

21110324 - Lương Đăng Khôi

Màn hình menu:

```
E:\dotNET Programming\CustomerCompany\bin\Debug\net8.0\CustomerCompany.exe
Default customers added successfully.

--- Company Customer Management ---
1. Add Customer
2. Remove Customer
3. Display Company Info
4. Search Customer
5. Exit
Choose an option:
```

```
Select E:\dotNET Programming\CustomerCompany\bin\Debug\net8.0\CustomerCompany.exe
2. Remove Customer
3. Display Company Info
4. Search Customer
5. Exit
Choose an option: 3
Company: Tech Solutions
Customers: 5
Info of Customer ID: C001
Name: Alice Nguyen
Address: 123 Main St
Phone: C021
Type: Khách hàng trung thành

Info of Customer ID: C002
Name: Bob Tran
Address: 456 Oak Ave
Phone: C123
Type: Khách hàng tiệm nail

Info of Customer ID: C003
Name: Charlie Do
Address: 789 Pine Rd
Phone: C095
Type: Khách hàng cần quan tâm

Info of Customer ID: C004
Name: Daisy Pham
Address: 101 Maple St
Phone: C956
Type: Khách hàng khác

Info of Customer ID: C005
Name: Ethan Vu
Address: 202 Birch Blvd
Phone: C033
Type: Khách hàng trung thành
```

Source code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CustomerCompany
{
    using MainData;

    public class Program
    {
        static void Main(string[] args)
        {
            Company company = new Company("Tech Solutions");
            AddDefaultCustomers(company);

            bool isRunning = true;
            while (isRunning)
            {
                Console.WriteLine("\n--- Company Customer Management ---");
                Console.WriteLine("1. Add Customer");
                Console.WriteLine("2. Remove Customer");
                Console.WriteLine("3. Display Company Info");
                Console.WriteLine("4. Search Customer");
                Console.WriteLine("5. Exit");
                Console.Write("Choose an option: ");

                string choice = Console.ReadLine();

                switch (choice)
                {
```

```

case "1":
    var customer = new Customer();
    customer.CustomerInput();
    company.AddCustomer(customer);
    break;
case "2":
    Console.Write("Enter Customer ID to remove: ");
    string customerId = Console.ReadLine();
    var toRemove = company.ListOfCustomers.FirstOrDefault(c => c.CustomerID
== customerId);
    if (toRemove != null)
    {
        company.RemoveCustomer(toRemove);
        Console.WriteLine("Customer removed.");
    }
    else
    {
        Console.WriteLine("Customer not found.");
    }
    break;
case "3":
    company.CompanyInfo();
    break;
case "4":
    Console.Write("Enter Customer Name to search: ");
    string name = Console.ReadLine();
    var result = company.SearchCustomer(name);
    if (result is Customer foundCustomer)
    {
        foundCustomer.CustomerInfo();
    }
    else if (result is List<Customer> foundList)
    {

```

```

        foreach (var c in foundList)
        {
            c.CustomerInfo();
            Console.WriteLine();
        }
    }
    else
    {
        Console.WriteLine("Customer not found.");
    }
    break;
case "5":
    isRunning = false;
    Console.WriteLine("Exiting program.");
    break;
default:
    Console.WriteLine("Invalid option. Try again.");
    break;
}
}
}

```

```

static void AddDefaultCustomers(Company company)
{
    var defaultCustomers = new List<Customer>
    {
        new Customer("C001", "Alice Nguyen", "123 Main St", "C021",
customerType.TrungThanh),
        new Customer("C002", "Bob Tran", "456 Oak Ave", "C123",
customerType.TiemNang),
        new Customer("C003", "Charlie Do", "789 Pine Rd", "C095",
customerType.CanQuanTam),
        new Customer("C004", "Daisy Pham", "101 Maple St", "C956",
customerType.KhachHangKhac),
    }
}

```

```
        new Customer("C005", "Ethan Vu", "202 Birch Blvd", "C033",  
customertype.TrungThanh)  
    };
```

```
    foreach (var customer in defaultCustomers)  
    {  
        company.AddCustomer(customer);  
    }
```

```
        Console.WriteLine("Default customers added successfully.");  
    }  
}  
}
```

```
namespace MainData
```

```
{  
    public enum customertype { TrungThanh, TiemNang, CanQuanTam, KhachHangKhac };
```

```
    public class Customer
```

```
    {  
        public string CustomerID { get; set; }  
        public string CustomerName { get; set; }  
        public string CustomerAddress { get; set; }  
        public string CustomerPhone { get; set; }  
        public customertype CustomerType = customertype.KhachHangKhac;
```

