KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÁC HỆ THỐNG THÔNG TIN

Chương 1. Môi trường lập trình

Hoàng Hữu Việt - IT Faculty, Vinh University
Email: viethh.vinhuni@gmail.com

Nội dung

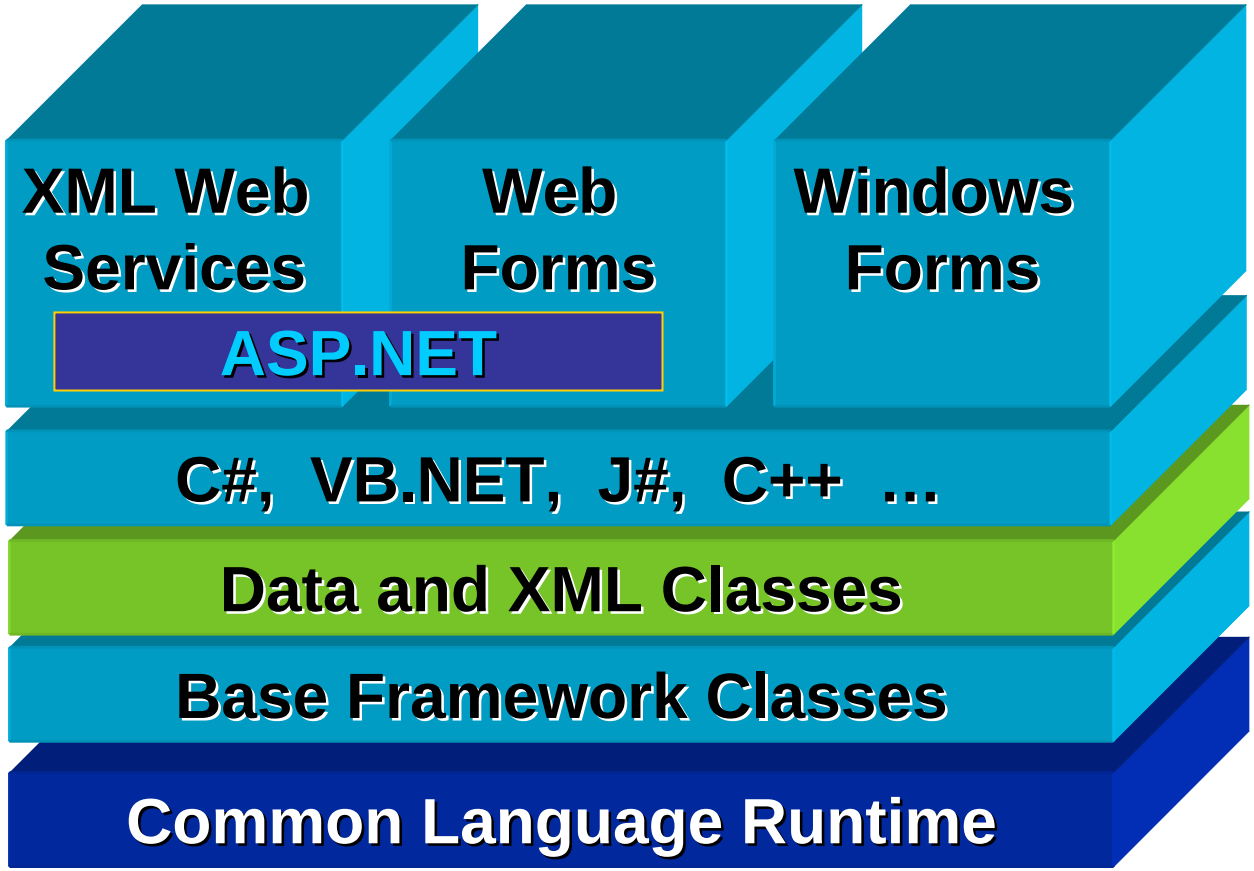
- Giới thiệu
- Môi trường lập trình
- Ví dụ
- Một số khái niệm

