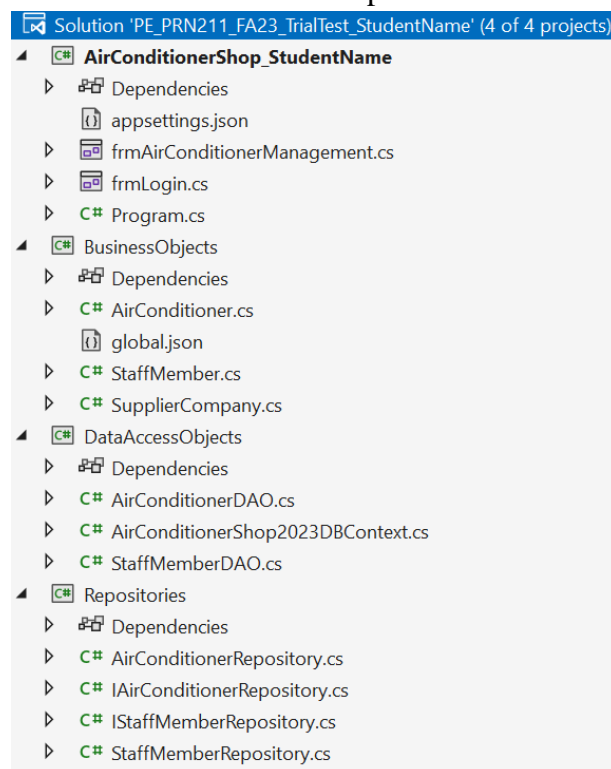


Khung Solution - Tạo 3 Project Class Library, 1 Project Windows Form

- BusinessObjects – Đối tượng nghiệp vụ
- DataAccessObjects - references BusinessObjects
- Repositories - references DataAccessObjects
- AirConditionerShop\_StudentName - references Repositories



## Bước 0. Kiểm tra lại môi trường chạy của máy cá nhân

Kiểm tra tool EFCore và uninstall/install tool

```
dotnet tool uninstall --global dotnet-ef
```

```
dotnet tool install --global dotnet-ef --version 6.0.22 (dotnet SDK dùng trong bài này là 6.0.405)
```

Trường hợp chuyển đổi version của dotnet SDK

dotnet new globaljson → sửa trong tập tin tập tin *global.json* version thành 6.0.405

Chạy Script *AirConditionerShop2023DB.sql* → tạo Database MyStore trong SQL Server

- Server = local
- Database = **AirConditionerShop2023DB**
- Uid = sa
- Pwd =123

→ Chuỗi kết nối tới CSDL – **Connection String**:

**“Server=(local); Database=AirConditionerShop2023DB; Uid=sa; Pwd=123”**

## Bước 1. Thao tác với Project Class Library: BusinessObjects

- Khai báo các thư viện NuGet packages trong Project **BusinessObjects** (dùng Developer PowerShell, có thể dùng NuGet Package Manager nhưng lưu ý version phù hợp với máy tính cá nhân)

```
dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 6.0.22
```

```
dotnet add package Microsoft.EntityFrameworkCore.Tools --version 6.0.22
```

```
dotnet add package Microsoft.Extensions.Configuration.Json --version 6.0.0
```

- Thực hiện ORM (Database First, từ cơ sở dữ liệu có sẵn, phát sinh các lớp Model)

```
dotnet ef dbcontext scaffold "server =(local); database =  
AirConditionerShop2023DB;uid=sa;pwd=123;TrustServerCertificate=True;“  
Microsoft.EntityFrameworkCore.SqlServer --output-dir Models
```

- Thay đổi chuỗi kết nối hardcode

```
using System.IO;
```

```
using Microsoft.Extensions.Configuration.Json;
```

```
private string GetConnectionString()
```

```
{
```

```
    IConfiguration configuration = new ConfigurationBuilder()
```

```
        .SetBasePath(Directory.GetCurrentDirectory())
```

```
        .AddJsonFile("appsettings.json", true, true).Build();
```

```
    return configuration["ConnectionStrings:DBDefault"];
```

```
}
```

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

```
{
```

```
    optionsBuilder.UseSqlServer(GetConnectionString());
```

```
}
```

- Chuyển file *AirConditionerShop2023DBContext.cs* sang Project *DataAccessObjects*. Lưu ý đổi tên namespace trong file *AirConditionerShop2023DBContext.cs*.