```
        public static readonly Dictionary<customertype, string> CustomerTypeDescriptions =  
new Dictionary<customertype, string>
```

```
    {  
        { customertype.TrungThanh, "Khach hang trung thanh" },  
        { customertype.TiemNang, "Khach hang tiem nang" },  
        { customertype.CanQuanTam, "Khach hang can quan tam" },  
        { customertype.KhachHangKhac, "Khach hang khac" }
```

```
};
```

```
public Customer() { }
```

```
public Customer(string id, string name, string address, string phone, customertype type)
```

```
{
```

```
    CustomerID = id;
```

```
    CustomerName = name;
```

```
    CustomerAddress = address;
```

```
    CustomerPhone = phone;
```

```
    CustomerType = type;
```

```
}
```

```
public void CustomerInput()
```

```
{
```

```
    Console.WriteLine("Enter Customer's information: ");
```

```
    Console.Write("ID: "); CustomerID = Console.ReadLine();
```

```
    Console.Write("Name: "); CustomerName = Console.ReadLine();
```

```
    Console.Write("Address: "); CustomerAddress = Console.ReadLine();
```

```
    Console.Write("Phone: "); CustomerPhone = Console.ReadLine();
```

```
}
```

```
public void CustomerInfo()
```

```
{
```

```
    string typeDescription = CustomerTypeDescriptions.ContainsKey(CustomerType)
```

```
        ? CustomerTypeDescriptions[CustomerType]
```

```
        : "Khong xac dinh";
```

```
    Console.WriteLine($"Info of Customer ID: {CustomerID}\n" +
```

```
        $"Name: {CustomerName}\n" +
```

```
        $"Address: {CustomerAddress}\n" +
```

```
        $"Phone: {CustomerPhone}\n" +
```

```
        $"Type: {typeDescription}");
```

```
}
```

```
public string ValCusID
{
    get { return CustomerID; }
    set { CustomerID = value; }
}

public string ValCusName
{
    get { return CustomerName; }
    set { CustomerName = value; }
}

public string ValCusAddress
{
    get { return CustomerAddress; }
    set { CustomerAddress = value; }
}

public string ValCusPhone
{
    get { return CustomerPhone; }
    set { CustomerPhone = value; }
}

public customertype ValCusType
{
    get { return CustomerType; }
    set { CustomerType = value; }
}
}
```

```

public class Company
{
    public string CompanyName { get; set; }
    public List<Customer> ListOfCustomers { get; set; } = new List<Customer>();

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

    public int NumberOfCustomer => ListOfCustomers.Count;

    public Company() { }
    public Company(string CompanyName)
    {
        this.CompanyName = CompanyName;
    }

    public void CompanyInfo()
    {
        Console.WriteLine($"Company: {CompanyName}");
        Console.WriteLine($"Customers: {NumberOfCustomer}");

        int count = Math.Min(ListOfCustomers.Count, 5);
        for (int i = 0; i < count; i++)
        {
            ListOfCustomers[i].CustomerInfo();
            Console.WriteLine();
        }

        if (ListOfCustomers.Count > 5)
        {
            Console.WriteLine("...and more.");
        }
    }
}

```



```
}
```

```
public dynamic SearchCustomer<T>(T searchTerm)
```

```
{
```

```
    List<Customer> foundCustomers = new List<Customer>();
```

```
    if (searchTerm is string name)
```

```
    {
```

```
        foundCustomers = ListOfCustomers
```

```
            .Where(customer => customer.CustomerName.Equals(name,  
StringComparison.OrdinalIgnoreCase))
```

```
            .ToList();
```

```
    }
```

```
    if (foundCustomers.Count == 1)
```

```
    {
```

```
        return foundCustomers[0];
```

```
    }
```

```
    else if (foundCustomers.Count > 1)
```

```
    {
```

```
        return foundCustomers;
```

```
    }
```

```
    return null;
```

```
}
```

```
public void OnCompanyChanger(Company company)
```

```
{
```

```
    CompanyAddorRemoveEvent?.Invoke(this);
```

```
}
```

```
public void AddCustomer(Customer customer)
```

```
{
```

```
        ListOfCustomers.Add(customer);  
        OnCompanyChanger(this);  
    }  
  
    public void RemoveCustomer(Customer customer)  
    {  
        ListOfCustomers.Remove(customer);  
        OnCompanyChanger(this);  
    }  
}  
}
```