

Lab 07

Collections

Mục tiêu

- Sử dụng một số lớp Collections và Collection Generics

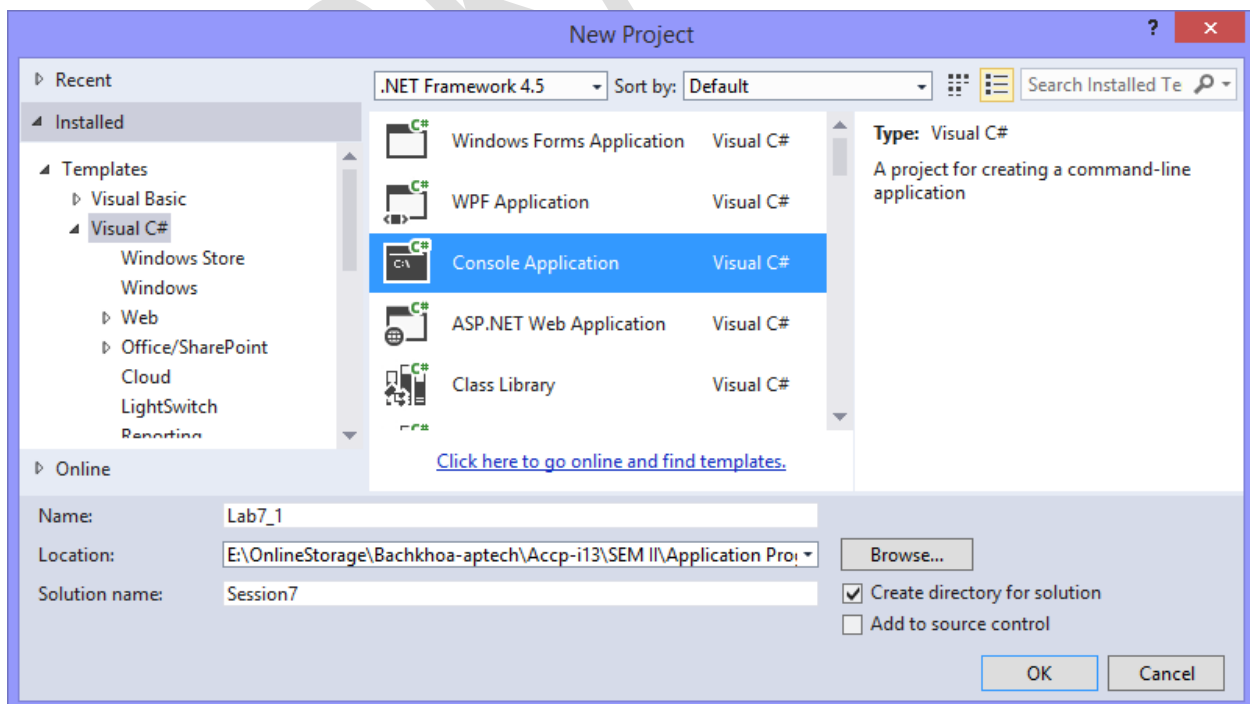
Phần I Bài tập step by step

Bài 7.1

Viết chương trình C# minh họa việc sử dụng ArrayList

- Tạo lớp Product với các trường: name, cost, onhand
- Ghi đè phương thức ToString của lớp Product để trả về chuỗi thông tin của Product
- Viết code thêm 5 sản phẩm vào ArrayList và hiển thị ra màn hình.

Bước 1: Mở Visual Studio 2013, vào menu File -> New -> Project -> chọn loại project “Console Application”, nhập tên project, tên solution -> OK.



