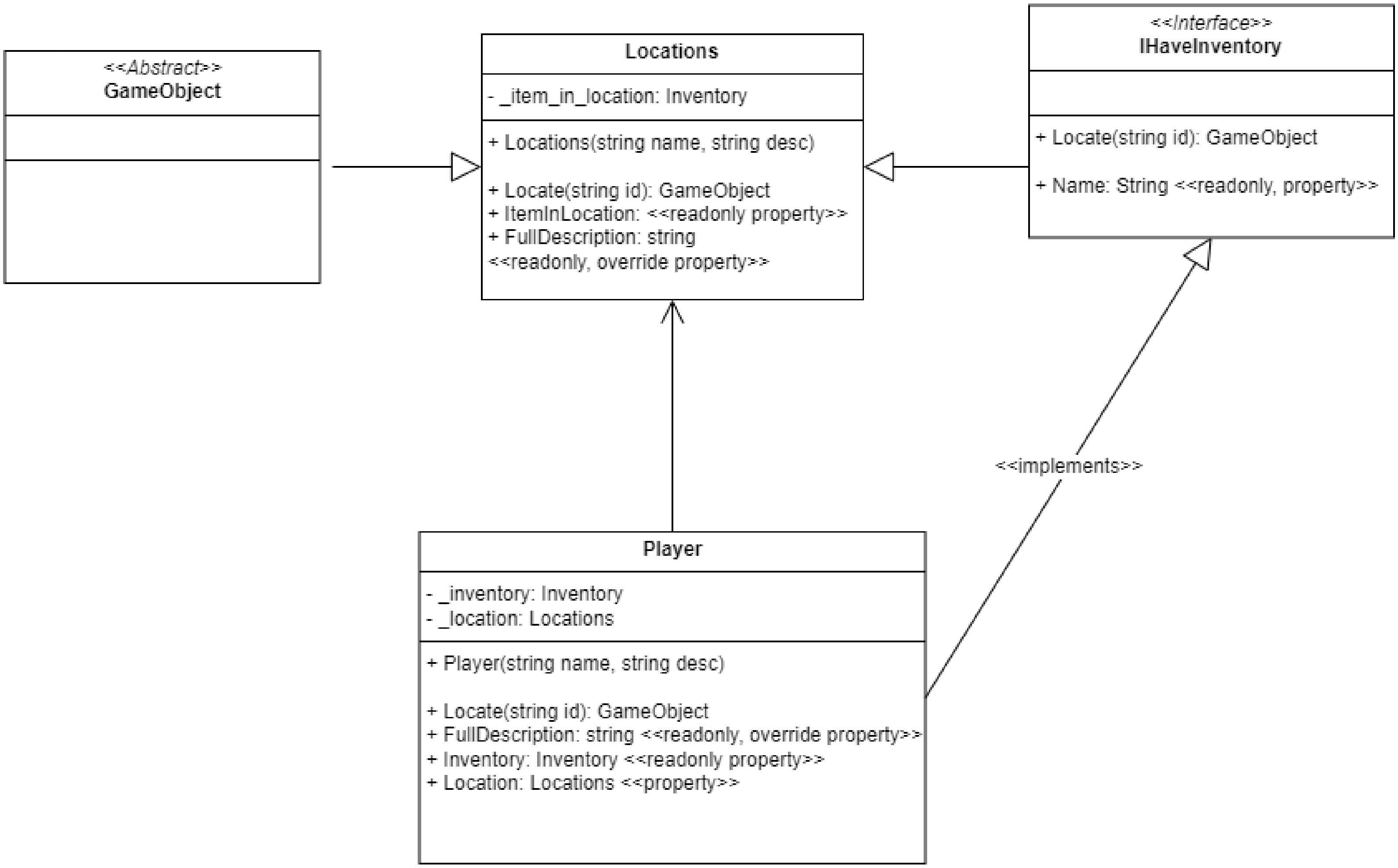
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IdentifiableObject
{
    public interface IHaveInventory
    {
        public GameObject Locate(string id);

