```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Runtime.InteropServices;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7  using SwinAdventure;
 8
 9  namespace SwinAdventure
10  {
11      public class Location : GameObject, IHaveInventory
12      {
13          //Attributes
14          private Inventory _inventory;
15          List<Paths> _paths; //New attribute
16          //Constructor and methods
17          public Location(string name, string desc) : base(new string[]
              { "location" }, name, desc)
18          {
19              _inventory = new Inventory();
20              _paths = new List<Paths>(); //New code line
21          }
22          //New constructor
23          public Location(string name, string desc, List<Paths> paths) :
              this(name, desc)
24          {
25              _paths = paths;
26          }
27          public GameObject Locate(string id)
28          {
29              if (AreYou(id))
30                  return this;
31              foreach (var item in _inventory.Items)
32              {
33                  if (item.AreYou(id))
34                      return item;
35              }
36              return null;
37          }
38          public override string FullDescription
39          {
40              get
41              {
42                  return $"{Name}, You are in {base.FullDescription}\nItems
                      that available here:\n{_inventory.ItemList}";
43              }
44          }
45
46          public Inventory Inventory
```

```
47          {
48              get { return _inventory; }
49          }
50          //New methods
51          public Paths FindPath(string id)
52          {
53              foreach (Paths path in  _paths)
54              {
55                  if (path.AreYou(id))
56                  {
57                      return path;
58                  }
59              }
60              return null;
61          }
62          public void AddPath(Paths path)
63          {
64              _paths.Add(path);
65          }
66          public string PathList
67          {
68              get
69              {
70                  if (_paths.Count == 0)
71                  {
72                      return "\nHere are no exits.";
73                  }
74                  else
75                  {
76                      string List = "\nPaths that you can go to:\n";
77                      foreach (Paths path in _paths)
78                      {
79                          List += path.FirstId + "\n";
80                      }
81                      return List;
82                  }
83              }
84          }
85          public bool PathExists(Paths CheckPath)
86          {
87              return _paths.Contains(CheckPath);
88          }
89          public string InventoryDescription
90          {
91              get
92              {
93                  if (_inventory.Items.Count() == 0)
94                  {
95                      return "There's no available item.";
```

```
 96                  }
 97                  return _inventory.ItemList;
 98              }
 99          }
100      }
101  }
102
```