Bước 2: Tạo lớp Product với code gợi ý như sau:

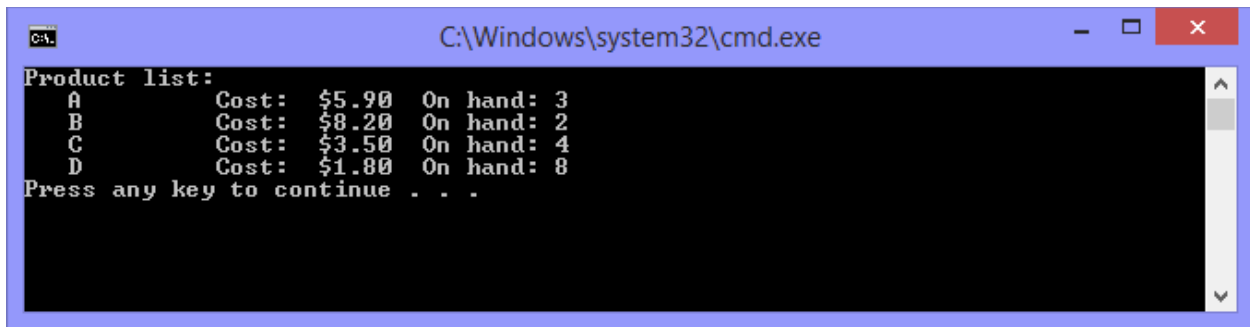
```
//tạo lớp Product
class Product
{
    //khai báo trường
    string name;
    double cost;
    int onhand;
    //constructor
    public Product(string n, double c, int h)
    {
        name = n;
        cost = c;
        onhand = h;
    }
    //ghi đè phương thức
    public override string ToString()
    {
        return String.Format("{0,-10}Cost: {1,6:C} On hand: {2}",name,
        cost, onhand);
    }
}
```

Bước 3: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
static void Main(string[] args)
{
    //tạo ArrayList
    ArrayList inv = new ArrayList();
    //thêm phần tử vào danh sách
    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));
    //in ra danh sách phần tử
    Console.WriteLine("Product list:");
}
```

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

Bước 4: Nhấn Ctrl+F5 để chạy và xem kết quả



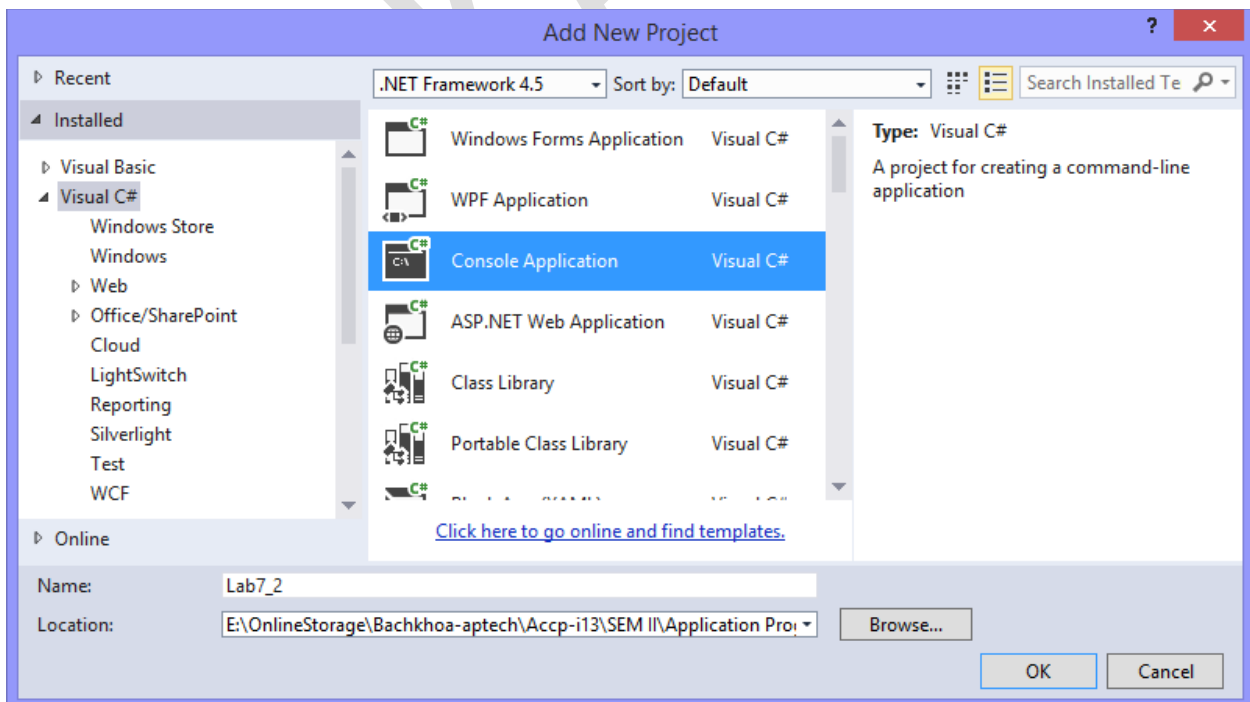
```
C:\Windows\system32\cmd.exe

