

Họ và Tên: Nguyễn Thanh Kiên.

MSSV: 22110092.

Bài tập:

Xây dựng lớp Customer (Khách hàng) gồm các thuộc tính CustomerID (ví dụ dạng C001), CustomerName, CustomerAddress, CustomerPhone, CustomerType. Trong đó CustomerType có các loại sau (TrungThanh, TiemNang, CanQuanTam, KhachHangKhac). Sau đó xây dựng lớp Company với các thuộc tính (ListOfCustomers (danh sách các khách hàng của công ty), CompanyName (Tên công ty))

- Xây dựng các phương thức thiết lập cho hai lớp Customer và Company 📌
- Xây dựng các phương thức get và set cho từng thuộc tính 📌
- Xây dựng các hàm CustomerInfo và CompanyInfo để xuất thông tin của khách hàng và của công ty 📌
- Viết hàm SearchCustomer với tham số truyền vào có thể là Tên Khách Hàng hoặc là Index (dùng generic) 📌
- Viết một hàm mở rộng ConvertToString cho lớp Customer để có thể dùng Console.WriteLine xuất thông tin khách hàng (dùng Extensions) 📌
- Xây dựng một từ điển để định nghĩa các khái niệm loại khách hàng và xuất thông tin đó kèm theo thông tin khách hàng (dùng Dictionary) 📌
- Tạo một sự kiện để cập nhật số lượng khách hàng khi thêm hoặc xóa khách hàng đó khỏi công ty 📌

Làm thêm: Không.

Bài làm:

1. Xây dựng các phương thức thiết lập cho hai lớp Customer và Company. Lớp Customer (Khách hàng) gồm các thuộc tính CustomerID (ví dụ dạng C001), CustomerName, CustomerAddress, CustomerPhone, CustomerType. Trong đó CustomerType có các loại sau (TrungThanh, TiemNang, CanQuanTam, KhachHangKhac). Sau đó xây dựng lớp Company với các thuộc tính (ListOfCustomers (danh sách các khách hàng của công ty), CompanyName (Tên công ty))

+ Đầu tiên, em khai báo các thư viện System. Sau đó, để xây dựng phương thức thiết lập cho hai lớp, em tạo một namespace MainData, trong đó có chứa hai lớp Customer và Company.

+ Ở lớp Customer có yêu cầu thuộc tính CustomerType gồm nhiều loại, vì vậy, em sử dụng kiểu dữ liệu Enum, là kiểu dữ liệu dùng để gán một cách thoả mái nhất, nhất là các yêu cầu về loại trong 1 thuộc tính. Lập trình .Net thì chúng ta dùng Enum nhiều mới chuẩn.

+ Em khai báo lớp Customer với 3 phương thức thiết lập dạng chuỗi, 1 phương thức thiết lập dạng số và 1 phương thức còn lại thiết lập dưới dạng Enum.

```
namespace MainData
{
    8 references
    public enum CustomerType { TrungThanh, TiemNang, CanQuanTam, KhachHangKhac };

    8 references
    public class Customer
    {
        string CustomerID = "C001";
        string CustomerName = "Nguyen Thanh Kien";
        string CustomerAddress = "Vinhomes Grand Park S1.02";
        int CustomerPhone = 0123456789;
        CustomerType type = CustomerType.TrungThanh;

        0 references
        public Customer() {}

        3 references
        public Customer(string CustomerID, string CustomerName, string CustomerAddress, int CustomerPhone, CustomerType type)
        {
            this.CustomerID = CustomerID;
            this.CustomerName = CustomerName;
            this.CustomerAddress = CustomerAddress;
            this.CustomerPhone = CustomerPhone;
            this.type = type;
        }
    }
}
```

+ Ở lớp Company, em khai báo với 2 phương thức thiết lập, 1 là sử dụng kiểu dữ liệu List<> cho ListOfCustomers và phương thức thiết lập còn lại là dạng chuỗi cho tên công ty.

```

public class Company
{
    6 references
    public List<Customer> ListOfCustomers { get; set; }
    string CompanyName;

    1 reference
    public Company()
    {
        ListOfCustomers = new List<Customer>();
        CompanyName = "Not Assigned";
    }
}

```

2. Xây dựng các phương thức get và set cho từng thuộc tính

Em xây dựng phương thức get và set cho từng thuộc tính như sau.