        public string Name
        {
            get;
        }
    }
}

```

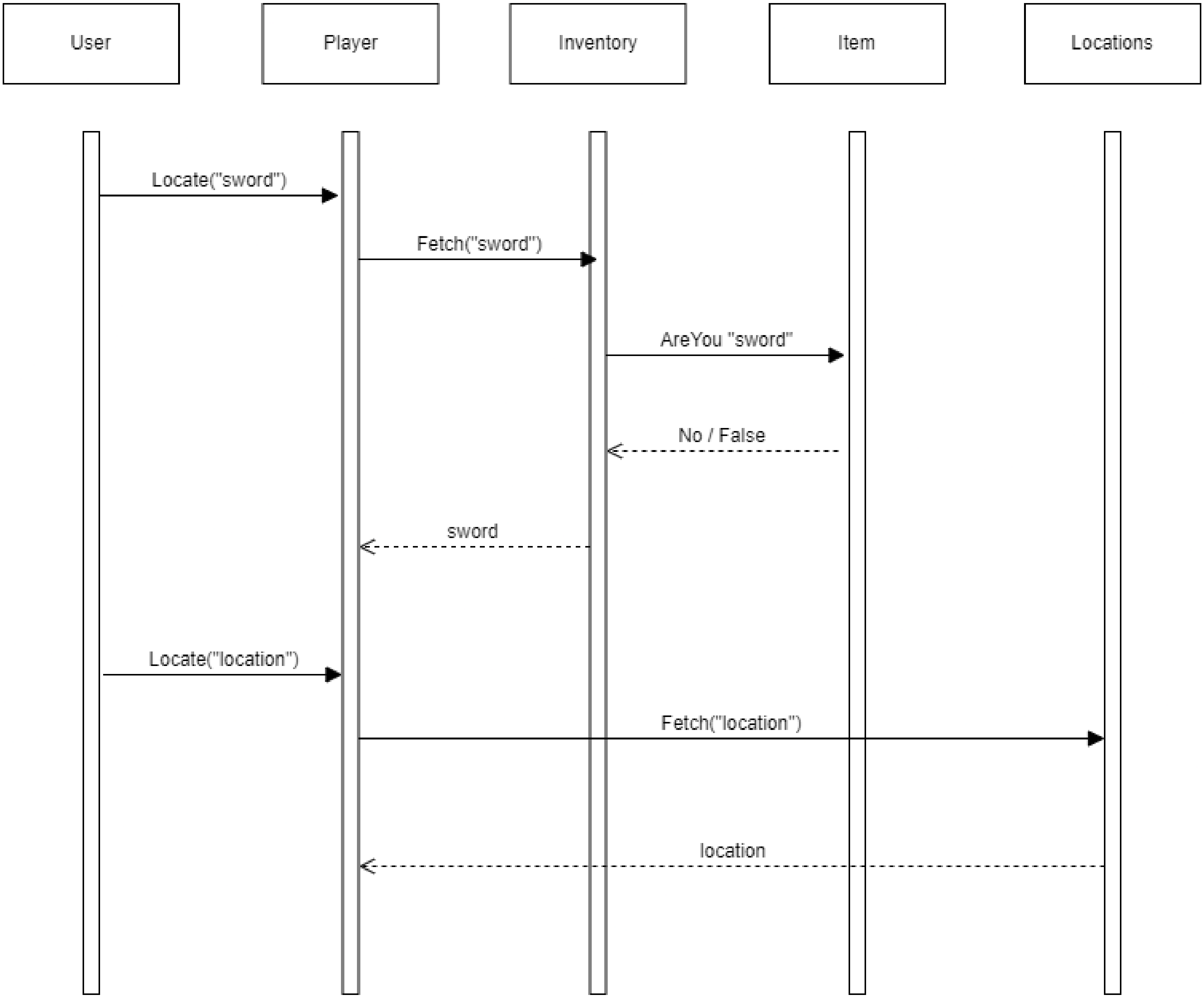
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IdentifiableObject
{
    public class Locations:GameObject, IHaveInventory
    {
        private Inventory _item_in_location;

        public Locations(string name, string desc):base(new string[] {"location"}, name, desc)
        {
            _item_in_location = new Inventory();

        }

        public GameObject Locate(string id)
        {
            if(AreYou(id))
            {
                return this;
            }
            else
            {
                return _item_in_location.Fetch(id);
            }
        }

        public Inventory ItemInLocation
        {
            get
            {
                return _item_in_location;
            }
        }

        public override string FullDescription
        {
            get
            {
                return $"Welcome, {base.FullDescription}\nIn this location you can see:\n{ItemInLocation.ItemList} ";
            }
        }
    }
}
```

