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

namespace SwinAdventureTest
{
    public class LookCommandtest //Change from internal to public
    {
        Item gem;
        Bag bag;
        Player player;
        LookCommand look;
        [SetUp]
        public void SetUp()
        {
            look = new LookCommand();

            player = new Player("Truong Ngoc Gia Hieu", "A brave Swinburne Warrior");
            bag = new Bag(new string[] { "bag" }, "Hieu's Bag", "A small Bag");

            gem = new Item(new string[] { "gem" }, "a gem", "A bright red gem");

            player.Inventory.Put(bag);
        }
        [Test]
        public void TestLookAtGem()
        {
            player.Inventory.Put(gem);

            string output = look.Execute(player, new string[] { "look", "at", "gem" });
            string expected = "A bright red gem";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestLookAtMe()
        {
            string output = look.Execute(player, new string[] { "look", "at", "inventory" });
            string expected = "Truong Ngoc Gia Hieu, A brave Swinburne Warrior\nList of Items that you have:\nHieu's Bag: bag\n";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestLookAtGemInMe()
        {
            player.Inventory.Put(gem);

            string output = look.Execute(player, new string[] { "look", "at", "gem", "in", "inventory" });
            string expected = "A bright red gem";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestLookAtUnk()
        {
            // Test looking at an unknown item
            string output = look.Execute(player, new string[] { "look", "at", "gem" });
            string expected = "I cannot find the gem in the Truong Ngoc Gia Hieu";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestLookatGemInNoBag()
        {
            Player player1 = new Player("NPC", "Robot Warrior");

            string output = look.Execute(player1, new string[] { "look", "at", "gem", "in", "bag" });
            string expected = "I cannot find the gem";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestLookAtGemInBag()
        {
            // Put the gem in the bag
            bag.Inventory.Put(gem);

            // Test looking at the gem in the bag
            string output = look.Execute(player, new string[] { "look", "at", "gem", "in", "bag" });
            string expected = "A bright red gem";
            Assert.AreEqual(expected, output);
        }

        [Test]
        public void TestInvalidLookCommands()
        {
            string output1 = look.Execute(player, new string[] { "look", "around" });
            string output2 = look.Execute(player, new string[] { "hello" });
            string expected = "I don't know how to look like that";

            Assert.AreEqual(expected, output1);
            Assert.AreEqual(expected, output2);
        }

        [Test]
        public void TestLookAtNoGemInBag()
        {
            // Create an empty bag
            Bag emptyBag = new Bag(new string[] { "emptybag" }, "Empty Bag", "An empty bag");

            // Put the empty bag in the player's inventory
            player.Inventory.Put(emptyBag);

            // Test looking at the gem in the empty bag
            string output = look.Execute(player, new string[] { "look", "at", "gem", "in", "emptybag" });
            string expected = "I cannot find the gem in the Empty Bag";
            Assert.AreEqual(expected, output);
        }
    }
}
```