

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

namespace SwinAdventureTest
{
    public class InventoryTest
    {
        [Test]
        public void TestFindItem()
        {
            Inventory inventory = new Inventory();
            Item axe = new Item(new string[] { "axe" }, "a power wooden axe",
                                "+10 ATK Damage points");
            Item sword = new Item(new string[] { "sword" }, "an valuable iron
                                sword", "This is a sword, + 20 ATK Damage points");

            inventory.Put(axe);
            Assert.IsTrue(inventory.HasItem(axe.FirstID));
        }

        [Test]
        public void TestFetchItems()
        {
            Inventory inventory = new Inventory();
            Item axe = new Item(new string[] { "axe" }, "a power wooden axe",
                                "+10 ATK Damage points");
            Item sword = new Item(new string[] { "sword" }, "an valuable iron
                                sword", "This is a sword, + 20 ATK Damage points");

            inventory.Put(axe);
            Item fetchItem = inventory.Fetch(axe.FirstID);

            Assert.IsNotNull(fetchItem);
            Assert.AreEqual(axe, fetchItem);
            Assert.IsTrue(inventory.HasItem(axe.FirstID));
        }

        [Test]
        public void TestNoFindItem()
        {
            Inventory inventory = new Inventory();
            Item axe = new Item(new string[] { "axe" }, "a power wooden axe",
                                "+10 ATK Damage points");
            Item sword = new Item(new string[] { "sword" }, "an valuable iron
```

```
        sword", "This is a sword, + 20 ATK Damage points");

        inventory.Put(axe);
        Assert.IsFalse(inventory.HasItem(sword.FirstID));
    }
    [Test]
    public void TestItemList()
    {
        Inventory inventory = new Inventory();
        Item axe = new Item(new string[] { "axe" }, "a power wooden axe",
            "+10 ATK Damage points");
        Item sword = new Item(new string[] { "sword" }, "a valuable iron
            sword", "This is a sword, + 20 ATK Damage points");

        inventory.Put(sword);
        inventory.Put(axe);

        string itemList = inventory.ItemList;

        string expectedItemList = "a valuable iron sword (sword)\na power
            wooden axe (axe)\n";

        Assert.AreEqual(expectedItemList, itemList);
    }

    [Test]
    public void TestTakenItem()
    {
        Inventory inventory = new Inventory();
        Item axe = new Item(new string[] { "axe" }, "a power wooden axe",
            "+10 ATK Damage points");
        Item sword = new Item(new string[] { "sword" }, "a valuable iron
            sword", "This is a sword, + 20 ATK Damage points");

        inventory.Put(axe);
        Assert.IsTrue(inventory.HasItem(axe.FirstID));

        Item takeItem = inventory.Take(axe.FirstID);

        Assert.AreEqual(axe, takeItem);
        Assert.IsFalse(inventory.HasItem(axe.FirstID));
    }
}
```