Giới thiệu

- Visual Studio .NET
  - Môi trường phát triển tích hợp của Microsoft
  - Chương trình viết trong nhiều ngôn ngữ khác nhau
    - **Visual C#**
    - **Visual C++**
    - **VB.NET**
  - Được công bố vào tháng 7 năm 2000
- Visual C#
  - Là ngôn ngữ hướng đối tượng, lập trình trực quan và dựa trên ngôn ngữ C, C++ và Java

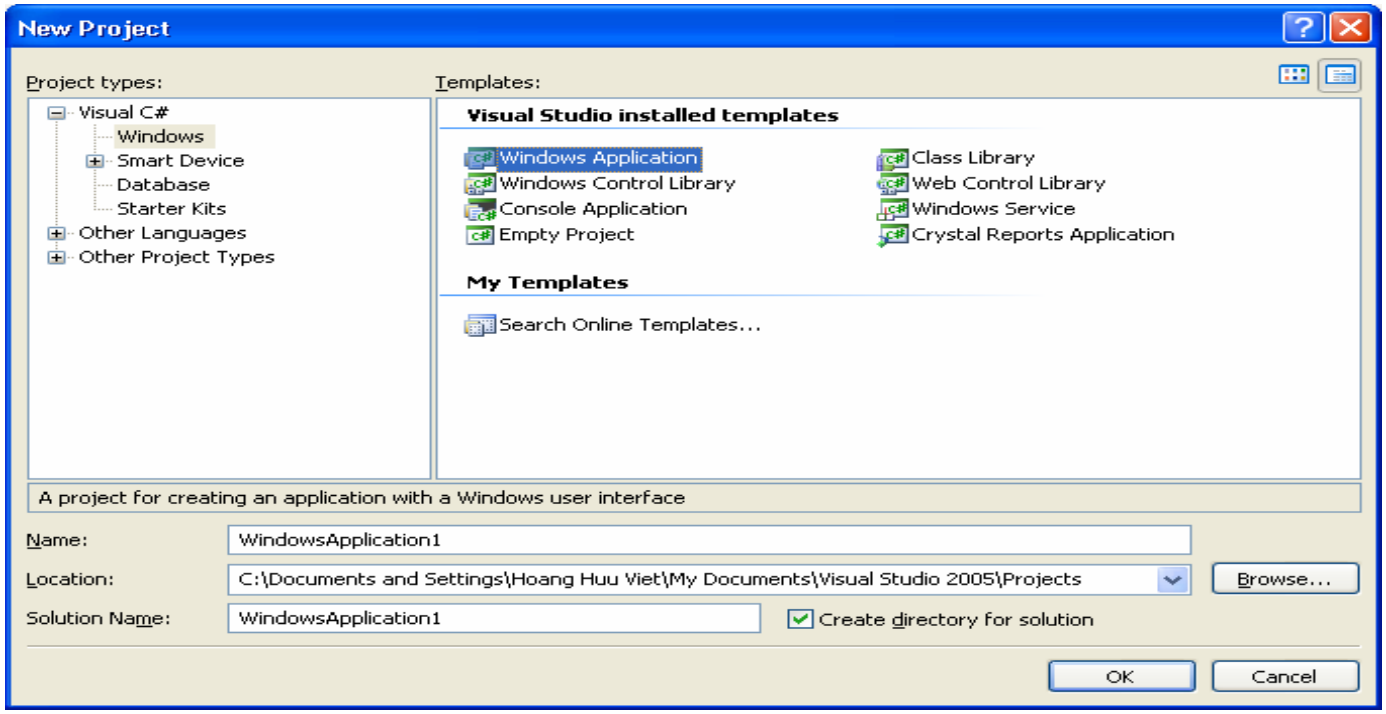
Giới thiệu

- Microsoft.NET Framework



Môi trường lập trình .NET

- Tạo một Project
  - Chọn **File → New → Project**



