```csharp
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4
5  namespace TG2_RFID
6  {
7      public class Ambient
8      {
9          /// <summary>
10         /// Holds the ambient name.
11         /// </summary>
12         protected string name;
13
14         /// <summary>
15         /// Holds the total thermal load.
16         /// A value that is proportional to the number of people in the room.
17         /// </summary>
18         protected int totalThermalLoad;
19
20         /// <summary>
21         /// Holds a map of people inside of room using their tag's EPC as key.
22         /// </summary>
23         protected Dictionary<string, Cardholder> cardholders;
24
25         /// <summary>
26         /// Holds the local antenna for this ambient.
27         /// </summary>
28         protected Antenna localAntenna;
29
30         /// <summary>
31         /// Constructor given the ambient name.
32         /// </summary>
33         public Ambient(string ambientName)
34         {
35             name = ambientName;
36             cardholders = new Dictionary<string, Cardholder>();
37         }
38
39         /// <summary>
40         /// Setter for the ambient antenna.
41         /// </summary>
42         public void SetAntenna(Antenna antenna)
43         {
44             localAntenna = antenna;
45         }
46
47         /// <summary>
48         /// Getter for the total thermal capacity in this room.
49         /// </summary>
50         public int GetTotalThermalLoad()
51         {
52             return totalThermalLoad;
53         }
54
55         /// <summary>
56         /// Getter for the total number of people inside of the ambient.
```

```
57              /// </summary>
58              public int GetNumberOfCarholdersInside()
59              {
60                  return cardholders.Count;
61              }
62
63              /// <summary>
64              /// Adds a cardholder for this ambient.
65              /// </summary>
66              /// <param name="cardholder">Cardholder.</param>
67              public void AddCardholder(Cardholder cardholder)
68              {
69                  try
70                  {
71                      cardholders.Add(cardholder.GetTagEPC(), cardholder);
72                  }
73                  catch(Exception e)
74                  {
75                      // Handle .NET errors.
76                      Console.WriteLine("Exception : {0}", e.Message);
77                  }
78              }
79
80              /// <summary>
81              /// Removes a cardholder for this ambient.
82              /// If the cardholder is not in this ambient nothing happens.
83              /// </summary>
84              /// <param name="cardholder">Cardholder.</param>
85              public void RemoveCardholder(Cardholder cardholder)
86              {
87                  if (cardholders.ContainsKey(cardholder.GetTagEPC()))
88                  {
89                      cardholders.Remove(cardholder.GetTagEPC());
90                  }
91              }
92
93              /// <summary>
94              /// Getter ambient name
95              /// </summary>
96              public string GetName()
97              {
98                  return name;
99              }
100         }
101     }
102
```