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

namespace SwinAdventure
{
    public class MoveCommand : Command
    {
        public MoveCommand() : base(new string[] { "move", "go", "head", "leave" })
        {

        }
        public override string Execute(Player p, string[] text)
        {
            // Validate the primary command verb
            if (!IsMoveCommand(text[0].ToLower()))
            {
                return "I don't know how to move like that."; // Or "I don't know where to go"
            }

            // Handle "move" (without direction)
            if (text.Length == 1)
            {
                return "Where do you want to go?";
            }

            // Handle correct format "move <direction>"
            if (text.Length == 2)
            {
                string nextLocation = text[1].ToLower();
                return MoveTo(nextLocation, p);
            }

            // Handle any other invalid lengths (e.g., "move north east")
            return "I don't know where to go.";
        }
        private bool IsMoveCommand(string command)
        {
            return command == "move" || command == "go" || command == "head" || command == "leave";
        }
        private string MoveTo(string NewLocation, Player p)
        {
            Paths path = p.Location.FindPath(NewLocation);
```

```csharp

            if (path == null)
            {
                return $"Could not find the {NewLocation}";
            }
            else
            {
                p.Move(path);
                return $"You have moved {path.FirstId} to the {p.Location.Name}...\n{p.Location.FullDescription}\n{p.Location.PathList}";
            }
        }
    }
}

```