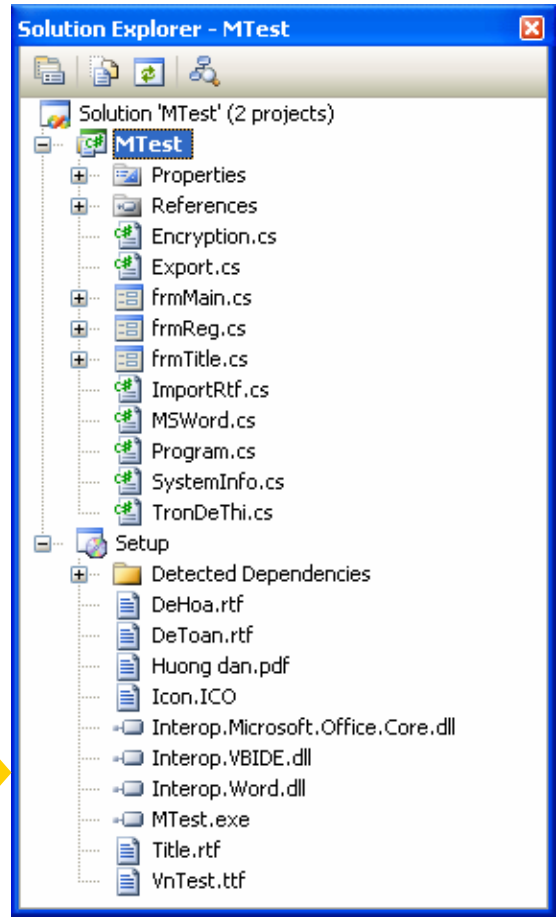
Môi trường lập trình .NET

- Tạo một Project
  - Chọn Visual C# → Windows
  - Chọn Windows Application
  - Chọn Location
  - Gõ tên Project
  - Chú ý: Ngầm định tự động tạo ra thư mục chứa Project

Môi trường lập trình .NET

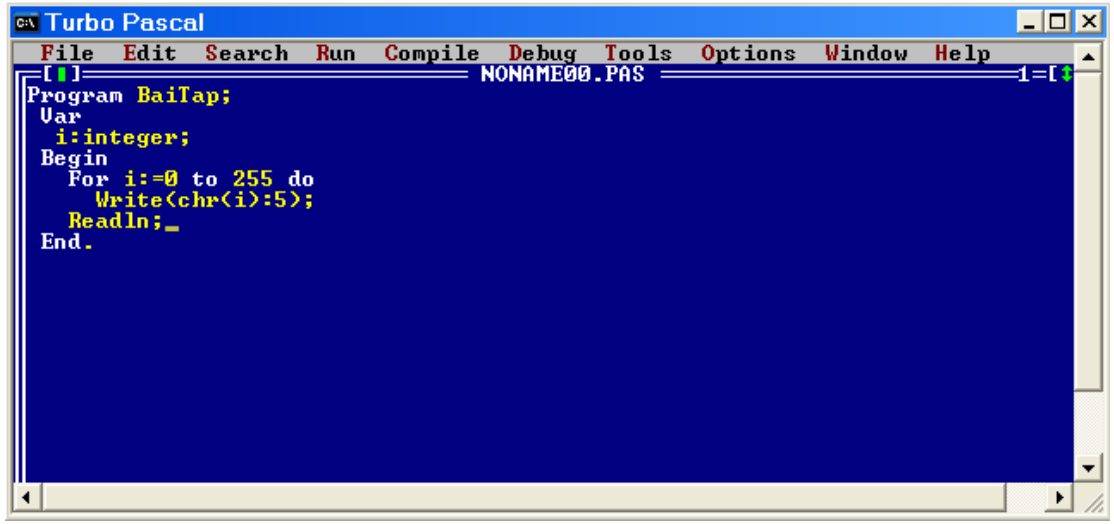
- Một số khái niệm
  - C# .NET Project
    - Tập hợp và tệp, hình ảnh, tài liệu của Project
  - C# .NET solution
    - Tập hợp các Project tạo ra một hoặc nhiều ứng dụng

