return false;
35         }
36
37         public string FirstID
38         {
39             get
40             {
41                 if (_identifiers.Count == 0)
42                 {
43                     return "";
44                 }
45                 else
46                 {
47                     return _identifiers.First();
48
49                 }
```

```
50              }
51          }
52
53          public void AddIdentifier(string id)
54          {
55              _identifiers.Add(id.ToLower());
56          }
57      }
58  }
59
```

```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Diagnostics;
 4  using System.Linq;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7
 8  namespace Iteration2
 9  {
10      public class Inventory
11      {
12          private List<Item> _items;
13          public Inventory()
14          {
15              _items = new List<Item>();
16          }
17          public bool HasItem(string id)
18          {
19              foreach (var item in _items)
20              {
21                  if (item.AreYou(id))
22                  {
23                      return true;
24                  }
25              }
26              return false;
27          }
28          public void Put(Item i)
29          {
30              _items.Add(i);
31          }
32          public Item Take(string id)
33          {
34              foreach (var item in _items)
35              {
36                  if (item.AreYou(id))
37                  {
38                      _items.Remove(item);
39                      return item;
40                  }
41
42              }
43              return null;
44          }
45          public Item Fetch(string id)
46          {
47              foreach (var item in _items)
48              {
49                  if (item.AreYou(id))
```

```
50                  {
51                      return item;
52
53                  }
54              }
55              return null;
56          }
57          public string ItemList
58          {
59              get
60              {
61                  string listItem = "";
62                  foreach (Item i in _items)
63                  {
64                      listItem = listItem + i.ShortDescription +"\n";
65                  }
66                  return listItem;
67              }
68          }
69      }
70  }
71
72
73
```

```
 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 Iteration2
 9  {
10      public class Item : GameObject
11      {
12          public Item(string[] idents, string name, string desc) : base
              (idents, name, desc)
13          {
14
15          }
16      }
17  }
18
```

```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.IO.Pipes;
 4  using System.Linq;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7  using System.Xml.Linq;
 8
 9  namespace Iteration2
10  {
11      public class Player : GameObject
12      {
13          private Inventory _inventory;
14          public Player(string name, string desc) : base(new string[] { "me",
                "inventory" }, name, desc)
15          {
16              _inventory = new Inventory();
17          }
18          public GameObject Locate(string id)
19          {
20              if (AreYou(id))
21              {
22                  return this;
23              }
24              else
25              {
26                  return _inventory.Fetch(id);
27              }
28          }
29          public override string FullDescription
30          {
31              get
32              {
33                  return $"You are {name} {base.FullDescription}. You are
                      carrying \n" + _inventory.ItemList;
34              }
35          }
36          public Inventory Inventory
37          {
38              get => _inventory;
39          }
40
41      }
42  }
43
```