## Bước 2. Thao tác với Project Class Library: *DataAccessObjects*

- Reference tới Project *BusinessObjects*
- Viết lớp thao tác CRUD với bảng *AirConditioner*: *AirConditionerDAO.cs*, và lớp *StaffMemberDAO.cs* kiểm tra phân quyền chức năng.

Lớp thao tác với bảng *StaffMember*, kiểm tra tài khoảnLogin *StaffMemberDAO.cs*  
`using BusinessObjects;`

```
namespace DataAccessObjects
{
    5 references
    public class StaffMemberDAO
    {
        private static StaffMemberDAO instance = null;
        private static object lockObject = new object();

        1 reference
        private StaffMemberDAO() { }
        1 reference
        public static StaffMemberDAO Instance
        {
            get
            {
                if(instance == null)
                {
                    instance = new StaffMemberDAO();
                }
                return instance;
            }
        }

        1 reference
        public StaffMember CheckLogin(string email, string password)...
```

Trong đó hàm *CheckLogin()*

```
public StaffMember CheckLogin(string email, string password)
{
    using var db = new AirConditionerShop2023DBContext();
    return db.StaffMembers.SingleOrDefault(m => m.EmailAddress.Equals(email) &&
m.Password.Equals(password));
}
```

Lớp thao tác với bảng *AirConditioner*, thực hiện *CRUD* *AirConditionerDAO.cs*

```
using BusinessObjects;
using Microsoft.EntityFrameworkCore;

namespace DataAccessObjects
{
    10 references
    public class AirConditionerDAO
    {
        private static AirConditionerDAO instance = null;
        private static readonly object objLock = new object();
        1 reference
        private AirConditionerDAO() { }
        6 references
        public static AirConditionerDAO Instance ...
        2 references
        public List<AirConditioner> GetAirConditioners() ...
        1 reference
        public AirConditioner GetAirConditionerById(int? id) ...
        1 reference
        public List<SupplierCompany> GetSupplierCompanies() ...
        1 reference
        public void SaveAirConditioner(AirConditioner a) ...
        1 reference
        public void UpdateAirConditioner(AirConditioner a) ...
        1 reference
        public void DeleteAirConditioner(AirConditioner a) ...
    }
}
```

Chi tiết các hàm và khai báo Property (for Singleton Pattern)

```
public static AirConditionerDAO Instance
{
    get
    {
        lock (objLock)
        {
            if(instance == null)
            {
                instance = new AirConditionerDAO();
            }
            return instance;
        }
    }
}
```

Lấy danh sách

```
public List<AirConditioner> GetAirConditioners()
{
    using var db = new AirConditionerShop2023DBContext();
    return db.AirConditioners.Include(f=>f.Supplier).ToList();
}
```

Lấy Air Conditioner theo ID

```

public AirConditioner GetAirConditionerById(int? id)
{
    var temp = GetAirConditioners()
        .SingleOrDefault(m => m.AirConditionerId == id);
    return temp;
}

```

Lấy danh sách các Supplier Company

```

public List<SupplierCompany> GetSupplierCompanies()
{
    using var db = new AirConditionerShop2023DBContext();
    return db.SupplierCompanies.ToList();
}

```

Lưu Air Conditioner

```

public void SaveAirConditioner(AirConditioner a)
{
    using var db = new AirConditionerShop2023DBContext();
    db.AirConditioners.Add(a);
    db.SaveChanges();
}

```

Cập nhật Air Conditioner

```

public void UpdateAirConditioner(AirConditioner a)
{
    using var db = new AirConditionerShop2023DBContext();
    db.AirConditioners.Update(a);
    db.SaveChanges();
}

```

Xóa Air Conditioner

```

public void DeleteAirConditioner(AirConditioner a)
{
    using var db = new AirConditionerShop2023DBContext();
    var a1 = db.AirConditioners
        .SingleOrDefault(m => m.AirConditionerId == a.AirConditionerId);
    db.AirConditioners.Remove(a1);
    db.SaveChanges();
}