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 . . .
```

Bài 7.2

Viết chương trình C# minh họa việc sử dụng lớp SortedList Generic

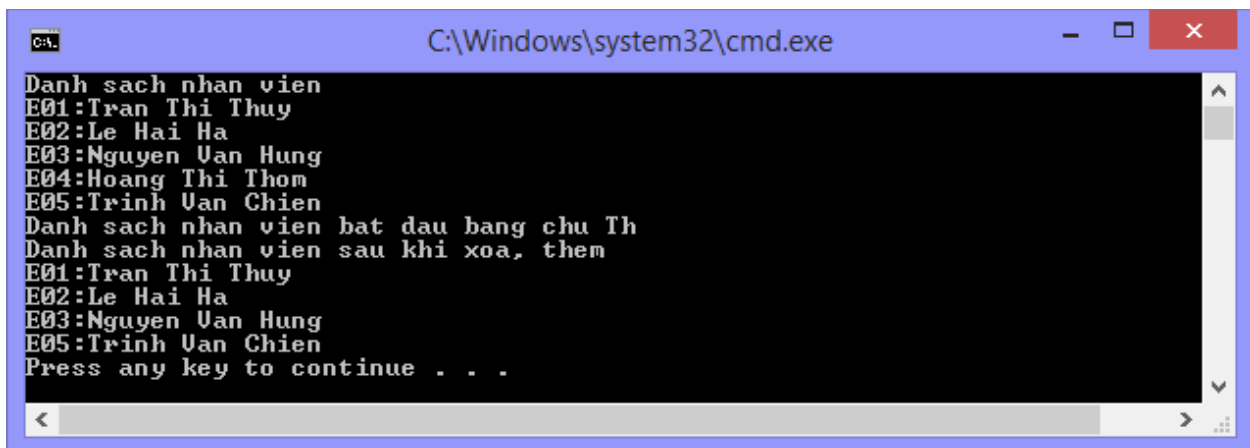
Bước 1: Kích chuột phải vào Solution “Session7” chọn Add -> New Project -> nhập tên.



Bước 2: Viết code cho hàm Main theo gợi ý sau.