+ Ở lớp Customer:

```

0 references
public Customer() {}
3 references
public Customer(string CustomerID, string CustomerName, string CustomerAddress, int CustomerPhone, CustomerType type)
{
    this.CustomerID = CustomerID;
    this.CustomerName = CustomerName;
    this.CustomerAddress = CustomerAddress;
    this.CustomerPhone = CustomerPhone;
    this.type = type;
}
0 references
public string ValCustomerID
{
    get { return CustomerID; }
    set { CustomerID = value; }
}
0 references
public string ValCustomerName
{
    get { return CustomerName; }
    set { CustomerName = value; }
}
0 references
public string ValCustomerAddress
{
    get { return CustomerAddress; }
    set { CustomerAddress = value; }
}

```

```

0 references
public int ValCustomerPhone
{
    get { return CustomerPhone; }
    set { CustomerPhone = value; }
}

0 references
public CustomerType valtype
{
    get { return type; }
    set { type = value; }
}

```

+ Ở lớp Company:

```

1 reference
public string ValCompanyName
{
    get { return CompanyName; }
    set { CompanyName = value; }
}

```

Chú ý rằng, vì ListOfCustomer đã có sẵn thuộc tính tự động, vì vậy ta sẽ không cần khai báo thêm riêng get và set cho List đó.

3. Xây dựng các hàm CustomerInfo và CompanyInfo để xuất thông tin của khách hàng và của công ty

Để xuất thông tin khách hàng và công ty ra màn hình Console, em xây dựng thêm 2 hàm Info trong từng class tương ứng.

+ Ở lớp Customer, em xây dựng hàm CustomerInfo:

```

1 reference
public void CustomerInfo()
{
    string typeName = Enum.GetName(typeof(CustomerType), type);

    if(CustomerID != "C001")
        Console.WriteLine("Khách Hàng {0} có tên là {1}, địa chỉ là {2}, số điện thoại là {3}, \nLa đang khách hàng {4}\n",
            CustomerID, CustomerName, CustomerAddress, CustomerPhone, typeName);
}

```

Chú ý rằng, em không xuất ra thông tin khách hàng tương ứng với CustomerID C001.

+ Ở lớp Company, em xây dựng hàm CompanyInfo:

```
1 reference
public void CompanyInfo()
{
    Console.WriteLine("Cong ty {0} co {1} khách hàng. Gồm các khách hàng như sau:\n", CompanyName, ListOfCustomers.Count);
    foreach (Customer rm in ListOfCustomers)
    {
        rm.CustomerInfo();
    }
}
```

+ Sau đó, em tạo một class Program trong namespace CompanyManagement, class này chứa hàm main dùng để chứa thông tin 3 customer mà em tạo sẵn với các thông tin tương ứng có sẵn để xuất ra màn hình.

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        Company company = new Company();
        company.ValCompanyName = "Vinhomes";
        company.ListOfCustomers.Add(new Customer("C002", "Nguyen Dang Khoi", "Vinhomes Grand Park S7.02", 0987654321, CustomerType.TiemNang));
        company.ListOfCustomers.Add(new Customer("C003", "Tran Hong Tinh", "Vinhomes Grand Park S5.03", 0135792468, CustomerType.CanQuanTam));
        company.ListOfCustomers.Add(new Customer("C004", "KevinSanjaya", "Vinhomes Grand Park S7.02", 0246813579, CustomerType.KhachHangKhac));

        company.CompanyInfo();