```
1  using NUnit.Framework;
2  using Iteration2;
3
4  namespace UnitTestIta2
5  {
6      public class InventoryTests
7      {
8          private Inventory inventory;
9          private Item shovel;
10         private Item pc;
11         private Item sword;
12
13         [SetUp]
14         public void Setup()
15         {
16             Item shovel = new Item(new string[] { "shovel"," spade"}, "a
                 shovel", "This is a shovel");
17             Item sword = new Item(new string[] { "sword", " blade" }, "a
                 sword", "This is a sword");
18             Item pc = new Item(new string[] { "pc", " computer" }, "a pc",
                 "This is a computer");
19             inventory = new Inventory();
20             inventory.Put(shovel);
21             inventory.Put(sword);
22             inventory.Put(pc);
23         }
24
25         [Test]
26         public void TestFindItem()
27         {
28             Assert.IsTrue(inventory.HasItem("shovel"));
29             Assert.IsTrue(inventory.HasItem("sword"));
30             Assert.IsTrue(inventory.HasItem("pc"));
31         }
32
33         [Test]
34         public void TestNoFindItem()
35         {
36             Assert.IsFalse(inventory.HasItem("stick"));
37         }
38
39         [Test]
40         public void TestFetchItems()
41         {
42             Assert.NotNull(inventory.Fetch("shovel"));
43             Assert.IsTrue(inventory.HasItem("shovel"));
44         }
45
46         [Test]
```

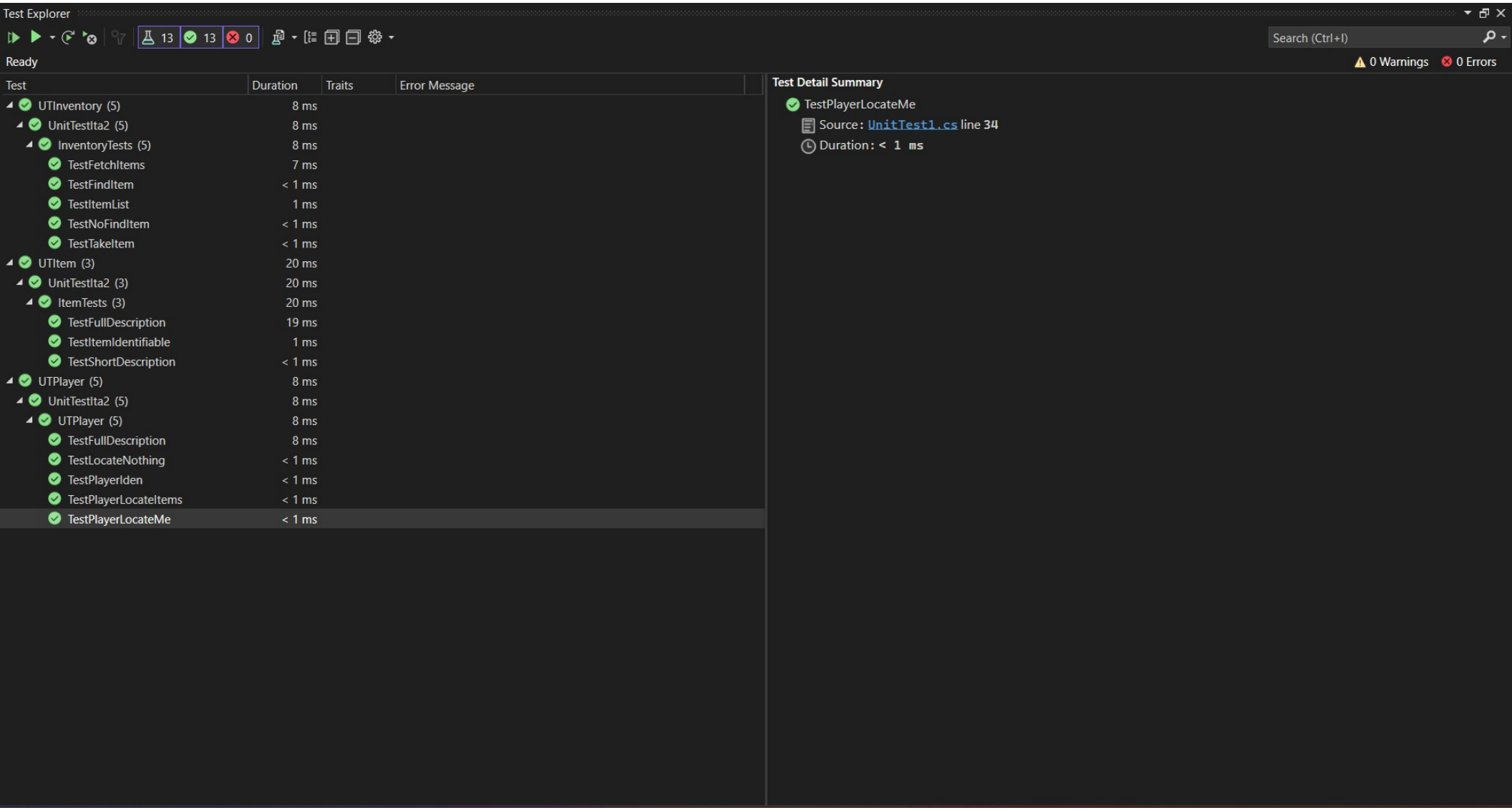
```
47          public void TestTakeItem()
48          {
49              Assert.NotNull(inventory.Take("sword"));
50              Assert.IsFalse(inventory.HasItem("sword"));
51          }
52
53          [Test]
54          public void TestItemList()
55          {
56              string ItemList = "a shovel (shovel)\n a sword (sword)\na small ↵
                   computer(pc)";
57              Assert.That(ItemList, Is.EqualTo(ItemList));
58          }
59
60      }
61  }
62
```

