```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public abstract class GameObject:IdentifiableObject
10     {
11         private string _description;
12         private string _name;
13
14         public GameObject(string[] ids, string name, string desc) : base
             (ids)
15         {
16             _name = name;
17             _description = desc;
18         }
19
20         public string Name
21         {
22             get
23             {
24                 return _name;
25             }
26         }
27
28         public string ShortDescription
29         {
30             get
31             {
32                 return $"{_name} ({FirstId})";
33             }
34         }
35
36         public virtual string FullDescription
37         {
38             get
39             {
40                 return _description;
41             }
42         }
43     }
44 }
45
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class IdentifiableObject
10     {
11         private List<string> _identifiers;
12
13         public IdentifiableObject(string[] idents)
14         {
15             _identifiers = new List<string>();
16             for(int i=0; i<idents.Length; i++)
17             {
18                 _identifiers.Add(idents[i].ToLower());
19             }
20         }
21
22         public bool AreYou(string id)
23         {
24             if(_identifiers.Contains(id.ToLower()))
25             {
26                 return true;
27             }
28             else
29             {
30                 return false;
31             }
32         }
33
34         public string FirstId
35         {
36             get
37             {
38                 if(_identifiers.Count==0)
39                 {
40                     return "";
41                 }
42                 else
43                 {
44                     return _identifiers.First();
45                 }
46             }
47         }
48
49         public void AddIdentifier(string id)
```

```
50          {
51              _identifiers.Add(id.ToLower());
52          }
53
54          public void PrivilegeEscalation(string pin)
55          {
56              if(pin == "6473")
57              {
58                  _identifiers[0] = "6473";
59              }
60          }
61      }
62  }
63
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class Inventory
10     {
11         private 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 take_item = this.Fetch(id);
38             _items.Remove(take_item);
39             return take_item;
40         }
41
42         public Item Fetch(string id)
43         {
44             foreach (Item i in _items)
45             {
46                 if(i.AreYou(id))
47                 {
48                     return i;
49                 }
```

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

```
1
2  namespace IdentifiableObject
3  {
4      public class Tests
5      {
6          private Item _shield;
7          private Item _shovel;
8          private Inventory _my_invent;
9
10         [SetUp]
11         public void Setup()
12         {
13             _shield = new Item(new string[] { "shield" }, "a shield",
                 "Shield level 1");
14             _shovel = new Item(new string[] { "shovel" }, "a shovel",
                 "Shovel level 2");
15             _my_invent = new Inventory();
16         }
17
18         [Test]
19         public void TestFindItem()
20         {
21             _my_invent.Put(_shield);
22             Assert.IsTrue(_my_invent.HasItem(_shield.FirstId));
23         }
24
25         [Test]
26         public void TestNoItemFind()
27         {
28             Assert.IsFalse(_my_invent.HasItem(_shield.FirstId));
29         }
30
31         [Test]
32         public void TestFetchItem()
33         {
34             _my_invent.Put(_shovel);
35             Item fetch_item = _my_invent.Fetch(_shovel.FirstId);
36
37             Assert.AreEqual(fetch_item, _shovel);
38             Assert.IsTrue(_my_invent.HasItem(_shovel.FirstId));
39         }
40
41         [Test]
42         public void TestTakeItem()
43         {
44             _my_invent.Put(_shovel);
45             _my_invent.Take(_shovel.FirstId);
46
47             Assert.IsFalse(_my_invent.HasItem(_shovel.FirstId));
```

```
48          }
49
50          [Test]
51          public void TestItemList()
52          {
53              _my_invent.Put(_shield);
54              _my_invent.Put(_shovel);
55
56              Assert.AreEqual(_my_invent.ItemList, "a shield (shield)\n" + "a ➥
                   shovel (shovel)\n");
57          }
58      }
59  }
```