        Console.ReadKey();
    }
}
```

Màn hình Console:

```
Cong ty Vinhomes co 3 khách hàng. Gồm các khách hàng như sau:

Khách Hàng C002 có tên là Nguyen Dang Khoi, địa chỉ là Vinhomes Grand Park S7.02, số điện thoại là 987654321,
La dang khách hàng TiemNang

Khách Hàng C003 có tên là Tran Hong Tinh, địa chỉ là Vinhomes Grand Park S5.03, số điện thoại là 135792468,
La dang khách hàng CanQuanTam

Khách Hàng C004 có tên là KevinSanjaya, địa chỉ là Vinhomes Grand Park S7.02, số điện thoại là 246813579,
La dang khách hàng KhachHangKhac
```

- Viết hàm SearchCustomer với tham số truyền vào có thể là Tên Khách Hàng hoặc là Index (dùng generic).

+ Trong lớp Company, em khai báo hàm SearchCustomer với sử dụng generic tham số truyền vào. Tức là em tạo kiểu dữ liệu generic, kiểu dữ liệu tổng thể. Lớp này sẽ được xác định kiểu dữ liệu gì khi mình truyền vào. Nếu T là kiểu string thì sẽ tìm kiếm dưới dạng string, tìm kiếm xong sẽ trả về đối tượng r. Hàm tìm kiếm sẽ lục tìm từng customer và kiểm tra customerID bằng đúng ID mình search và trả về cho chúng ta. Nếu T là kiểu số, em sẽ convert nó dưới dạng 32 bit và so sánh với số lượng customer. Còn trong trường hợp tìm kiếm không chữ hay không số thì trả về Customer is not found.

```
4 references
public Customer SearchCustomer<T> (T search)
{
    Customer r = new Customer();
    if (typeof(T) == typeof(string))
    {
        r = ListOfCustomers.FirstOrDefault(o => o.ValCustomerID == search.ToString());
        if (r != null)
            return r;
    }
    else if (typeof(T) == typeof(int))
    {
        if (Convert.ToInt32(search) < ListOfCustomers.Count)
            return ListOfCustomers[Convert.ToInt32(search)];
    }
    Console.WriteLine("Customer is not found");
    return new Customer();
}
```

+ Em xây dựng hàm Main với các test case search như sau:

```
company.CompanyInfo();

Console.WriteLine("Searching Process:");
company.SearchCustomer("C002").CustomerInfo();
company.SearchCustomer("C005").CustomerInfo();
company.SearchCustomer(2).CustomerInfo();
company.SearchCustomer(3).CustomerInfo();

Console.ReadKey();
}
```

Màn hình Console:

```
Searching Process:
Khach Hang C002 co ten la Nguyen Dang Khoi, dia chi la Vinhomes Grand Park S7.02, so dien thoai la 987654321,
La dang khach hang TiemNang

Customer is not found
Khach Hang C004 co ten la KevinSanjaya, dia chi la Vinhomes Grand Park S7.02, so dien thoai la 246813579,
La dang khach hang KhachHangKhac

Customer is not found
```

Ở test case “C005” không cho ra kết quả vì C005 không tồn tại.

Ở test case dạng số, khi gọi SearchCustomer(3), em đang tìm phòng với chỉ số 3. Tuy nhiên, **chỉ số** ở đây bắt đầu từ 0, không phải từ 1. Danh sách Customers của em hiện có 3 người: C002 -> index = 0, C003 -> index = 1, C004 -> index = 2. Vì vậy, với index = 3, sẽ không tìm thấy customer tương ứng vì vượt quá customer.count với chỉ số tối đa là 2.

- Viết một hàm mở rộng ConvertToString cho lớp Customer để có thể dùng Console.WriteLine xuất thông tin khách hàng (dùng Extensions)

+ Trong namespace MainData, em tạo lớp MyExtensions để tạo thêm một hàm mở rộng hàm ConvertToString cho lớp Customer để xuất thông tin khách hàng. Mở rộng phương thức là mở rộng phương thức có sẵn, lớp MyExtensions này sẽ giúp em viết lại hàm ConvertToString dưới kiểu của mình. Mở rộng sẽ giúp em thay thế hoặc là tạo thêm, vì em đã mở rộng cho Customers rồi nên sẽ thực hiện phương thức này sau khi mở rộng