```
static void Main(string[] args)
{
    //Tạo sortedlist generic
    SortedList<string, string> listEm = new SortedList<string, string>();
    //đưa dữ liệu vào
    listEm.Add("E01", "Tran Thi Thuy");
    listEm.Add("E02", "Le Hai Ha");
    listEm.Add("E03", "Nguyen Van Hung");
    listEm.Add("E04", "Hoang Thi Thom");
    listEm.Add("E05", "Trinh Van Chien");
    //in danh sách
    Console.WriteLine("Danh sach nhan vien");
    foreach (var key in listEm.Keys)
    {
        Console.WriteLine(key + ":" + listEm[key]);
    }
    //tìm kiếm tất cả các nhân viên có tên bắt đầu bằng chữ Th
    Console.WriteLine("Danh sach nhan vien bat dau bang chu Th");
    foreach (var key in listEm.Keys)
    {
        if (listEm[key].StartsWith("Th"))
            Console.WriteLine(key + ":" + listEm[key]);
    }
    //xóa nhân viên có mã E04
    listEm.Remove("E04");
    //Kiểm tra nếu chưa có nhân viên E06 thì thêm vào
    if (!listEm.ContainsKey("E06"))
        listEm.Add("E06", "Nguyen Hoai Linh");
    //in danh sách sau khi xóa, thêm
    Console.WriteLine("Danh sach nhan vien sau khi xoa, them");
    foreach (var key in listEm.Keys)
    {
        Console.WriteLine(key + ":" + listEm[key]);
    }
}
```

Bước 3: Ctrl+F5 để chạy và kiểm tra kết quả.

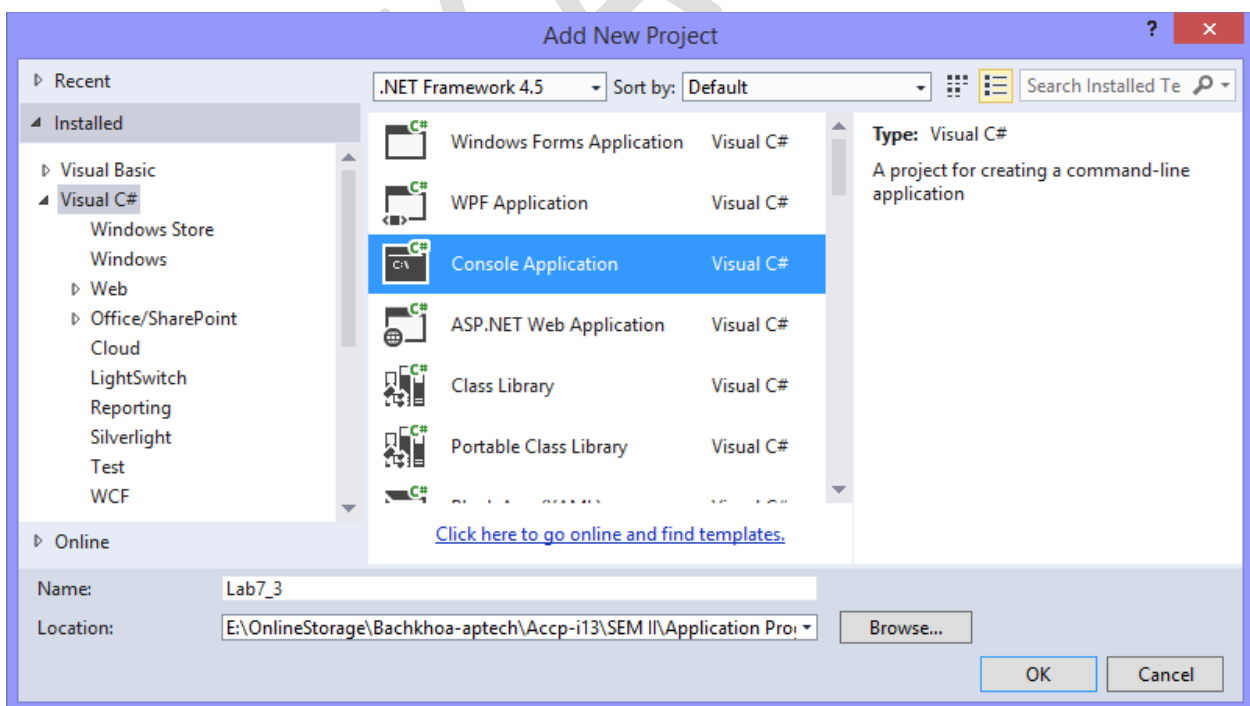