Một Solution có 2 Project



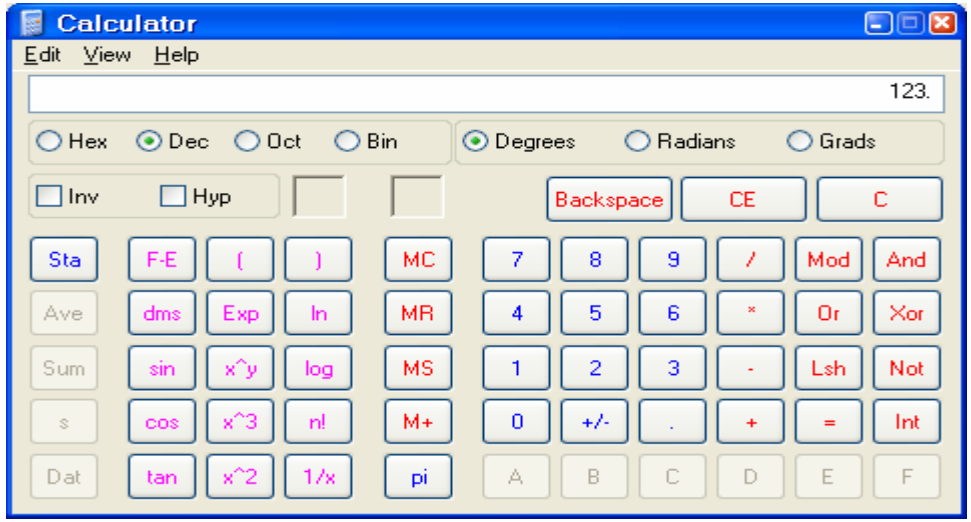
Môi trường lập trình .NET

- Một số khái niệm
  - Ứng dụng Console applications
    - Chạy trên môi trường MS DOS Prompt của Window
    - Không có giao diện lập trình đồ hoạ
    - Không có các thành phần lập trình trực quan