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

```
1
2  namespace IdentifiableObject
3  {
4      public class Tests
5      {
6          private Item _shield;
7          private Item _shovel;
8
9          [SetUp]
10         public void Setup()
11         {
12             _shield = new Item(new string[] { "shield" }, "a shield",
                 "Shield level 1");
13             _shovel = new Item(new string[] { "shovel" }, "a shovel",
                 "Shovel level 2");
14         }
15
16         [Test]
17         public void TestItemIdentifiable()
18         {
19             Assert.IsTrue(_shield.AreYou("shield"));
20         }
21
22         [Test]
23         public void TestShortDescription()
24         {
25             Assert.AreEqual(_shield.ShortDescription, "a shield (shield)");
26         }
27
28         [Test]
29         public void TestFullDescription()
30         {
31             Assert.AreEqual(_shovel.FullDescription, "Shovel level 2");
32         }
33
34         [Test]
35         public void TestPrivilegeEscalarion()
36         {
37             _shield.PrivilegeEscalation("6473");
38             Assert.AreEqual(_shield.FirstId, "6473");
39         }
40     }
41 }
```

```
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 IdentifiableObject
9  {
10     public class Player:GameObject
11     {
12         private Inventory _inventory;
13
14         public Player(string name, string desc) : base(new string[] {"me",
             "inventory"}, name, desc)
15         {
16             _inventory = new Inventory();
17         }
18
19         public GameObject Locate(string id)
20         {
21             if (AreYou(id) == true)
22             {
23                 return this;
24             }
25             else
26             {
27                 return _inventory.Fetch(id);
28             }
29
30         }
31
32         public override string FullDescription
33         {
34             get
35             {
36                 return $"You are ({Name}), ({base.FullDescription}). You
                     are carrying:\n{_inventory.ItemList}";
37             }
38         }
39
40         public Inventory Inventory
41         {
42             get
43             {
44                 return _inventory;
45             }
46         }
47     }
```

```
48  }
49
```



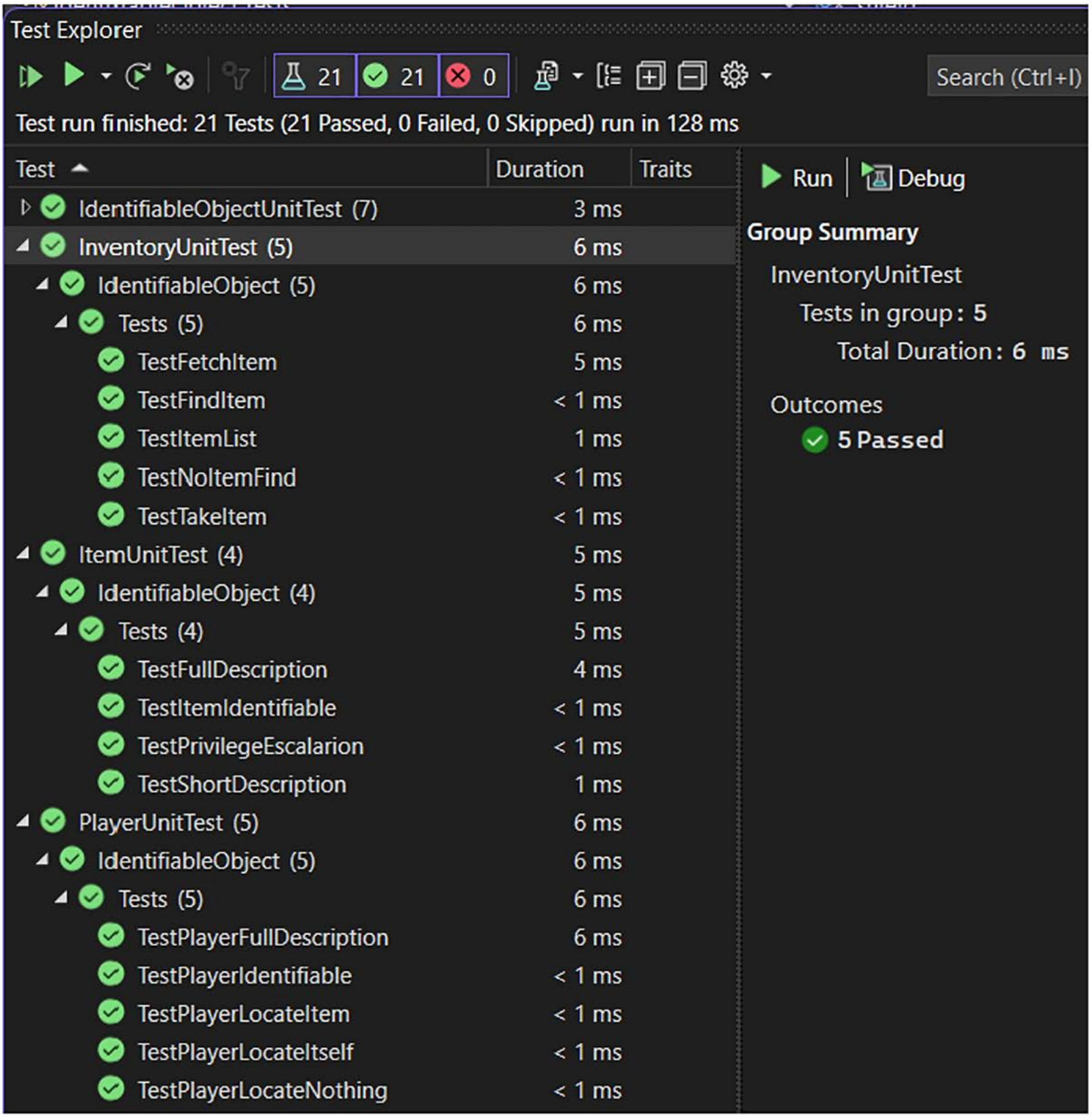
```
1
2   namespace IdentifiableObject
3   {
4       public class Tests
5       {
6           private Item _shield;
7           private Item _shovel;
8           private Player _player;
9
10          [SetUp]
11          public void Setup()
12          {
13              _shield = new Item(new string[] { "shield" }, "a shield",
                  "Shield level 1");
14              _shovel = new Item(new string[] { "shovel" }, "a shovel",
                  "Shovel level 2");
15              _player = new Player("Duc Thang", "Student");
16          }
17
18          [Test]
19          public void TestPlayerIdentifiable()
20          {
21              Assert.IsTrue(_player.AreYou("me") && _player.AreYou
                  ("inventory"));
22          }
23
24          [Test]
25          public void TestPlayerLocateItem()
26          {
27              bool test = false;
28              _player.Inventory.Put(_shield);
29              var located_itm = _player.Locate("shield");
30              if(located_itm == _shield)
31              {
32                  test = true;
33              }
34              Assert.IsTrue(test);
35          }
36
37          [Test]
38          public void TestPlayerLocateItself()
39          {
40              bool test = false;
41              var myself = _player.Locate("me");
42              var invent = _player.Locate("inventory");
43              if (myself == _player || invent==_player)
44              {
45                  test = true;
46              }
```