```
C:\Windows\system32\cmd.exe
Danh sach nhan vien
E01:Tran Thi Thuy
E02:Le Hai Ha
E03:Nguyen Van Hung
E04:Hoang Thi Thom
E05:Trinh Van Chien
Danh sach nhan vien bat dau bang chu Th
Danh sach nhan vien sau khi xoa, them
E01:Tran Thi Thuy
E02:Le Hai Ha
E03:Nguyen Van Hung
E05:Trinh Van Chien
Press any key to continue . . .
```

Bài 7.3

Viết chương trình C# minh họa việc sử dụng lớp List Generic để lưu trữ 1 danh sách sinh viên, tìm sinh viên có điểm trung bình cao nhất, sắp xếp danh sách sinh viên tăng dần theo điểm trung bình

Bước 1: Kích chuột phải vào Solution “Session7” chọn Add -> New Project ->nhập tên.



Bước 2: Tạo lớp Student với code gợi ý như sau.

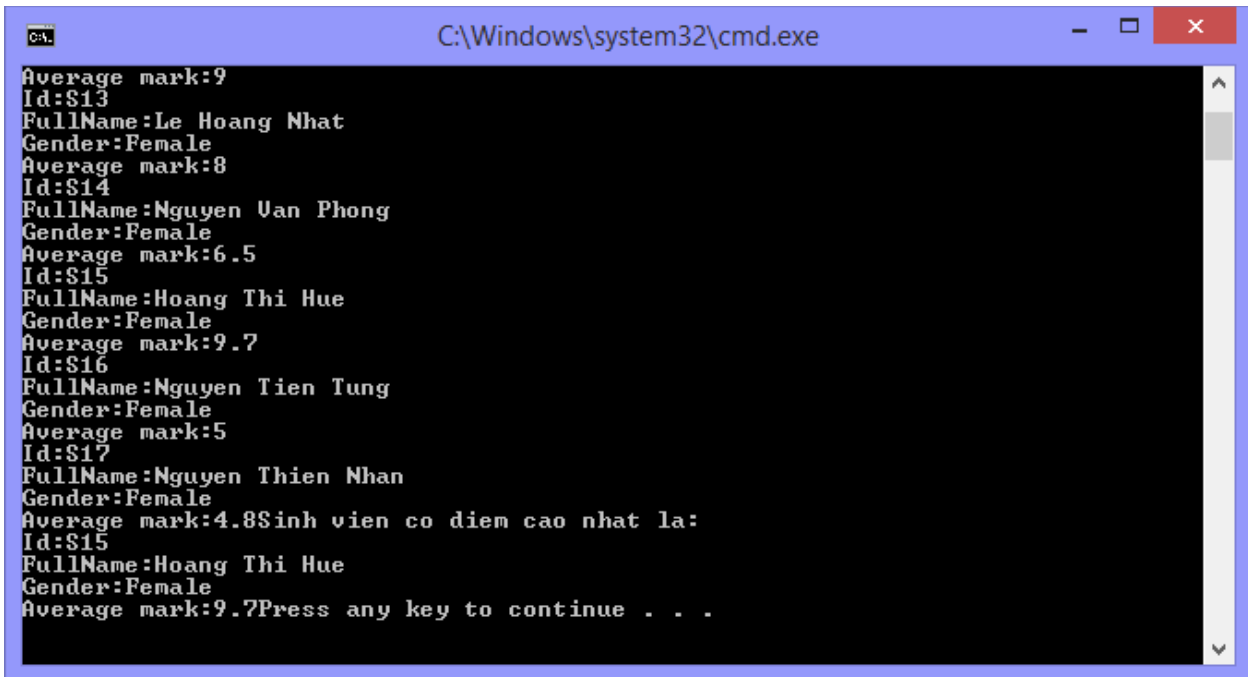
```
//lớp sinh viên thực thi từ giao diện IComparable để thực hiện so sánh điểm
trung bình 2 sinh viên với nhau
class Student:IComparable<Student>
{
    //khai báo các thuộc tính tự động
    public string Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public bool Gender { get; set; }
    public double Avg { get; set; }
    //ghi đè phương thức toString
    public override string ToString()
    {
        return "\nId:" + Id + "\nFullName:" + FirstName + " " + LastName
            + "\nGender:" + (Gender ? "Male" : "Female") + "\nAverage mark:" +
            Avg;
    }
    //thực thi phương thức CompareTo và so sánh tuổi để sắp xếp tăng dần,
    nếu sắp xếp giảm dần thì đảo lại đối tượng là xong
    public int CompareTo(Student other)
    {
        return this.Avg.CompareTo(other.Avg);
    }
}
```

Bước 3: Viết code cho hàm Main theo gợi ý sau.

