

## BÀI TẬP LAB NGÀY 12/09/2017 – MẢNG ĐỘNG & LỚP

**Sau khi thực hành xong bài này, học viên có khả năng:**

- Khai báo và khởi tạo biến kiểu tập hợp.
- Xây dựng ứng dụng đơn giản để hiện thực việc sử dụng tập hợp (mảng động).
- Khai báo và xây dựng ứng dụng Lập trình Hướng đối tượng cơ bản

### Bài tập 1:

Viết 1 ứng dụng Console thực hiện các chức năng sau với List, ArrayList hoặc Hashtable

- Nhập các tên hàng hóa (kiểu chuỗi) từ bàn phím, thêm vào List/ArrayList khi nào người dùng nhập vào chuỗi "stop" thì ngừng.
- Xuất các hàng hóa ra màn hình.
- Nhập vào tên một hàng hóa, tiến hành tìm kiếm theo tên, trả về **true** nếu tìm thấy, ngược lại trả về **false**
- Xóa một hàng hóa.
- Thêm 1 hàng hóa vào vị trí index bất kỳ với index nhập từ bàn phím.

### Hướng dẫn:

Trường hợp sử dụng List

```
List<string> mang = new List<string>();
string tmp;
//nhập chuỗi đưa vào mảng cho tới khi gặp STOP
while (true)
{
    Console.Write("Chuoi nhap : ");
    tmp = Console.ReadLine().Trim();
    if (tmp.ToUpper() == "STOP")
    {
        break; //thoát while
    }
    //thêm vào mảng
    mang.Add(tmp);
    Console.WriteLine("Count = {0}, Capacity = {1}", mang.Count, mang.Capacity);
} //end while
XuatMang(mang);
//Nhập chuỗi cần tìm
Console.Write("Can tim : ");
tmp = Console.ReadLine().Trim();
if (mang.Contains(tmp)) //true : có chứa
{
    //...
}
```

```

        Console.WriteLine("Tim thay {0} tai vi tri {1}", tmp, mang.IndexOf(tmp));
        mang.Remove(tmp);
        //mang.RemoveAt(mang.IndexOf(tmp));
        XuatMang(mang);
    }
    mang.Sort();
    Console.WriteLine("Sau khi sap : ");
    XuatMang(mang);
    mang.Clear();
    Console.WriteLine("Sau khi xoa, mang con {0} phan tu", mang.Count);

```

Hàm XuatMang:

```

static void XuatMang(List<string> M)
{
    string chuoi = string.Join(", ", M);
    Console.WriteLine(chuoi);
}

```

Trường hợp sử dụng ArrayList

```

//Khai bao va su dung ArrayList
ArrayList al = new ArrayList();
string n;
do
{
    n = Console.ReadLine();
    if (n == "stop") break;
    al.Add(n);
} while (true);
XuatMang(al);
Console.WriteLine("So phan tu hien co:{0}", al.Count);
Console.WriteLine("\n Nhap phan tu can tim:");
string x = Console.ReadLine();
int vitri = al.BinarySearch(x);
if (vitri >= 0)
{
    Console.WriteLine("\nTim thay");
    Console.WriteLine("Xoa phan tu(y/n)");
    string ch = Console.ReadLine();
    if (ch.ToUpper() == "Y")
        al.RemoveAt(vitri);
    Console.WriteLine("Mang sau khi xoa:");
    XuatMang(al);
}
else
    Console.WriteLine("\nKhong tim thay");
//Dao tat ca cac phan tu:
al.Reverse();
XuatMang(al);
//Dao mot phan cua mang: al.reverse(vitri,soluong);

```

**Bài tập 2:**

Thông tin một mặt hàng bao gồm: Mã mặt hàng (MaMH - số nguyên), tên mặt hàng (TenMH - chuỗi), số lượng (SoLuong - số nguyên), đơn giá (DonGia - số thực).

- Khai báo cấu trúc mặt hàng nói trên, bao gồm cả hàm tạo và hàm tính thành tiền.
- Viết hàm thêm một mặt hàng vào trong danh sách.
- Viết hàm tìm một mặt hàng dựa vào mã mặt hàng, trả về kiểu bool (có hay không).
- Viết hàm xuất danh sách các mặt hàng.
- viết hàm xóa mặt hàng dựa vào mã mặt hàng.
- Viết hàm main thực hiện các chức năng sau:
  - o nhập vào danh sách các mặt hàng. Nhập xong mỗi mặt hàng hỏi người dùng có tiếp tục nhập hay không?
  - o xuất danh sách các mặt hàng.
  - o nhập vào mã mặt hàng, tiến hành tìm kiếm mặt hàng nói trên. Nếu tìm thấy, xóa mặt hàng và xuất danh sách các mặt hàng (sau khi xóa).

