

Programming in C#

Mục tiêu:

Sau nội dung thực hành này bạn có khả năng:

Sử dụng các lớp Collections để làm việc với tập các đối tượng

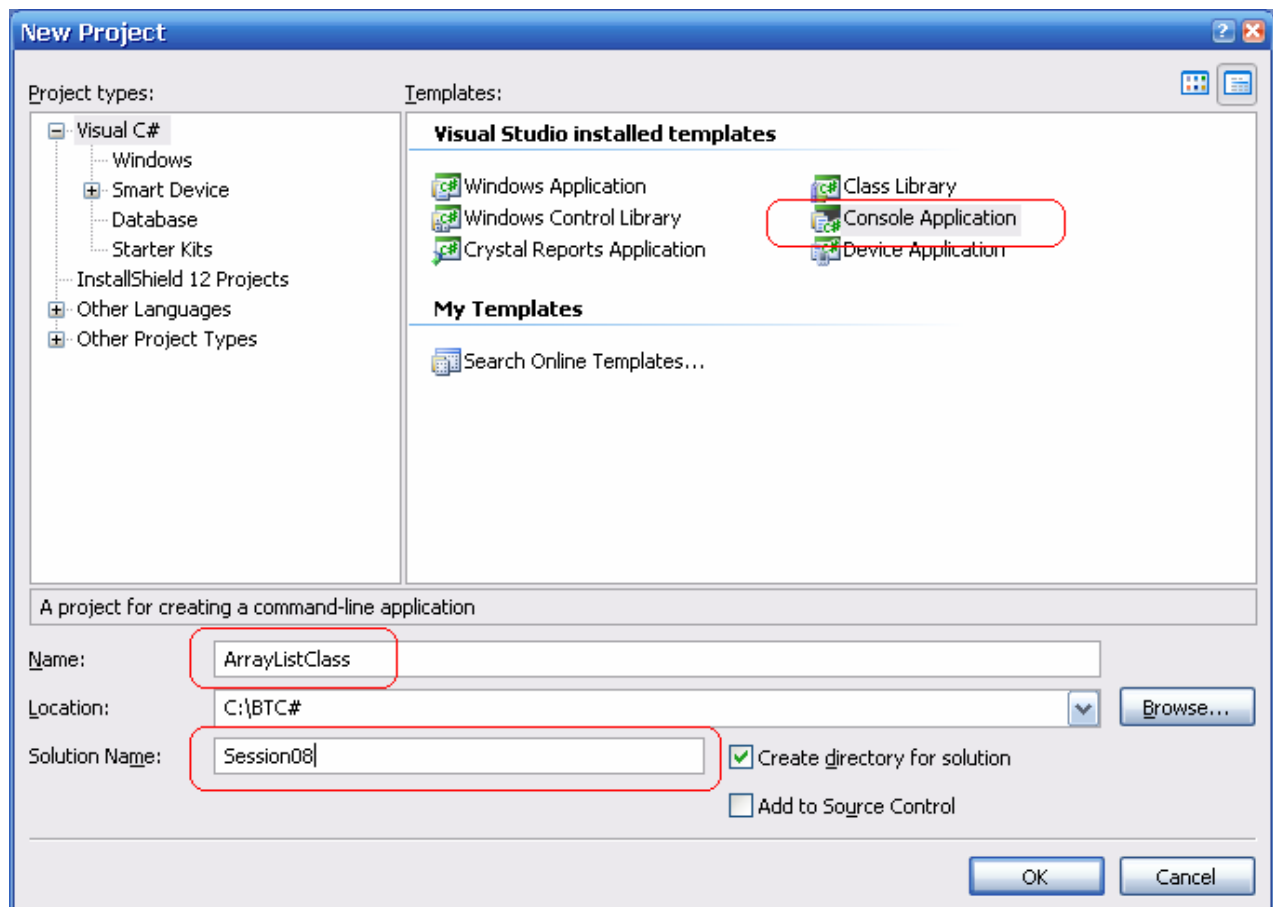
Phần I: Thực hành từng bước – 45 phút

Exercise 1.1: Add user-defined object to an ArrayList

Step 1: Open Visual Studio

Step 2: Select the menu File->New->Project to create console based project named 'ArrayListClass' and

Solution named Session04 as following



Step 3: Rename the class file 'program.cs' to 'ArrayListClass.cs'