```

### Bước 3. Thao tác với Project Class Library: Repositories

- Reference tới DataAccessObjects, BusinessObjects
- Tạo Interface để giao tiếp: *IAirConditionerRepository.cs*, *IStaffMemberRepository.cs*

Interface *IAirConditionerRepository*

`using BusinessObjects;`

`namespace Repositories`

`{`

2 references

`public interface IAirConditionerRepository`

`{`

2 references

`List<AirConditioner> GetAirConditioners();`

2 references

`List<SupplierCompany> GetSupplierCompanies();`

1 reference

`AirConditioner GetAirConditionerById(int id);`

2 references

`void DeleteAirConditioner(AirConditioner airConditioner);`

2 references

`void UpdateAirConditioner(AirConditioner airConditioner);`

2 references

`void SaveAirConditioner(AirConditioner airConditioner);`

`}`

`}`

Interface *IStaffMemberRepository*

`using BusinessObjects;`

`namespace Repositories`

`{`

2 references

`public interface IStaffMemberRepository`

`{`

2 references

`StaffMember CheckLogin(string username, string password);`

`}`

`}`

- Tạo lớp implements Interface: AirConditionerRepository.cs, StaffMemberRepository.cs

Lớp AirConditionerRepository

```
using BusinessObjects;
using DataAccessObjects;
```

```
namespace Repositories
{
    1 reference
    public class AirConditionerRepository : IAirConditionerRepository
    {
        2 references
        public List<AirConditioner> GetAirConditioners()
            => AirConditionerDAO.Instance.GetAirConditioners();
        2 references
        public List<SupplierCompany> GetSupplierCompanies()
            => AirConditionerDAO.Instance.GetSupplierCompanies();
        1 reference
        public AirConditioner GetAirConditionerById(int id)
            => AirConditionerDAO.Instance.GetAirConditionerById(id);
        2 references
        public void DeleteAirConditioner(AirConditioner airConditioner)
            => AirConditionerDAO.Instance.DeleteAirConditioner(airConditioner );
        2 references
        public void UpdateAirConditioner(AirConditioner airConditioner)
            => AirConditionerDAO.Instance.UpdateAirConditioner(airConditioner);
        2 references
        public void SaveAirConditioner(AirConditioner airConditioner)
            => AirConditionerDAO.Instance.SaveAirConditioner(airConditioner);
    }
}
```

Lớp StaffMemberRepository chỉ có xử lý hàm kiểm tra login gọi đến DAO

```
using BusinessObjects;
using DataAccessObjects;
```

```
namespace Repositories
{
    1 reference
    public class StaffMemberRepository: IStaffMemberRepository
    {
        2 references
        public StaffMember CheckLogin(string username, string password)
            => StaffMemberDAO.Instance.CheckLogin(username, password);
    }
}
```

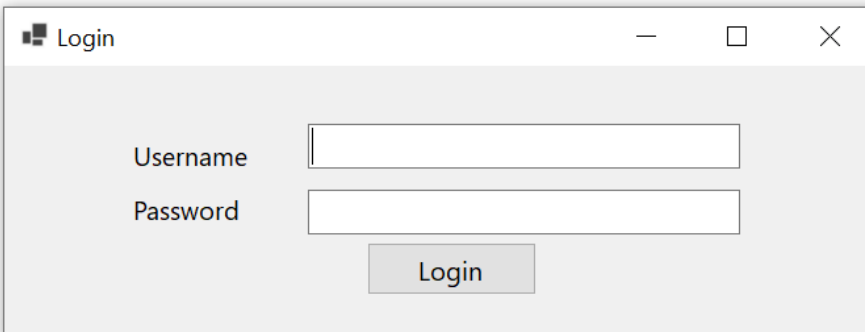
#### Bước 4. Thực hiện với Project WindowsForms AirConditionerShop\_StudentName

- References Project: Repositories, BusinessObjects
- Tạo tập tin appsettings.json - khai báo chuỗi kết nối- thay đổi CopyToOutputDirectory= **Copy Always**.  
Nội dung tập tin:

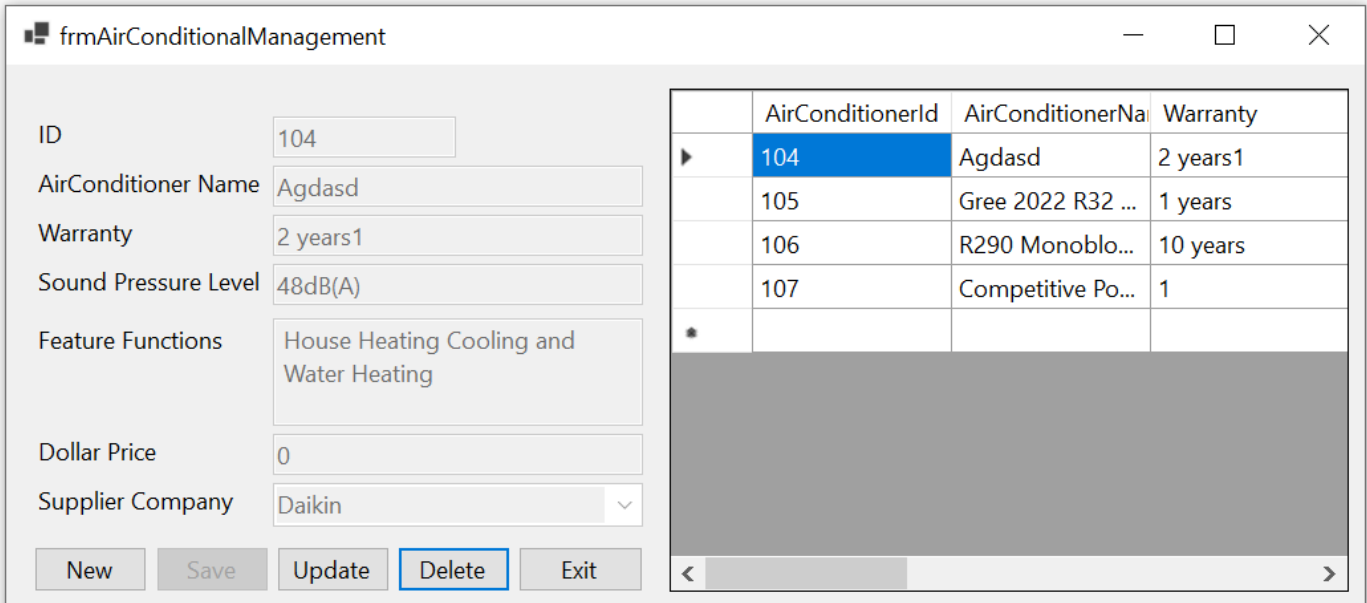
```
{
  "ConnectionStrings": {
    "DBDefault": "Server=local;Database=AirConditionerShop2023DB;Uid=sa;Pwd=123"
```

```
}  
}
```

- Tạo form cho Login
- Tạo form cho Air Conditioner Management



A screenshot of a Windows-style application window titled "Login". It contains two text input fields: "Username" and "Password". Below the "Password" field is a "Login" button.



A screenshot of a Windows-style application window titled "frmAirConditionalManagement". The window is divided into two main sections. On the left is a form with several input fields and buttons. On the right is a data grid displaying a list of air conditioners.

	AirConditionerId	AirConditionerNa	Warranty
▶	104	Agdasd	2 years1
	105	Gree 2022 R32 ...	1 years
	106	R290 Monoblo...	10 years
	107	Competitive Po...	1
*			

The left form contains the following fields and buttons:

- ID: 104
- AirConditioner Name: Agdasd
- Warranty: 2 years1
- Sound Pressure Level: 48dB(A)
- Feature Functions: House Heating Cooling and Water Heating
- Dollar Price: 0
- Supplier Company: Daikin (dropdown menu)
- Buttons: New, Save, Update, Delete (highlighted with a blue border), Exit