```
public static class MyExtensions
{
    4 references
    public static string ConvertToString(this Customer customer)
    {
        string typeName = Enum.GetName(typeof(CustomerType), customer.valtype);

        if (customer.ValCustomerID != "C001")
            return String.Format("Khach Hang: {0} - {1} - {2} - {3} la loai khach hang {4}",
                customer.ValCustomerID, customer.ValCustomerName, customer.ValCustomerAddress, customer.ValCustomerPhone, typeName);
        else
            return " ";
    }
}
```

+ Trong hàm Main, em vẫn giữ các test case tương ứng.

```
static void Main(string [] args)
{
    Company company = new Company();
    company.ValCompanyName = "Vinhomes";
    company.ListOfCustomers.Add(new Customer("C002", "Nguyen Dang Khoi", "Vinhomes Grand Park S7.02", 0987654321, CustomerType.TiemNang);
    company.ListOfCustomers.Add(new Customer("C003", "Tran Hong Tinh", "Vinhomes Grand Park S5.03", 0135792468, CustomerType.CanQuanTa);
    company.ListOfCustomers.Add(new Customer("C004", "KevinSanjaya", "Vinhomes Grand Park S7.02", 0246813579, CustomerType.KhachHangKhac);

    company.CompanyInfo();

    Console.WriteLine("Searching Process:");
    Console.WriteLine(company.SearchCustomer("C002").ConvertToString());
    Console.WriteLine(company.SearchCustomer("C005").ConvertToString());
    Console.WriteLine(company.SearchCustomer(2).ConvertToString());
    Console.WriteLine(company.SearchCustomer(3).ConvertToString());

    Console.ReadKey();
}
```

Màn hình Console:

```
Searching Process:
Khach Hang: C002 - Nguyen Dang Khoi - Vinhomes Grand Park S7.02 - 987654321 la loai khach hang TiemNang
Customer is not found

Khach Hang: C004 - KevinSanjaya - Vinhomes Grand Park S7.02 - 246813579 la loai khach hang KhachHangKhac
Customer is not found
```

Xuất hiện Customer is not found em đã giải thích ở trên tương tự với test case này.

6. Xây dựng một từ điển để định nghĩa các khái niệm loại khách hàng và xuất thông tin đó kèm theo thông tin khách hàng (dùng Dictionary)

+ Trong lớp Company, em sử dụng Dictionary là từ điển gồm chìa khóa và giá trị, để ánh xạ giữa thông tin khách hàng với thông tin chi tiết thêm. Hơn nữa, em dùng Lambda để tìm kiếm thông tin dựa trên khách hàng trong hàm truy xuất thông tin công ty. Khi mà tìm kiếm, thì o (biểu thức lambda) sẽ lần lượt là đối tượng của TypeInfo và nó sẽ kiểm tra khi bằng cái em muốn hay không, nếu bằng thì sẽ trả về o.