Step 4: Replace code in 'ArrayListClass.cs' with given code

```
using System;
using System.Collections;
class Product{
    string
    name;
    double
    cost; int
    onhand;

    public Product(string n, double c, int h){
        name = n;
        cost = c;
        onhand =
        h;
    }

    public override string ToString(){
        return
            String.Format("{0,-10}Cost: {1,6:C}    On hand:
                           {2}", name, cost, onhand);
    }
}
class MainClass{
    public static void Main(){
        ArrayList inv = new ArrayList();
        // Add elements to the list
        inv.Add(new Product("A", 5.9,
3)); inv.Add(new Product("B",
8.2, 2)); inv.Add(new
Product("C", 3.5, 4));
        inv.Add(new Product("D", 1.8,
8));

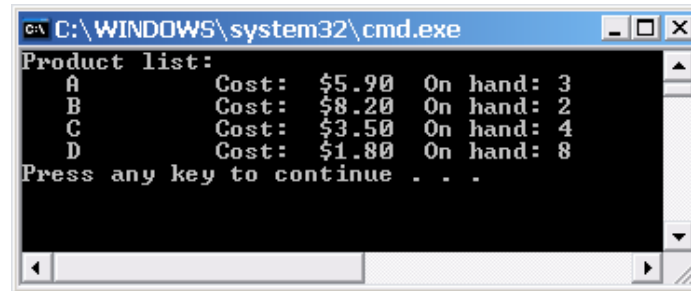
        Console.WriteLine("Product list:");
        foreach (Product i in inv){
            Console.WriteLine("    " + i);
        }
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ArrayListClass' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of the program as following



```
C:\WINDOWS\system32\cmd.exe
Product list:
  A      Cost:  $5.90  On hand: 3
  B      Cost:  $8.20  On hand: 2
  C      Cost:  $3.50  On hand: 4
  D      Cost:  $1.80  On hand: 8
Press any key to continue . . .
```

Exercise 1.2: SortedList class

Step 1: Add a console based project '**SortedListClass**' to the solution

Step 2: Right click on project **SortedListClass** -> set as Startup project

Step 3: Rename the class file 'Program.cs' to '**SortedListClass.cs**'

Step 4: Replace the code in '**SortedListClass.cs**' with the given code

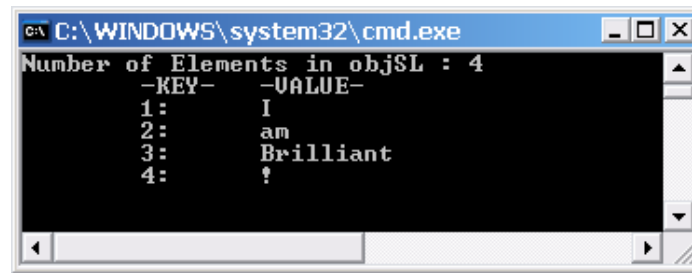
```
using System;
using System.Collections;
public class SampleSortedList{
    public static void Main() {
        SortedList objSL = new SortedList();
        objSL.Add("4", "!");
        objSL.Add("3",
            "Brilliant");
        objSL.Add("2", "am");
        objSL.Add("1", "I");

        Console.WriteLine("Number of Elements in objSL : {0}", objSL.Count);
        Console.WriteLine("\t-KEY-\t-VALUE-");
        for (int i = 0; i < objSL.Count; i++){
            Console.WriteLine("\t{0}:\t{1}",
                objSL.GetKey(i), objSL.GetByIndex(i));
        }
        Console.ReadLine();
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build '**SortedListClass**' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program The output of program as following



```
C:\WINDOWS\system32\cmd.exe
Number of Elements in objSL : 4
-KEY- -VALUE-
1: I
2: am
3: Brilliant
4: !
```

Exercise 1.3: Get the value list using the GetValueList() method of SortedList Class

Step 1: Add a console based project 'GetValueList' to the solution

Step 2: Right click on project GetValueList -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'GetValueList.cs'