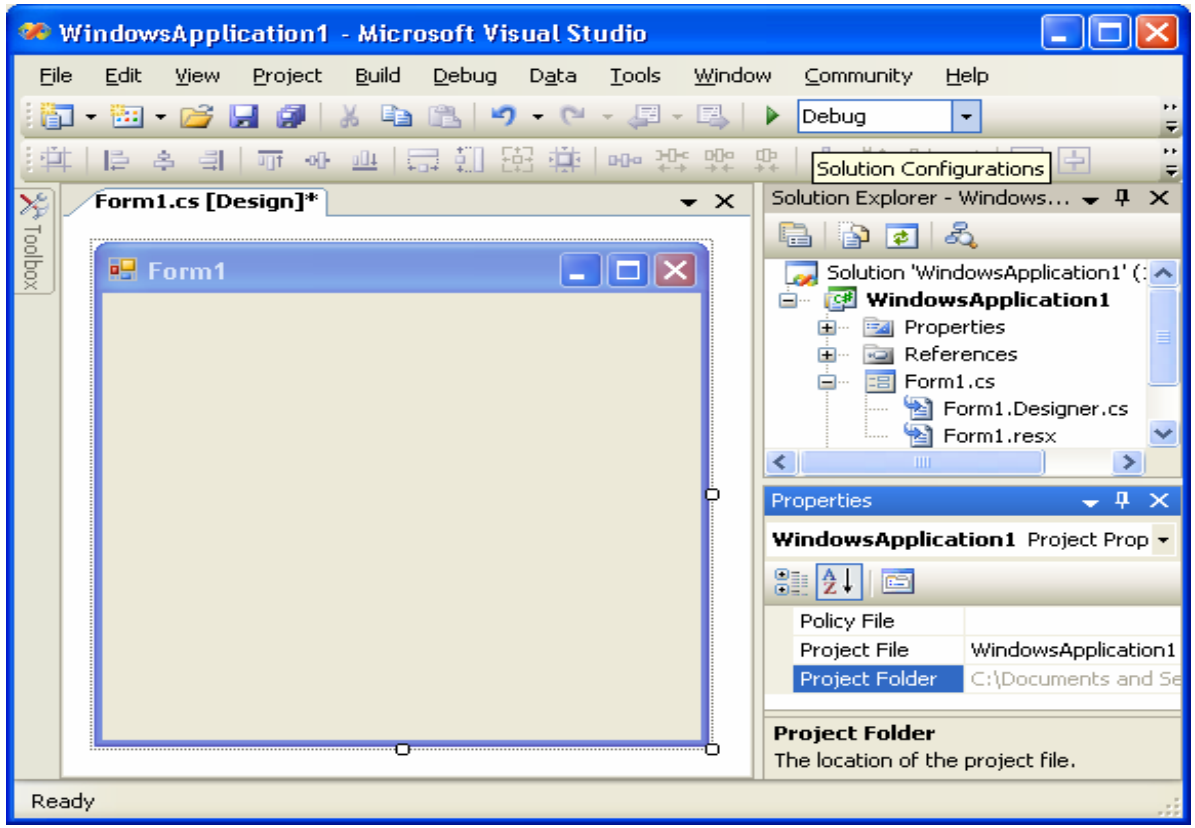
Môi trường lập trình .NET

- Một số khái niệm
  - Ứng dụng Windows applications
    - Chạy trên môi trường Windows
    - Có giao diện lập trình đồ hoạ
    - Có các thành phần lập trình trực quan



Môi trường lập trình .NET

- Giao diện .NET

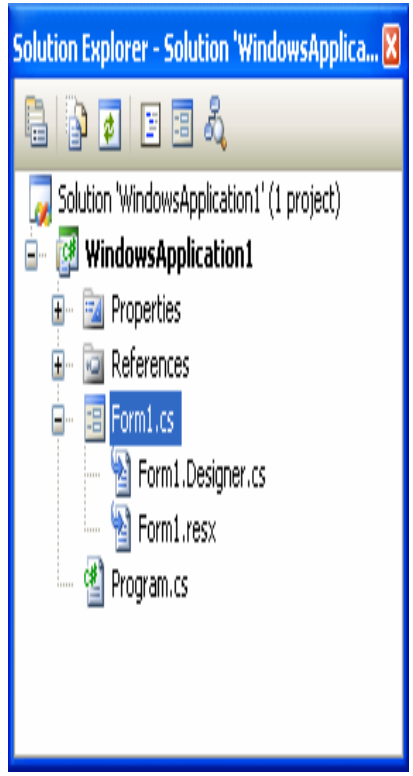


Môi trường lập trình .NET

- Các thành phần
  - **Form**
    - Là thành phần của giao diện đồ hoạ
    - Chứa các điều khiển lập trình sử dụng lại (**Reused**)
  - **Tabs**
    - Một tài liệu được mở trong một tab
    - Dùng để tiết kiệm không gian khi lập trình
  - **Menu**
    - Chứa các lệnh dùng để xây dựng và thực hiện chương trình
  - **ToolBar**
    - Chứa các biểu tượng dùng để thực hiện các lệnh

