**Add and Remove Hall**

Tokyo Fest is an initiative facilitated by many organizations and they want to celebrate the spirit of the city. As a part of business event a trade fare is being organized. Help the organizing committee to manage the halls details in the exhibition by supporting them to add, remove hall details.

Write a program to add and remove hall details for management.

**[Note :  Strictly adhere to the object-oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned. Create separate classes in separate files.]**

Consider a class named **Hall**with the following member attributes,

|  |  |
| --- | --- |
| **Data Type** | **Attribute** |
| string | \_name |
| string | \_contactNumber |
| double | \_costPerDay |
| string | \_ownerName |

The methods for **getters**, **setters** and **constructors** are given in the template code.  
  
Consider a class named as **HallBO**and define the following methods,

|  |  |  |
| --- | --- | --- |
| **S.No** | **Method Name** | **Description** |
| 1 | Hall CreateHall(string hallDetails) | This method is used to create a Hall using parameterized hallDetails (which is comma separated). Split that string and create a hall object and return that created Hall object to the Main method. |
| 2 | bool RemoveHallFromList(List<Hall> hallList, int index) | This method is used to remove the hall from the parameterized hallList using the parameterized index. Remove the index object from the parameterized index. If the hall is removed then return true and also display the available halls. If the hall list is empty then display "**Hall list empty**". |
| 3 | void DisplayHalls(List<Hall> hallList) | This method is used to display the hall details using the parameterized hallList. |

Consider a class named **Program**, write a Main method to test the above class.  
  
**Note:**  
If the hall adds to the hallList successfully, then print " **Hall Created Successfully** ".

Display the total number of halls in the list, when the user selects the second option(Eg: "**Total 3 Halls**")  
If the list is empty(when the user wants to remove a Hall from the empty list) print a message "**List is Empty**"  and terminate the program.  
Display the list every time in tabular format after removing an element from the hallList. Display the hall list in **insertion order**.

**Input Format:**  
The first line of an input consists of an integer, that corresponds to the choice.  
If the input is 1, then get the input from the user as Hall details(separated by comma) format(hall name, contact number, cost per day, owner name) and create a hall object(use CreateHall method ) and add that object to the hallList.

If the input is 2,  
              if the hallList is not empty ->display the total number of the hall and then get the index number to delete the hall from the list and delete the hall in that index.  
             if the hallList is empty -> halt(terminate) the program.  
  
If the input is 3, then the program will be terminated.  
  
**Output Format:**  
Use the following format to print the output  
**Console.WriteLine("{0}{1,15}{2,15}{3,15}", "Name", "Contact Number", "Cost per day", "Owner Name");**  
Refer sample input and output for formatting specifications.

**[All text in bold corresponds to the input and the rest corresponds to output.]**  
  
**Sample Input and Output 1:**

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Residency,7748554596,2600,Shira**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Rajeshwari,7221022362,1600,Jenifer**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Manas,8224589556,2400,Kesha**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

Total 3 Halls

Enter the index:

**2**

Name Contact Number   Cost per day     Owner Name

Residency     7748554596           2600          Shira

Manas     8224589556           2400          Kesha

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

Total 2 Halls

Enter the index:

**2**

Name Contact Number   Cost per day     Owner Name

Residency     7748554596           2600          Shira

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Hill Town,9665877410,1900,Leonie**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

Total 2 Halls

Enter the index:

**1**

Name Contact Number   Cost per day     Owner Name

Hill Town     9665877410           1900        Leonie

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

Total 1 Halls

Enter the index:

**1**

Hall list empty

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

List is Empty

**Sample Input and Output 2:**

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Residency,78445810223,3600,Shira**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**1**

Enter hall detail:

**Manas,8445159632,2400,Devin**

Hall Created Successfully

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**2**

Total 2 Halls

Enter the index:

**2**

Name Contact Number   Cost per day     Owner Name

Residency    78445810223           3600          Shira

1.Add Hall

2.Remove Hall

3.Exit

Enter the choice:

**3**