## Bước 5. Thực hiện các chức năng Login, List, Update, Create, Delete

Xử lý form Login

using Repositories;

namespace AirConditionerShop\_StudentName

```
{  
    3 references  
    public partial class frmLogin : Form  
    {  
        IStaffMemberRepository _staffRepository = new StaffMemberRepository();  
        1 reference  
        public frmLogin()  
        {  
            InitializeComponent();  
        }  
        1 reference  
        private void btnLogin_Click(object sender, EventArgs e)  
        {  
            var tmp = _staffRepository.CheckLogin(txtUsername.Text, txtPassword.Text);  
            if (tmp != null && tmp.Role == 1)  
            {  
                frmAirConditionerManagement f = new frmAirConditionerManagement();  
                f.ShowDialog(); this.Hide();  
            }  
            else  
                MessageBox.Show("You have no permission to do this function!",  
                                "Air Conditioner Management");  
        }  
    }  
}
```

## Xử lý form Air Conditioner Management

```
using BusinessObjects;
using Repositories;
namespace AirConditionerShop_StudentName
{
    4 references
    public partial class frmAirConditionerManagement : Form
    {
        1 reference
        IAirConditionerRepository repo = new AirConditionerRepository();
        bool CreateOrUpdate = false;
        1 reference
        public frmAirConditionerManagement()...
        1 reference
        private void frmAirConditionalManagement_Load(object sender, EventArgs e)
        1 reference
        public void LoadSupplierCompanyList()...
        5 references
        public void LoadAirConditionerList()...
        1 reference
        private void btnDelete_Click(object sender, EventArgs e)
        1 reference
        private void btnNew_Click(object sender, EventArgs e)
        1 reference
        private void ClearText()...
        6 references
        private void EnableText(bool status)
        1 reference
        private void btnSave_Click(object sender, EventArgs e)
        1 reference
        private void btnUpdate_Click(object sender, EventArgs e)
        1 reference
        private void frmAirConditionerManagement_FormClosing(object sender, Fo
        1 reference
        private void btnExit_Click(object sender, EventArgs e)
    }
}
```

Xử lý sự kiện Load của Form, lấy thông tin hiển thị vào DataGridView, ComboBox và Disable các TextBox

```
private void frmAirConditionalManagement_Load(object sender, EventArgs e)
{
    LoadSupplierCompanyList();
    LoadAirConditionerList();
    EnableText(false);
}
1 reference
public void LoadSupplierCompanyList()
{
    try
    {
        var categoryList = repo.GetSupplierCompanies();
        cboSupplierCompany.DataSource = categoryList;
        cboSupplierCompany.DisplayMember = "SupplierName";
        cboSupplierCompany.ValueMember = "SupplierId";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error on load list of categories");
    }
}
```

Load danh sách Air Conditioner vào DataGridView, bind các TextBox với các thuộc tính dùng Binding Source

```
public void LoadAirConditionerList()
{
    try
    {
        var aList = repo.GetAirConditioners();
        BindingSource source = new BindingSource();
        source.DataSource = aList;

        txtAirConditionerID.DataBindings.Clear();
        txtAirConditionerName.DataBindings.Clear();
        txtWarranty.DataBindings.Clear();
        txtSound.DataBindings.Clear();
        txtFeatureFunction.DataBindings.Clear();
        txtPrice.DataBindings.Clear();
        cboSupplierCompany.DataBindings.Clear();

        txtAirConditionerID.DataBindings.Add("Text", source, "AirConditionerID");
        txtAirConditionerName.DataBindings.Add("Text", source, "AirConditionerName");
        txtWarranty.DataBindings.Add("Text", source, "Warranty");
        txtSound.DataBindings.Add("Text", source, "SoundPressureLevel");
        txtFeatureFunction.DataBindings.Add("Text", source, "FeatureFunction");
        txtPrice.DataBindings.Add("Text", source, "DollarPrice");
        cboSupplierCompany.DataBindings.Add("Text", source, "Supplier.SupplierName");

        dgvList.DataSource = null;
        dgvList.DataSource = source;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error on load list of air conditioners");
    }
}
```

Xử lý nút xóa dữ liệu, dùng MessageBox hiển thị confirm dialog, trường hợp nhấn OK mới thật sự xóa dữ liệu

```
private void btnDelete_Click(object sender, EventArgs e)
{
    DialogResult d;
    d = MessageBox.Show("Do you really want to delete this record?", "Air Conditioner Management",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button1);

    if (d == DialogResult.OK)
    {
        var airConditioner = new AirConditioner
        {
            AirConditionerId = int.Parse(txtAirConditionerID.Text),
        };
        repo.DeleteAirConditioner(airConditioner);
        LoadAirConditionerList();
    }
}
```