Step 4: Replace the code in 'GetValueList.cs' with the given code

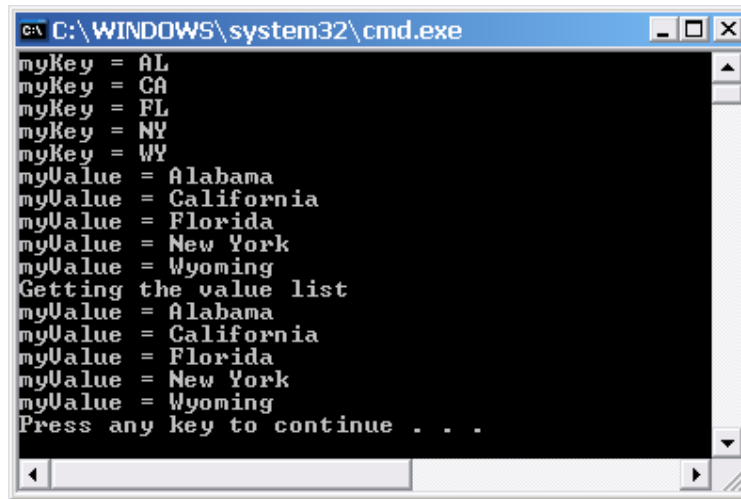
```
using System;
using System.Collections;
class MainClass{
    public static void Main() {
        SortedList mySortedList = new SortedList();
        mySortedList.Add("NY", "New York");
        mySortedList.Add("FL", "Florida");
        mySortedList.Add("AL", "Alabama");
        mySortedList.Add("WY", "Wyoming");
        mySortedList.Add("CA", "California");
        foreach (string myKey in mySortedList.Keys) {
            Console.WriteLine("myKey = " + myKey);
        }
        foreach (string myValue in mySortedList.Values) {
            Console.WriteLine("myValue = " + myValue);
        }
        Console.WriteLine("Getting the value list");
        IList myValueList = mySortedList.GetValueList();
        foreach (string myValue in myValueList) {
            Console.WriteLine("myValue = " + myValue);
        }
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build '**GetValueList**' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
myKey = AL
myKey = CA
myKey = FL
myKey = NY
myKey = WY
myValue = Alabama
myValue = California
myValue = Florida
myValue = New York
myValue = Wyoming
Getting the value list
myValue = Alabama
myValue = California
myValue = Florida
myValue = New York
myValue = Wyoming
Press any key to continue . . .
```

Exercise 1.4: Add key-value pair to Hashtable by using the indexer

- Step 1: Add a console based project '**Hashtable**' to the solution
- Step 2: Right click on project **Hashtable** -> set as Startup project
- Step 3: Rename the class file 'Program.cs' to '**Hashtable.cs**'
- Step 4: Replace the code in '**Hashtable.cs**' with the given code

```
using System;
using System.Collections;

class HashtableDemo{
    public static void Main() {
        Hashtable ht = new Hashtable();

        ht.Add("a", "A");
        ht.Add("b", "B");
        ht.Add("c", "C");
        ht.Add("e", "E");
        ht["f"] = "F";

        // Get a collection of the keys.
        ICollection c = ht.Keys;

        foreach (string str in c) Console.WriteLine(str + ": " + ht[str]);
    }
}
```



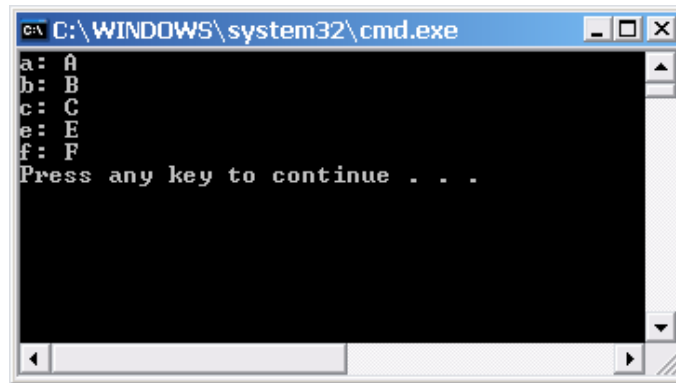
```
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build '**Hashtable**' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
a: A
b: B
c: C
e: E
f: F
Press any key to continue . . .
```

Phần II: Tự thực hành – 60 phút

Exercise 2.1: Temperatures Collection

Tạo một file mã nguồn mới. Trong một phương thức, khai báo một biến `temperatures` kiểu `List`. Thêm một số thành viên vào danh sách.

Viết vòng lệnh lặp `foreach` để đếm số lượng `temperature` mà bằng hoặc lớn hơn 25 độ.

Viết một phương thức `GreaterCount` với chữ ký `static int GreaterCount(List list, double min)` { ... } mà trả về số lượng phần tử của danh sách mà lớn hoặc bằng `min`.

Gọi phương thức trong danh sách `temperatures` của bạn.

Create a new source file. In a method, declare a variable `temperatures` of type `List`. Add some numbers to the list. Write a `foreach` loop to count the number of `temperatures` that equal or exceed 25 degrees. Write a method `GreaterCount` with signature `static int GreaterCount(List list, double min)` { ... } that returns the number of elements of list that are greater than or equal to `min`. Call the method on your `temperatures` list.

Exercise 2.2: GreaterCount Method

Viết một phương thức với chữ ký `static int GreaterCount(IEnumerable eble, double min) { ... }` mà trả về số lượng phân tử của một enumerable mà lớn hơn hoặc bằng min.

Gọi phương thức đó trên một mảng kiểu `double`.

Write a method with signature `static int GreaterCount(IEnumerable eble, double min) { ... }` that returns the number of elements of the enumerable that are greater than or equal to min. Then call the method on an array of type `double[]`.