```

public class Company
{
    9 references
    public List<Customer> ListOfCustomers { get; set; }
    Dictionary<CustomerType, string> TypeInfo = new Dictionary<CustomerType, string>();
    string CompanyName;

    1 reference
    public Company()
    {
        ListOfCustomers = new List<Customer>();
        CompanyName = "Not Assigned";
        TypeInfo.Add(CustomerType.TiemNang, "Khach hang nay co tiem nang gia han hop dong thue va mua ban nha");
        TypeInfo.Add(CustomerType.CanQuanTam, "Khach hang nay can co nhieu thoi gian de tim hieu de co the thue hoac mua nha");
        TypeInfo.Add(CustomerType.KhachHangKhac, "Khach hang nay khong co du dinh mua hay thue nha");
        TypeInfo.Add(CustomerType.TrungThanh, "Khach hang nay chac chan se thue hoac mua nha");
    }
}

```

Em tiếp tục thêm KeyValuePair vào CompanyInfo

```

1 reference
public void CompanyInfo()
{
    Console.WriteLine("Cong ty {0} co {1} khach hang. Gom cac khach hang nhu sau:\n", CompanyName, ListOfCustomers.Count);
    foreach (Customer rm in ListOfCustomers)
    {
        KeyValuePair<CustomerType, string> info = TypeInfo.FirstOrDefault(o => o.Key == rm.valtype);
        rm.CustomerInfo();
        Console.WriteLine("----Thong Tin Khach Hang: {0}\n", info.Value);
    }
}

```

+ Em thay đổi, xóa searching ở hàm main:

```

namespace CompanyManagement
{
    0 references
    class Program
    {
        0 references
        static void Main(string [] args)
        {
            Company company = new Company();
            company.ValCompanyName = "Vinhomes";
            company.ListOfCustomers.Add(new Customer("C002", "Nguyen Dang Khoi", "Vinhomes Grand Park S7.02", 0987654321, CustomerType.TiemNang);
            company.ListOfCustomers.Add(new Customer("C003", "Tran Hong Tinh", "Vinhomes Grand Park S5.03", 0135792468, CustomerType.CanQuanTa);
            company.ListOfCustomers.Add(new Customer("C004", "KevinSanjaya", "Vinhomes Grand Park S7.02", 0246813579, CustomerType.KhachHangKhac);

            company.CompanyInfo();

            Console.ReadKey();
        }
    }
}

```

Màn hình Console:

```

Cong ty Vinhomes co 3 khách hàng. Gồm các khách hàng như sau:

Khách Hàng C002 có tên là Nguyễn Đăng Khoa, địa chỉ là Vinhomes Grand Park S7.02, số điện thoại là 987654321,
La đang khách hàng TiemNang

----Thông Tin Khách Hàng: Khách hàng này có tiềm năng gia hạn hợp đồng thuê và mua bán nhà

Khách Hàng C003 có tên là Trần Hồng Tĩnh, địa chỉ là Vinhomes Grand Park S5.03, số điện thoại là 135792468,
La đang khách hàng CanQuanTam

----Thông Tin Khách Hàng: Khách hàng này cần có nhiều thời gian để tìm hiểu để có thể thuê hoặc mua nhà

Khách Hàng C004 có tên là KevinSanjaya, địa chỉ là Vinhomes Grand Park S7.02, số điện thoại là 246813579,
La đang khách hàng KháchHangKhac

----Thông Tin Khách Hàng: Khách hàng này không có dự định mua hay thuê nhà

```

7. Tạo một sự kiện để cập nhật số lượng khách hàng khi thêm hoặc xóa khách hàng đó khỏi công ty.

+ Trong lớp hotel, em khai báo delegate CompanyHandler và Event CompanyAddorRemoveEvent. Với CompanyHandler là con trỏ hàm đến những hàm có tham số truyền vào là Company cần cập nhật. Em tiếp tục khai báo thuộc tính NumberOfCustomer với hai phương thức get set và cho giá trị ban đầu = 0, và cuối cùng em thêm 2 thao tác xóa và thêm customer ra khỏi company. Em cập nhật lại hàm companyinfo để xuất thông tin.