...4\Task4_2P\Swin_Adventure\PlayerUnitTest\UnitTest1.cs 2

```
47              Assert.IsTrue(test);
48          }
49
50          [Test]
51          public void TestPlayerLocateNothing()
52          {
53              Assert.IsNull(_player.Locate("shield"));
54          }
55
56          [Test]
57          public void TestPlayerFullDescription()
58          {
59              _player.Inventory.Put(_shield);
60              _player.Inventory.Put(_shovel);
61              string output = "You are (Duc Thang), (Student). You are
                  carrying:\n" + "a shield (shield)\n" + "a shovel (shovel)\n";
62              Assert.AreEqual(_player.FullDescription, output);
63          }
64      }
65  }
```

...Task4_2P\Swin_Adventure\IdentifiableObject\Program.cs 1

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


Test Explorer
21
21
0
Search (Ctrl+I)
Test run finished: 21 Tests (21 Passed, 0 Failed, 0 Skipped) run in 128 ms
Test
Duration
Traits
IdentifiableObjectUnitTest (7) 3 ms
InventoryUnitTest (5) 6 ms
IdentifiableObject (5) 6 ms
Tests (5) 6 ms
TestFetchItem 5 ms
TestFindItem < 1 ms
TestItemList 1 ms
TestNoItemFind < 1 ms
TestTakeItem < 1 ms
ItemUnitTest (4) 5 ms
IdentifiableObject (4) 5 ms
Tests (4) 5 ms
TestFullDescription 4 ms
TestItemIdentifiable < 1 ms
TestPrivilegeEscalarion < 1 ms
TestShortDescription 1 ms
PlayerUnitTest (5) 6 ms
IdentifiableObject (5) 6 ms
Tests (5) 6 ms
TestPlayerFullDescription 6 ms
TestPlayerIdentifiable < 1 ms
TestPlayerLocateItem < 1 ms
TestPlayerLocateItself < 1 ms
TestPlayerLocateNothing < 1 ms
Run
Debug
Group Summary
InventoryUnitTest
Tests in group: 5
Total Duration: 6 ms
Outcomes
5 Passed