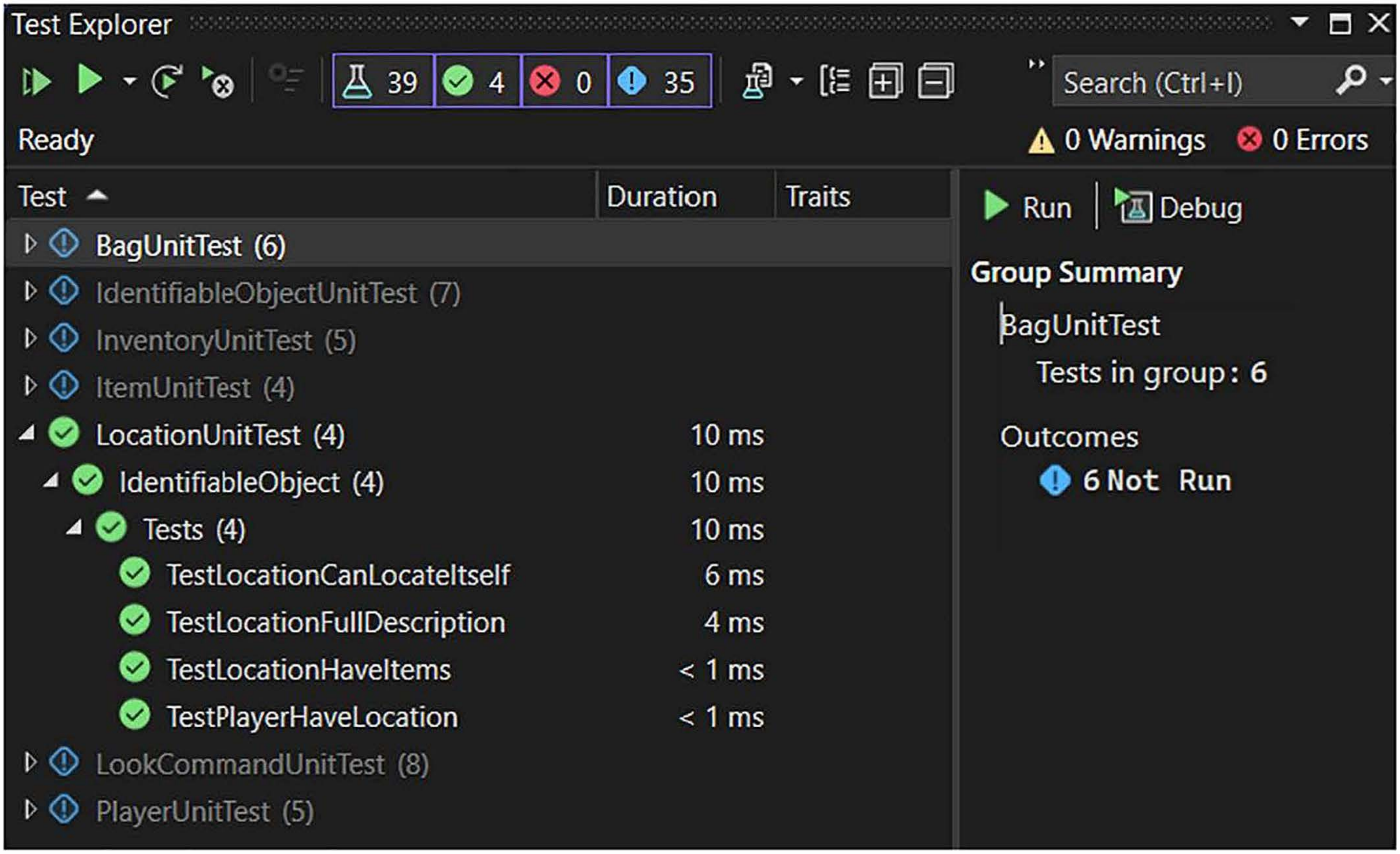
```
namespace IdentifiableObject
{
    public class Tests
    {
        private Player _player;
        private Item _wand;
        private Item _scaff;
        private Locations _store;
        private Command _command;

        [SetUp]
        public void Setup()
        {
            _player = new Player("Thang", "104776473");
            _wand = new Item(new string[] { "wand" }, "a wand", "wand level 5");
            _scaff = new Item(new string[] { "scaff" }, "a scaff", "a blue scaff");
            _store = new Locations("store", "This is an item store");
            _command = new LookCommand();

        }

        [Test]
        public void TestLocationCanLocateItself()
        {
            Assert.IsTrue(_store.AreYou("location"));
        }

        [Test]
        public void TestLocationHaveItems()
        {
            _store.ItemInLocation.Put(_wand);
            _store.ItemInLocation.Put(_scaff);

            Assert.AreEqual(_wand, _store.Locate("wand"));
        }

        [Test]
        public void TestPlayerHaveLocation()
        {
            Assert.AreEqual(_player.Locate("location"), _player.Location);
        }

        [Test]
        public void TestLocationFullDescription()
        {
            _player.Location = _store;
            _store.ItemInLocation.Put(_wand);
            _store.ItemInLocation.Put(_scaff);
            Assert.AreEqual(_command.Execute(_player, new string[] {"look", "at", "location"}), $"{_store.FullDescription}");
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;

namespace IdentifiableObject
{
    public class Player:GameObject, IHaveInventory
    {
        private Inventory _inventory;
        private Locations _location;

        public Player(string name, string desc) : base(new string[] {"me", "inventory"}, name, desc)
        {
            _inventory = new Inventory();
        }

        public GameObject Locate(string id)
        {
            if (AreYou(id)==true)
            {
                return this;
            }
            else if(_inventory.Fetch(id)!=null)
            {
                return _inventory.Fetch(id);
            }
            else if (_location!=null)
            {
                return _location.Locate(id);
            }
            else
            {
                return null;
            }

        }

        public override string FullDescription
        {
            get
            {
                return $"You are ({Name}), ({base.FullDescription}). You are carrying:\n{_inventory.ItemList}";
            }
        }

        public Inventory Inventory
        {
            get
            {
                return _inventory;
            }
        }

        public Locations Location
        {
            get
            {
                return _location;
            }
            set
            {
                _location = value;
            }
        }
    }
}