```

7 references
public class Company
{
    11 references
    public List<Customer> ListOfCustomers { get; set; }
    Dictionary<CustomerType, string> TypeInfo = new Dictionary<CustomerType, string>();

    public delegate void CompanyHandler (Company company);
    public event CompanyHandler CompanyAddorRemoveEvent;

    string CompanyName;
    int NumberOfCustomers = 0;

    1 reference
    public int ValNumberOfCustomers
    {
        get { return NumberOfCustomers; }
        set { NumberOfCustomers = value; }
    }
}

```

```

1 reference
public void CompanyInfo()
{
    Console.WriteLine("Cong ty {0} co {1} khách hàng. Gồm các khách hàng như sau:\n", CompanyName, NumberOfCustomers);
    foreach (Customer rm in ListOfCustomers)
    {
        KeyValuePair<CustomerType, string> info = TypeInfo.FirstOrDefault(o => o.Key == rm.valtype);
        rm.CustomerInfo();
        Console.WriteLine("----Thông Tin Khách Hàng: {0}\n", info.Value);
    }
}

```

```

0 references
public void AddCustomer(Customer customer)
{
    ListOfCustomers.Add(customer);
    OnCompanyChanger(this);
}

1 reference
public void RemoveCustomer(Customer customer)
{
    ListOfCustomers.Remove(customer);
    OnCompanyChanger(this);
}

2 references
public void OnCompanyChanger(Company company)
{
    if (CompanyAddorRemoveEvent != null)
    {
        CompanyAddorRemoveEvent(this);
    }
}

1 reference

```

+ Em cập nhật lại hàm Main để xóa thông tin ID C002 và thêm hàm UpdateNumberOfCustomers

```

class Program
{
    1 reference
    static void UpdateNumberOfCustomers(Company company)
    {
        company.ValNumberOfCustomers = company.ListOfCustomers.Count;
    }
    0 references
    static void Main(string [] args)
    {
        Company company = new Company();
        company.ValCompanyName = "Vinhomes";

        company.CompanyAddorRemoveEvent += new Company.CompanyHandler(UpdateNumberOfCustomers);

        company.ListOfCustomers.Add(new Customer("C002", "Nguyen Dang Khoi", "Vinhomes Grand Park S7.02", 0987654321, CustomerType.TiemNang);
        company.ListOfCustomers.Add(new Customer("C003", "Tran Hong Tinh", "Vinhomes Grand Park S5.03", 0135792468, CustomerType.CanQuanTam);
        company.ListOfCustomers.Add(new Customer("C004", "KevinSanjaya", "Vinhomes Grand Park S7.02", 0246813579, CustomerType.KhachHangKhac);

        Customer customer = company.SearchCustomer("C002");
        company.RemoveCustomer(customer);

        company.CompanyInfo();

        Console.ReadKey();
    }
}

```

Màn hình Console:

```

Cong ty Vinhomes co 2 khách hàng. Gom cac khách hàng như sau:

Khách Hàng C003 có tên là Tran Hong Tinh, địa chỉ là Vinhomes Grand Park S5.03, số điện thoại là 135792468,
La dang khách hàng CanQuanTam

----Thông Tin Khách Hàng: Khách hàng này cần có nhiều thời gian để tìm hiểu để có thể thuê hoặc mua nhà

Khách Hàng C004 có tên là KevinSanjaya, địa chỉ là Vinhomes Grand Park S7.02, số điện thoại là 246813579,
La dang khách hàng KháchHangKhac

----Thông Tin Khách Hàng: Khách hàng này không có dự định mua hay thuê nhà

```

Event là liên quan đến sự kiện. Một khi event diễn ra, CompanyHandler sẽ liên kết với 1 hàm nào đó có tham số truyền vào là company và trả về là void. Khi thêm customer vô, sự kiện OnCompanyChanger phải thực thi, nó sẽ gọi sự kiện mình đăng ký là CompanyAddorRemoveEvent, sự kiện thực thi sẽ yêu cầu con trỏ hàm CompanyHandler thực hiện 1 hàm nào đó. CompanyHandler sẽ thực hiện hàm UpdateNumberOfCustomers .