Xử lý nút tạo mới, enable các TextBox, đổi caption của nút thành “Cancel”, bật “Save” để lưu dữ liệu

```
private void btnNew_Click(object sender, EventArgs e)
{
    if (btnNew.Text == "New")
    {
        btnNew.Text = "Cancel";
        btnSave.Enabled = true;
        EnableText(true);

        // Clear DataBinding
        txtAirConditionerID.DataBindings.Clear();
        txtAirConditionerName.DataBindings.Clear();
        txtWarranty.DataBindings.Clear();
        txtSound.DataBindings.Clear();
        txtFeatureFunction.DataBindings.Clear();
        txtPrice.DataBindings.Clear();
        cboSupplierCompany.DataBindings.Clear();
        dgvList.ClearSelection();

        ClearText();
        CreateOrUpdate = true;
    }
    else
    {
        btnNew.Text = "New";
        btnSave.Enabled = false;
        EnableText(false);
        LoadAirConditionerList();
    }
}
```

Nút Save dùng cho 2 chức năng New và Update, chỉ kiểm tra “All fields are required”

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (txtAirConditionerID.Text.Trim() == "" || txtAirConditionerName.Text.Trim() == "" ||
        txtWarranty.Text.Trim() == "" || txtSound.Text.Trim() == "" ||
        txtFeatureFunction.Text.Trim() == "" || txtPrice.Text.Trim() == "")
    {
        MessageBox.Show("All fields are required!", "Air Conditioner Management System",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        var airConditioner = new AirConditioner
        {
            AirConditionerId = int.Parse(txtAirConditionerID.Text),
            AirConditionerName = txtAirConditionerName.Text,
            Warranty = txtWarranty.Text,
            SoundPressureLevel = txtSound.Text,
            FeatureFunction = txtFeatureFunction.Text,
            DollarPrice = int.Parse(txtPrice.Text),
            SupplierId = cboSupplierCompany.SelectedValue.ToString()
        };
        if (CreateOrUpdate) // create a new air conditioner
        {
            repo.SaveAirConditioner(airConditioner);
            btnNew.Text = "New";
        }
        else
        {
            repo.UpdateAirConditioner(airConditioner);
            btnUpdate.Text = "Update";
        }
        EnableText(false);
        btnSave.Enabled = false;
        LoadAirConditionerList();
    }
}
```

Xóa empty các TextBox, ComboBox chọn về phần tử đầu tiên

```
private void ClearText()
{
    txtAirConditionerID.Text = "";
    txtAirConditionerName.Text = "";
    txtWarranty.Text = "";
    txtSound.Text = "";
    txtFeatureFunction.Text = "";
    txtPrice.Text = "";
    cboSupplierCompany.SelectedIndex = 0;
}
```

Enable/Disable các TextBox tùy tham số truyền vào

```
private void EnableText(bool status)
{
    txtAirConditionerID.Enabled = status;
    txtAirConditionerName.Enabled = status;
    txtWarranty.Enabled = status;
    txtSound.Enabled = status;
    txtFeatureFunction.Enabled = status;
    txtPrice.Enabled = status;
    cboSupplierCompany.Enabled = status;
}
```

Xử lý nút tạo mới, enable các TextBox, đổi caption của nút thành “Cancel”, bật “Save” để lưu dữ liệu

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    EnableText(true);
    txtAirConditionerID.Enabled = false;

    btnUpdate.Text = "Cancel";
    btnSave.Enabled = true;
    CreateOrUpdate = false;
}
```

Xử lý sự kiện Form\_Closing mở hộp thoại xác nhận

```
private void frmAirConditionerManagement_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult d;
    d = MessageBox.Show("Are you sure you want to exit?", "Air Conditioner Management",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button1);
    if (d == DialogResult.Cancel)
    {
        e.Cancel = true;
    }
}
```

Xử lý nút “Exit”

```
private void btnExit_Click(object sender, EventArgs e)
{
    Close();
}
```