```
static void Main(string[] args)
{
    //khai báo và khởi tạo danh sách sinh viên sử dụng List generic
    List<Student> list = new List<Student>()
    {
        new Student{Id="S10",FirstName="Nguyen
        Thu",LastName="Phuong",Avg=9.5},
        new Student{Id="S12",FirstName="Tran Thi",LastName="Thuy",Avg=9.0},
        new Student{Id="S13",FirstName="Le Hoang",LastName="Nhat",Avg=8.0},
    }
```

```
new Student{Id="S14",FirstName="Nguyen
Van",LastName="Phong",Avg=6.5},
new Student{Id="S15",FirstName="Hoang Thi",LastName="Hue",Avg=9.7},
new Student{Id="S16",FirstName="Nguyen
Tien",LastName="Tung",Avg=5.0},
new Student{Id="S17",FirstName="Nguyen
Thien",LastName="Nhan",Avg=4.8}
};
//in danh sách sinh viên
Console.WriteLine("Danh sach sinh vien:");
foreach (var st in list)
{
    Console.Write(st);
}
//tìm sinh viên có điểm trung bình cao nhất
double max = list[0].Avg;
Student stmax = list[0];
foreach (var st in list)
{
    if (st.Avg > max)
    {
        max = st.Avg;
        stmax = st;
    }
}
//in kết quả
Console.WriteLine("\nSinh vien co diem cao nhat la:");
Console.Write(stmax);
//sắp xếp danh sách sinh viên tăng dần theo điểm trung bình
list.Sort();
Console.WriteLine("\nDanh sach sinh vien diem trung binh tang dan:");
foreach (var st in list)
{
    Console.Write(st);
}
}
```

Bước 4: Ctrl+F5 để chạy và kiểm tra kết quả.



```
C:\Windows\system32\cmd.exe
Average mark:9
Id:$13
FullName:Le Hoang Nhat
Gender:Female
Average mark:8
Id:$14
FullName:Nguyen Van Phong
Gender:Female
Average mark:6.5
Id:$15
FullName:Hoang Thi Hue
Gender:Female
Average mark:9.7
Id:$16
FullName:Nguyen Tien Tung
Gender:Female
Average mark:5
Id:$17
FullName:Nguyen Thien Nhan
Gender:Female
Average mark:4.8Sinh vien co diem cao nhat la:
Id:$15
FullName:Hoang Thi Hue
Gender:Female
Average mark:9.7Press any key to continue . . .
```

Phần II Bài tập tự làm

Bài 7.1: Viết chương trình C# minh họa việc sử dụng lớp List theo yêu cầu sau:

- Tạo lớp Book với các thông tin (Id, Title, Author, Publisher, Year, Price).
- Tạo 1 collection gồm 10 quyển sách.
- In danh sách các quyển sách tăng dần theo giá.
- Tìm quyển sách có title trùng với giá trị nhập từ bàn phím.
- Đưa ra những quyển sách xuất bản năm 2014.
- Xóa những quyển sách của nhà xuất bản “Nhi Dong”.

Bài 7.2: Viết chương trình C# minh họa việc sử dụng lớp Hashtable theo yêu cầu sau:

- Tạo một hashtable lưu danh sách các ngày trong tuần với key từ 1-8
- Tìm ngày TueDay, in ra thông báo nếu tìm thấy hoặc không
- In ra các ngày trong tuần bao gồm cả key và value.

Bài 7.3: Viết chương trình C# minh họa việc sử dụng lớp List theo yêu cầu sau:

- Tạo lớp Car lưu trữ thông tin (name, color)
- Khởi tạo một List lưu trữ danh sách 10 Car
- Xóa các Car có màu Red
- In danh sách các Car.

HẾT