**Hướng dẫn:**

Khai báo cấu trúc Mặt hàng

```
6 references
struct HangHoa
{
    public int MaHH;
    public string TenHH;
    public int SoLuong;
    public double DonGia;
    0 references
    public override string ToString()
    {
        return string.Format("{0} - {1} - {2} cái x {3} d = {4}",
            MaHH, TenHH, SoLuong, DonGia, SoLuong * DonGia);
    }
}
```

Demo đơn giản:

```
static void Main(string[] args)
{
    List<HangHoa> dssp = new List<HangHoa>();
    HangHoa hh1;
    hh1.MaHH = 1; hh1.TenHH = "Bia SG";
    hh1.SoLuong = 15; hh1.DonGia = 13500;
    dssp.Add(hh1);
    HangHoa hh2 = new HangHoa()
    {
        MaHH = 2, TenHH = "Tra da", SoLuong = 1, DonGia = 1000
    };
    dssp.Add(hh2);
    foreach (HangHoa hh in dssp)
    {
        string chuoai = string.Format("{0} - {1} - {2} cai x {3} d = {4}",
            hh.MaHH, hh.TenHH, hh.SoLuong, hh.DonGia, hh.SoLuong * hh.DonGia);
        //Console.WriteLine(chuoai);
        Console.WriteLine(hh); //hh.ToString()
    }
    Console.ReadKey();
}
```

Sinh viên tự thiết kế & làm những hàm còn lại.

### Bài tập 3:

#### 1.1 Lớp Sinh viên:

Tạo ứng dụng Console, thêm vào 1 class tên SinhVien bao gồm các thành phần dữ liệu: MaSV, HoTen, NgaySinh, DiaChi, DienThoai.

- Khai báo và định nghĩa các Constructor tham số và không tham số.
- Khai báo và định nghĩa các properties để truy cập đến giá trị của các thành phần dữ liệu (get, set) và viết thêm 1 properties chỉ đọc (get) dùng để lấy LayTuoi của SinhVien.
- Viết hàm Main() để kiểm tra lớp trên.

**Hướng dẫn:**

```

class SinhVien
{
    //Automatic Property : sử dụng khi không cần kiểm tra điều kiện cho set
    public int MaSV { get; set; }
    public string HoTen { get; set; }
    public DateTime NgaySinh { get; set; }
    public string DiaChi { get; set; }
    public string DienThoai { get; set; }
    //Property ReadOnly
    public int Tuoi
    {
        get { return DateTime.Now.Year - NgaySinh.Year; }
    }
    public override string ToString()
    {
        return string.Format("{0} - {1} sinh {2}, {3} tuoi, {4}, {5}.", MaSV, HoTen, NgaySinh.ToString("dd/MM/yyyy"), Tuoi, DienThoai, DiaChi);
    }
}

```

**Demo:**

```

static void Main(string[] args)
{
    List<SinhVien> dssv = new List<SinhVien>();
    dssv.Add(new SinhVien {
        MaSV = 1, HoTen = "Tran Van Teo",
        DienThoai = "0909789567", DiaChi = "123 Truong Dinh",
        NgaySinh = new DateTime(2000, 11, 28)
    });
    dssv.Add(new SinhVien {
        MaSV = 2,
        HoTen = "Tran Van Quang",
        DienThoai = "0909567789",
        DiaChi = "123 Truong Cong Dinh",
        NgaySinh = new DateTime(1999, 11, 28)
    });
    foreach (SinhVien sv in dssv)
    {
        Console.WriteLine(sv);
    }
}

```

**1.2 Lớp Hình chữ nhật**

Tạo ứng dụng Console, thêm vào 1 class tên HìnhChuNhat gồm các thành phần dữ liệu: mChieuCao, mChieuRong. Thực hiện các yêu cầu sau:

- Viết các constructor không tham số và có tham số để khởi tạo các giá trị cho mChieuCao và mChieuRong.
- Viết các properties để truy cập đến 2 thành phần trên.

- Viết 2 phương thức trả về chu vi ( $mChieuDai + mChieuRong$ ) x 2 và diện tích hình chữ nhật ( $mChieuDai \times mChieuRong$ ).
- Phương thức xuất thông tin hình chữ nhật: rộng, dài, diện tích và chu vi.
- Viết phương thức kiểm tra xem hình chữ nhật có phải là hình vuông hay không?
- Viết hàm Main để kiểm tra lớp `HinhChuNhat`: khởi tạo đối tượng, các giá trị cho đối tượng, gọi các phương thức, ...

### 1.3 Lớp sản phẩm

Xây dựng lớp sản phẩm bao gồm các trường mã, tên, giá.

- Hàm tạo 3 đối số và hàm tạo mặc định.
- Phương thức tính thuế VAT
- Phương thức nhập mã, tên, giá từ bàn phím
- Phương thức xuất thông tin sản phẩm: mã, tên, giá và thuế VAT
- Viết chương trình nhập 2 sản phẩm từ bàn phím
- Viết chương trình thực hiện các công việc sau
  - Tạo đối tượng sản phẩm
  - Gọi phương thức nhập
  - Gọi phương thức xuất

-----oOo-----