```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SwinAdventure;
using System.Xml.Linq;
using System.ComponentModel;
using System.Formats.Tar;


namespace SwinAdventure
{
    public class LookCommand : Command  //Change from internal to public
    {
        //Step 1 of the LookCommand.cs in the UML design
        public LookCommand() : base(new string[] { "look" })
        {

        }
        //Step 2 of the LookCommand.cs in the UML design
        public override string Execute(Player p, string[] text)
        {
            IHaveInventory _container = null;
            string _itemid;
            string _containerid;
            //Check the array text for the length
            if (text.Length != 3 && text.Length != 5)
            {
                return "I don't know how to look like that";
            }
            //If the first word must be “look”, return “Error in look input”
            if (text[0].ToLower() != "look")
            {
                return "Error in look input";
            }
            //The second word must be “at”, otherwise return “What do you want to look at?”
            if (text[1].ToLower() != "at")
            {
                return "What do you want to look at?";
            }
            //If there are 5 elements, then the 4th word must be “in”, otherwise return “What do you want to look in?”
            if (text.Length == 5)
            {
                if (text[3].ToLower() != "in")
                {
                    return "What do you want to look in?";
                }
                _containerid = text[4]; // "backpack"
                _container = FetchContainer(p, _containerid); // This gets the Bag object
            }
            //If there are 3 elements, the container is the player
            if (text.Length == 3)
            {
                _container = p;
            }

            _itemid = text[2]; // "map"
            return LookAtIn(_itemid, _container);
        }
        //Step 3 of the LookCommand.cs in the UML design
        private IHaveInventory FetchContainer(Player p, string containerId)
        {
            if (p.Location != null && p.Location.AreYou(containerId)) // New if_else
            {
                return p.Location;
            }
            if (p.AreYou(containerId)) // "me" or "inventory" should identify the player
            {
                return p;
            }
            // Otherwise, try to locate it in the player's inventory (for bags, etc.)
            return p.Locate(containerId) as IHaveInventory;
        }

        ////Step 4 of the LookCommand.cs in the UML design
        private string LookAtIn(string thingId, IHaveInventory container)
        {
            if (container == null)
            {
                return "I cannot find the " + thingId;
            }
            GameObject item = container.Locate(thingId); // This calls Bag.Locate("map")

            if (item == null)
            {
                return "I cannot find the " + thingId + " in the " + container.Name;
            }
            return item.FullDescription;
        }
    }
}
```