```
1  using NUnit.Framework;
2  using Iteration2;
3
4  namespace UnitTestIta2
5  {
6      public class ItemTests
7      {
8          Item shovel = new Item(new string[] { "shovel" }, "a shovel", "This
               is a shovel");
9          Item sword = new Item(new string[] { "sword" }, "a sword", "This is
               a sword");
10
11
12         [SetUp]
13         public void Setup()
14         {
15
16         }
17
18         [Test]
19         public void TestItemIdentifiable()
20         {
21             var result = shovel.AreYou("sword");
22             Assert.IsFalse(result); //Item cannot be defined
23
24             var result2 = shovel.AreYou("shovel");
25             Assert.IsTrue(result2);
26
27         }
28
29         [Test]
30         public void TestShortDescription()
31         {
32             Assert.AreEqual(shovel.ShortDescription, "a shovel
                 (shovel)"); //Description Correct!
33             Assert.AreNotEqual(shovel.ShortDescription, "This is a
                 shovel"); //Testing short with long Description showing they
                 are not Correct!
34         }
35
36         [Test]
37         public void TestFullDescription()
38         {
39             Assert.AreEqual(shovel.FullDescription, "This is a shovel"); //
                 Full Description is Correct!
40             Assert.AreNotEqual(shovel.FullDescription, "a shovel
                 (shovel)"); //Full Description is not Correct!
41
42         }
```

```
43      }
44  }
```

```
1  using NUnit.Framework;
2  using Iteration2;
3  using NuGet.Frameworks;
4
5  namespace UnitTestIta2
6  {
7      public class UTPlayer
8      {
9          private Player player;
10         [SetUp]
11         public void Setup()
12         {
13             player = new Player("fred", "the mighty programmer");
14             player.Inventory.Put(new Item(new string[] { "shovel",
                 "spade" }, " a shovel", "This is a fine shovel"));
15             player.Inventory.Put(new Item(new string[] { "sword",
                 "blade" }, " a sword", "This is a fine sword"));
16             player.Inventory.Put(new Item(new string[] { "pc",
                 "computer" }, " a computer", "This is a fine computer"));
17         }
18
19         [Test]
20         public void TestPlayerIden()
21         {
22             Assert.IsTrue(player.AreYou("me"));
23             Assert.IsTrue(player.AreYou("inventory"));
24         }
25
26         [Test]
27         public void TestPlayerLocateItems()
28         {
29             Assert.IsTrue(player.Locate("shovel").AreYou("shovel"));
30             Assert.IsTrue(player.Locate("sword").AreYou("sword"));
31             Assert.IsTrue(player.Locate("pc").AreYou("pc"));
32         }
33         [Test]
34         public void TestPlayerLocateMe()
35         {
36             Assert.IsTrue(player.Locate("me").AreYou("me"));
37             Assert.IsTrue(player.Locate("inventory").AreYou("inventory"));
38         }
39         [Test]
40         public void TestLocateNothing()
41         {
42             Assert.IsNull(player.Locate("stick"));
43         }
44         [Test]
45         public void TestFullDescription()
46         {
```

```
47              string pFullDescription = "You are fred the mighty programmer.
                  You are carrying \n a shovel (shovel)\n a sword (sword)\n a
                  computer (pc)\n";
48              Assert.That(player.FullDescription,Is.EqualTo
                  (pFullDescription));
49          }
50      }
51  }
```


Test Explorer
13
13
0
Search (Ctrl+I)
Ready
0 Warnings
0 Errors
Test
Duration
Traits
Error Message
UTInventory (5) 8 ms
UnitTestIta2 (5) 8 ms
InventoryTests (5) 8 ms
TestFetchItems 7 ms
TestFindItem < 1 ms
TestItemList 1 ms
TestNoFindItem < 1 ms
TestTakeItem < 1 ms
UTItem (3) 20 ms
UnitTestIta2 (3) 20 ms
ItemTests (3) 20 ms
TestFullDescription 19 ms
TestItemIdentifiable 1 ms
TestShortDescription < 1 ms
UTPlayer (5) 8 ms
UnitTestIta2 (5) 8 ms
UTPlayer (5) 8 ms
TestFullDescription 8 ms
TestLocateNothing < 1 ms
TestPlayerIden < 1 ms
TestPlayerLocateItems < 1 ms
TestPlayerLocateMe < 1 ms
Test Detail Summary
TestPlayerLocateMe
Source: UnitTest1.cs line 34
Duration: < 1 ms