# Player.cs



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

```
49          public Location CurrentLocation
50          {
51              get => _location;
52              set => _location = value;
53          }
54
55          public Inventory Inventory
56          {
57              get => _inventory;
58          }
59      }
60  }
61
```

# Location.cs



```
1  namespace Iteration1
2  {
3      public class Location : GameObject, IHaveInventory
4      {
5          private Inventory _inventory;
6
7          public Location(string[] ids, string name, string desc) : base(ids, ↵
               name, desc)
8          {
9              _inventory = new Inventory();
10         }
11
12         public GameObject Locate(string id)
13         {
14             if (AreYou(id))
15             {
16                 return this;
17             }
18             else if (_inventory.HasItem(id))
19             {
20                 return _inventory.Fetch(id);
21             }
22             else
23             {
24                 return null;
25             }
26         }
27
28         public Inventory Inventory
29         {
30             get => _inventory;
31         }
32     }
33 }
34
```

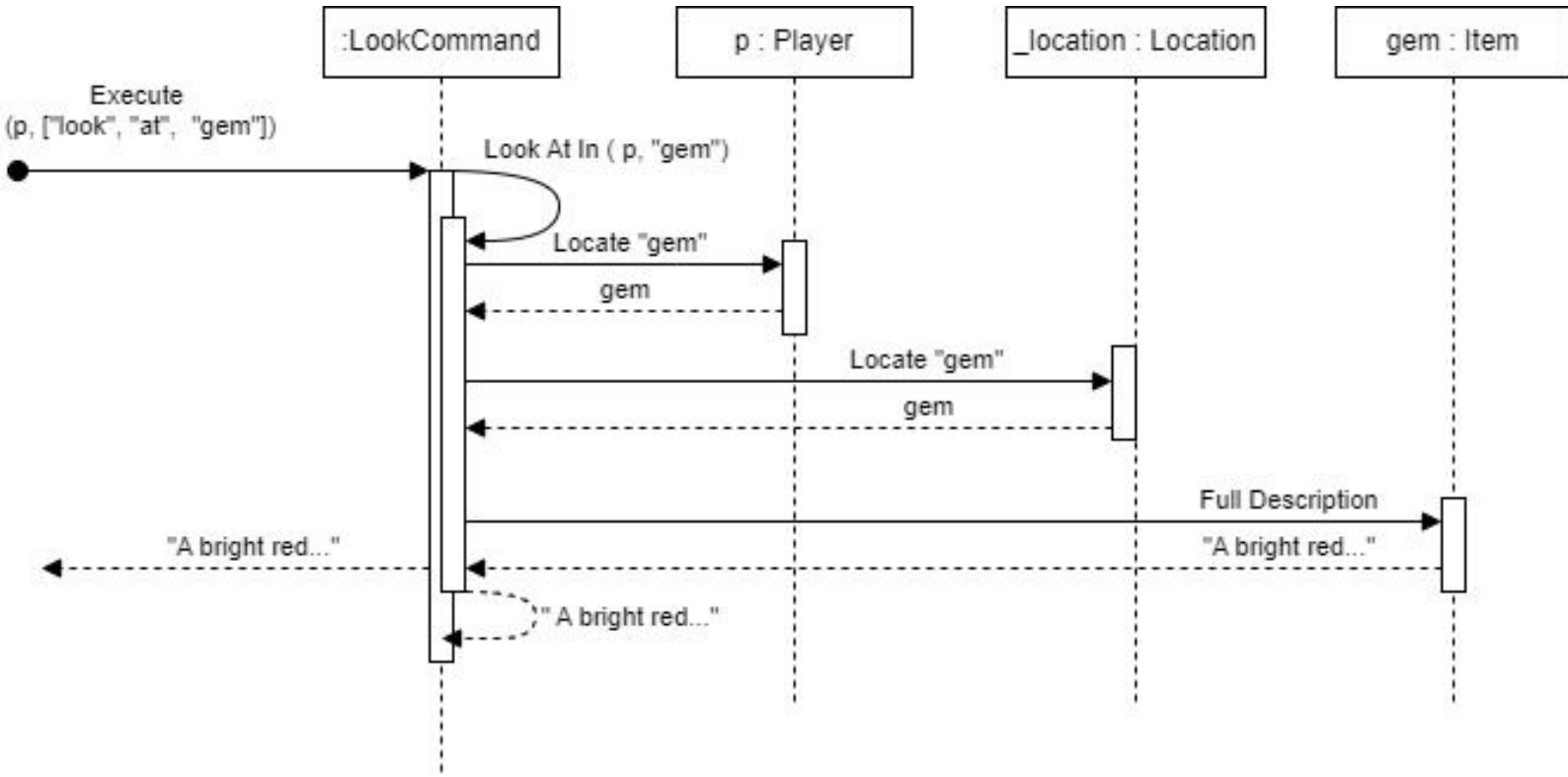
# TestLocations.cs



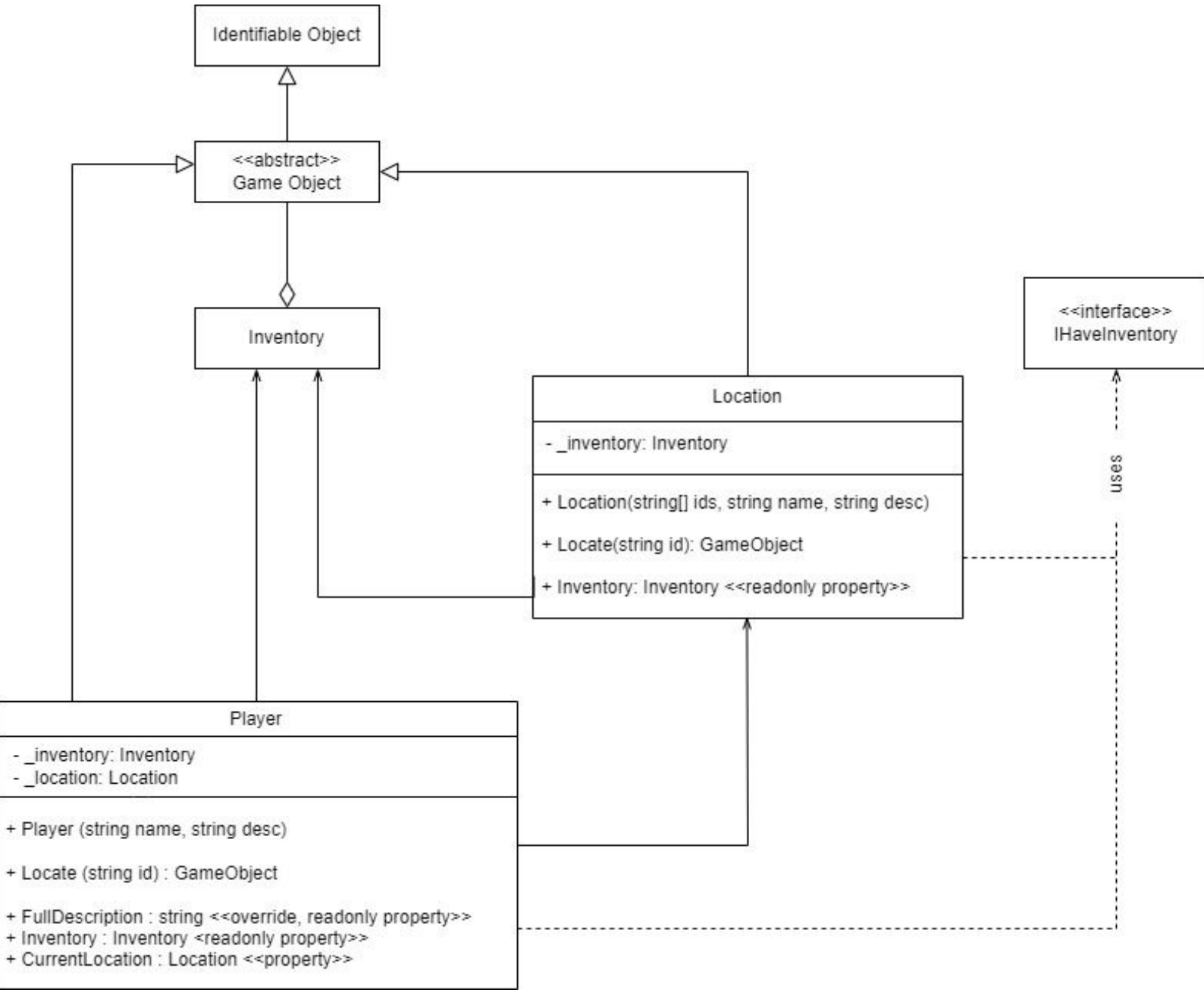
```
1  using Iteration1;
2
3  namespace TestLocations
4  {
5      public class Tests
6      {
7          Player _player;
8          Location _location;
9          Item _sword;
10         Item _ak47;
11         Item _grenade;
12
13         [SetUp]
14         public void Setup()
15         {
16             _player = new Player("Chien", "A boy with high curiosity");
17             _location = new Location(new string[] { "military base" },
                 "military base", "large area");
18             _sword = new Item(new string[] { "sword", "melee" }, "sword",
                 "Short range weapon");
19             _ak47 = new Item(new string[] { "ak47" }, "ak47", "Long range
                 weapon");
20             _grenade = new Item(new string[] { "grenade" }, "grenade",
                 "Very high damage weapon!");
21             _location.Inventory.Put(_sword);
22             _location.Inventory.Put(_ak47);
23             _player.CurrentLocation = _location;
24         }
25
26         [Test]
27         public void TestLocationLocatesItself()
28         {
29             Assert.That(_location.Locate("military base"), Is.EqualTo
                 (_location));
30             Assert.Pass();
31         }
32
33         [Test]
34         public void TestLocationLocatesItem()
35         {
36             Assert.That(_location.Locate("sword"), Is.EqualTo(_sword));
37             Assert.Pass();
38         }
39
40         [Test]
41         public void TestLocationLocatesNothing()
42         {
43             Assert.That(_location.Locate("grenade"), Is.EqualTo(null));
44             Assert.Pass();
```

```
45          }
46
47          [Test]
48          public void TestPLayerLocatesItemInLocation()
49          {
50              Assert.That(_player.Locate("ak47"), Is.EqualTo(_ak47));
51              Assert.Pass();
52          }
53
54          [Test]
55          public void TestPlayerLocatesNothingInLocation()
56          {
57              Assert.That(_player.Locate("grenade"), Is.EqualTo(null));
58              Assert.Pass();
59          }
60      }
61  }
```