```

```csharp
using IdentifiableObject;
using System;

class Program
{
    static void Main(string[] args)
    {
        string player_name, player_desc;
        Player player;
        Bag player_bag=new Bag(new string[]{"bag"}, "Thang's bag", "bag number 104776473");
        Command command = new LookCommand();
        Locations guild = new Locations("Guild", "This is the city's guild");

        Console.Write("Please enter your name: ");
        player_name = Console.ReadLine();
        Console.Write("Please enter your description: ");
        player_desc = Console.ReadLine();

        player = new Player(player_name, player_desc);

        Item shield = new Item(new string[] { "shield" }, "a shield", "Shield level 1");
        Item shovel = new Item(new string[] { "shovel" }, "a shovel", "Shovel level 2");

        player.Inventory.Put(shield);
        player.Inventory.Put(shovel);

        player.Inventory.Put(player_bag);

        Item blanket = new Item(new string[] { "blanket" }, "a blanket", "winter blanket");
        player_bag.Inventory.Put(blanket);

        player.Location = guild;

        Item sword = new Item(new string[] { "sword" }, "a sword", "Sword level 10");
        Item staff = new Item(new string[] { "staff" }, "a staff", "Staff level 20");
        guild.ItemInLocation.Put(sword);
        guild.ItemInLocation.Put(staff);

        while (true)
        {
            Console.Write("Orders: ");
            string user_input = Console.ReadLine();

            Console.WriteLine(command.Execute(player, user_input.Split()));

        }
    }
}
```


<<Abstract>>
GameObject
Locations
- _item_in_location: Inventory
+ Locations(string name, string desc)
+ Locate(string id): GameObject
+ ItemInLocation: <<readonly property>>
+ FullDescription: string
<<readonly, override property>>
<<Interface>>
IHaveInventory
+ Locate(string id): GameObject
+ Name: String <<readonly, property>>
<<implements>>
Player
- _inventory: Inventory
- _location: Locations
+ Player(string name, string desc)
+ Locate(string id): GameObject
+ FullDescription: string <<readonly, override property>>
+ Inventory: Inventory <<readonly property>>
+ Location: Locations <<property>>


User
Player
Inventory
Item
Locations
Locate("sword")
Fetch("sword")
AreYou "sword"
No / False
sword
Locate("location")
Fetch("location")
location



Ready

⚠ 0 Warnings ⊗ 0 Errors

Test	Duration	Traits
BagUnitTest (6)		
IdentifiableObjectUnitTest (7)		
InventoryUnitTest (5)		
ItemUnitTest (4)		
LocationUnitTest (4)	10 ms	
IdentifiableObject (4)	10 ms	
Tests (4)	10 ms	
TestLocationCanLocateItself	6 ms	
TestLocationFullDescription	4 ms	
TestLocationHaveItems	< 1 ms	
TestPlayerHaveLocation	< 1 ms	
LookCommandUnitTest (8)		
PlayerUnitTest (5)		