# Sequence Diagram



# UML Diagram


Identifiable Object
<<abstract>>
Game Object
Inventory
<<interface>>
IHaveInventory
Location
- _inventory: Inventory
+ Location(string[] ids, string name, string desc)
+ Locate(string id): GameObject
+ Inventory: Inventory <<readonly property>>
uses
Player
- _inventory: Inventory
- _location: Location
+ Player (string name, string desc)
+ Locate (string id) : GameObject
+ FullDescription : string <<override, readonly property>>
+ Inventory : Inventory <readonly property>>
+ CurrentLocation : Location <<property>>

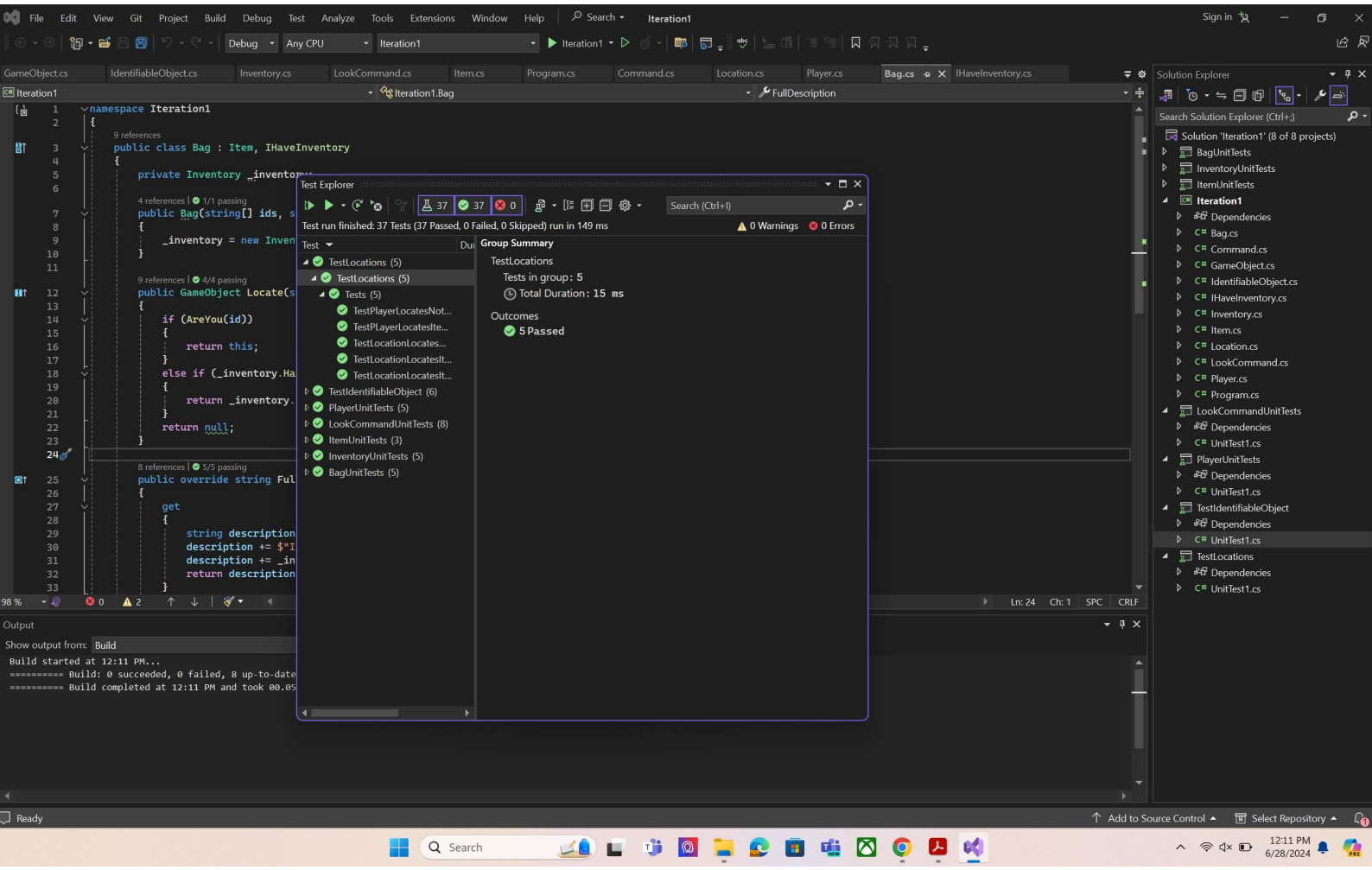
# Program.cs



```
1  using System.Runtime.InteropServices;
2
3  namespace Iteration1
4  {
5      internal class Program
6      {
7          static void Main(string[] args)
8          {
9              string name;
10             string desc_;
11             string command;
12             Location location;
13             Item redbull;
14             Item sword;
15             Item ak47;
16             Item grenade;
17
18             Console.Write("Player name: ");
19             name = Console.ReadLine();
20             Console.Write("Player description: ");
21             desc_ = Console.ReadLine();
22             Player player = new Player(name, desc_);
23
24             location = new Location(new string[] { "hospital" },
                 "hospital", "This is a state-of-the-art hospital");
25             redbull = new Item(new string[] { "Redbull" }, "Redbull",
                 "Drink to be more energetic!");
26             location.Inventory.Put(redbull);
27             player.CurrentLocation = location;
28
29             sword = new Item(new string[] { "sword" }, "sword", "Short
                 range weapon!");
30             ak47 = new Item(new string[] { "ak47" }, "ak47", "Average range
                  weapon with high damage!");
31
32             player.Inventory.Put(sword);
33             player.Inventory.Put(ak47);
34
35             grenade = new Item(new string[] { "grenade" }, "grenade",
                 "Extreme damage and short range weapon!");
36             Bag bag1 = new Bag(new string[] {"bag1"}, "bag1", "");
37             bag1.Inventory.Put(grenade);
38             player.Inventory.Put(bag1);
39
40             LookCommand player_command = new LookCommand();
41             while (true)
42             {
43                 Console.Write("Command -> ");
44                 command = Console.ReadLine();
```

```
45                  string message = player_command.Execute(player, new string
                      [] { command });
46                  Console.WriteLine(message);
47              }
48          }
49      }
50  }
```

# Screenshot of Test Passing



# Output Screenshot