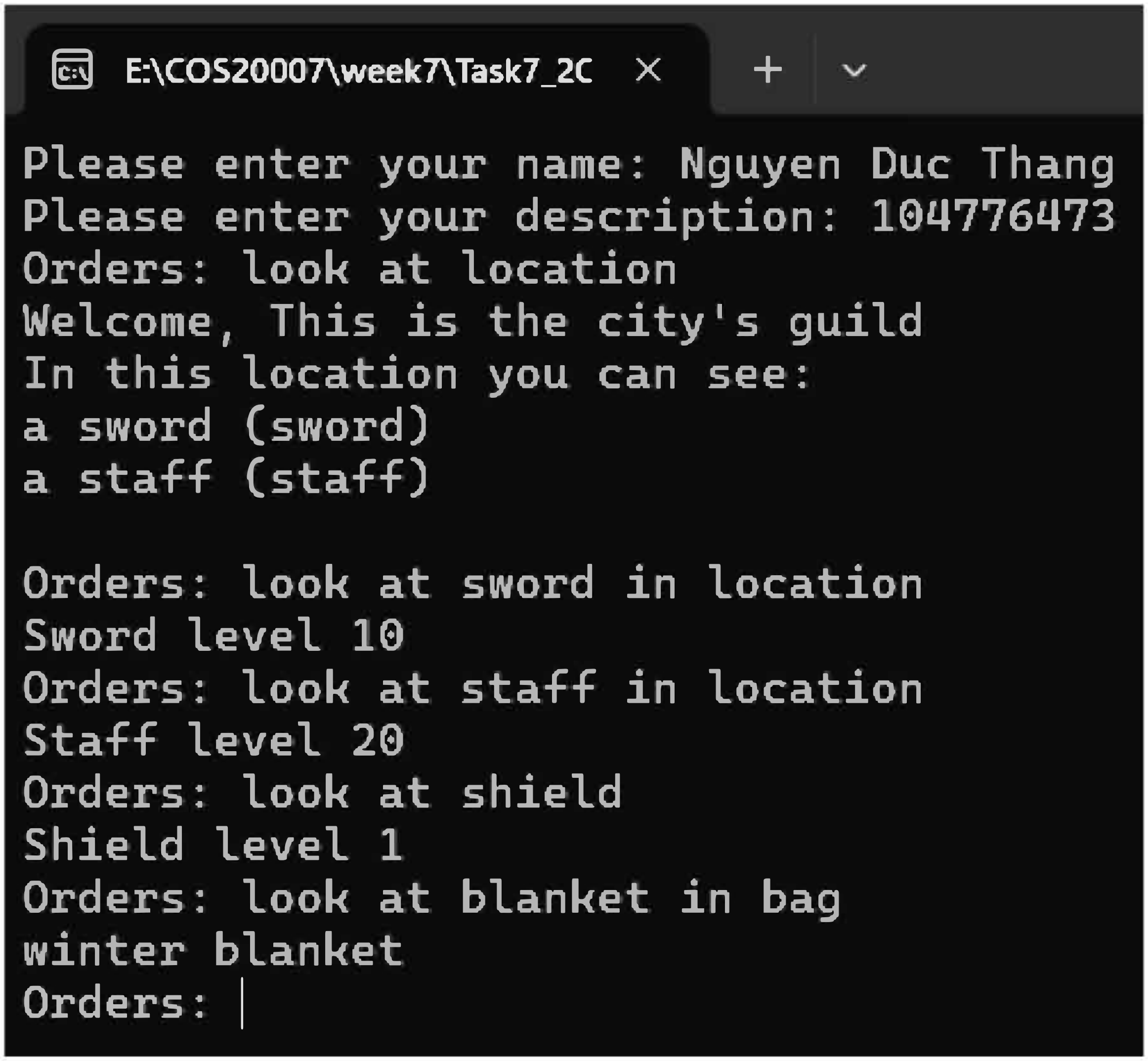
Group Summary

BagUnitTest

Tests in group: 6

Outcomes

6 Not Run



```
Please enter your name: Nguyen Duc Thang
Please enter your description: 104776473
Orders: look at location
Welcome, This is the city's guild
In this location you can see:
a sword (sword)
a staff (staff)

Orders: look at sword in location
Sword level 10
Orders: look at staff in location
Staff level 20
Orders: look at shield
Shield level 1
Orders: look at blanket in bag
winter blanket
Orders:
```