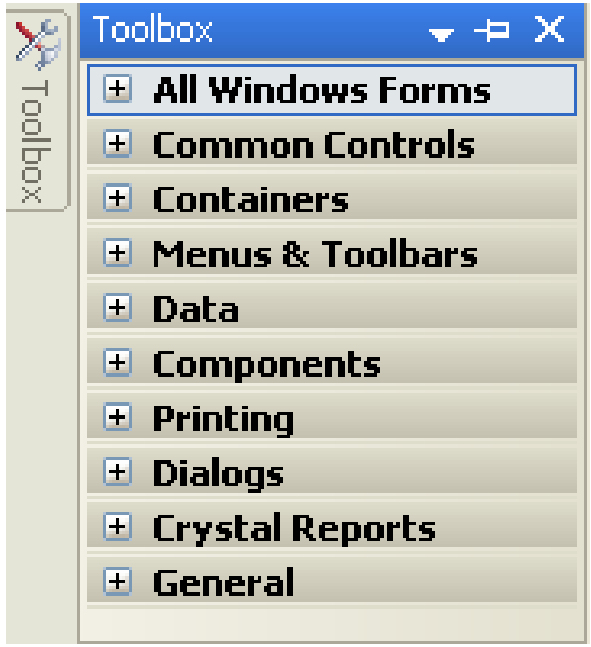
Môi trường lập trình .NET

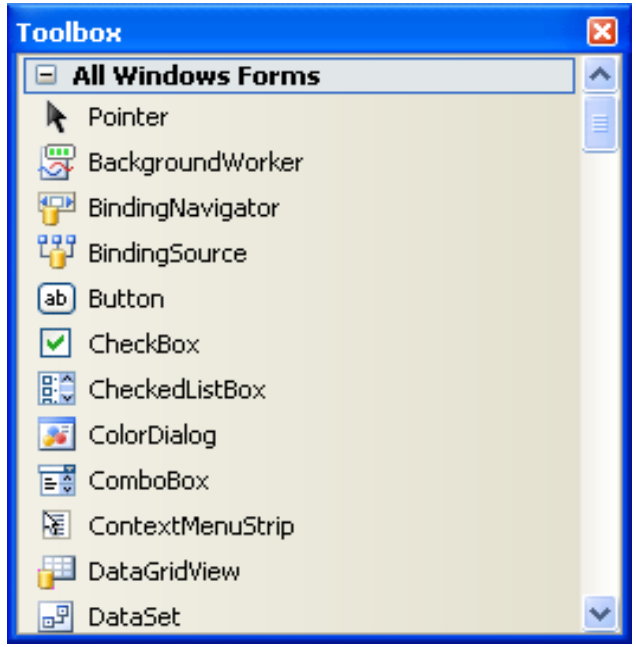
- Cửa sổ **Solution Explorer**
  - Hiển thị các Project trong Solution
    - **Project** đậm được chạy đầu tiên
  - Hiển thị tất cả các tệp trong Project
  - Thanh công cụ
    - **Show All files**: Hiển thị các tệp
    - **Refresh**: Đọc lại các tệp
    - **View Code**: Hiển thị mã lệnh
    - **View Design**: Hiển thị thiết kế
    - **Class Diagram**: Hiển thị lược đồ lớp



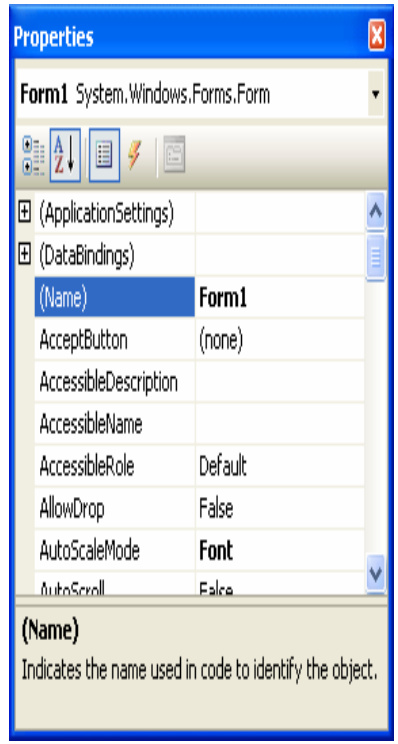
Môi trường lập trình .NET

- Cửa sổ **Toolbox**
  - Chứa các điều kiển lập trình trực quan
  - Nhóm theo các chức năng



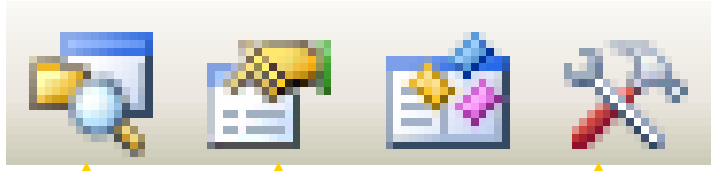


- Cửa sổ **Properties**
  - Hiển thị các thuộc tính và sự kiện của đối tượng được chọn
    - Cột trái là thuộc tính hoặc sự kiện
    - Cột phải là giá trị
  - Thanh công cụ
    - **Alphabetic**: Sắp xếp vần abc..
    - **Properties**: Thuộc tính của đối tượng
    - **Events**: Sự kiện của đối tượng
  - Description
    - Mô tả thuộc tính hoặc sự kiện



Môi trường lập trình .NET

- Một số thao tác cơ bản
  - Hiển thị cửa sổ **`Solution Explorer`**
    - Chọn biểu tượng Solution Explorer
  - Hiển thị cửa sổ **`Properties`**
    - Chọn biểu tượng Properties
  - Hiển thị cửa sổ **`ToolBox`**
    - Chọn biểu tượng ToolBox



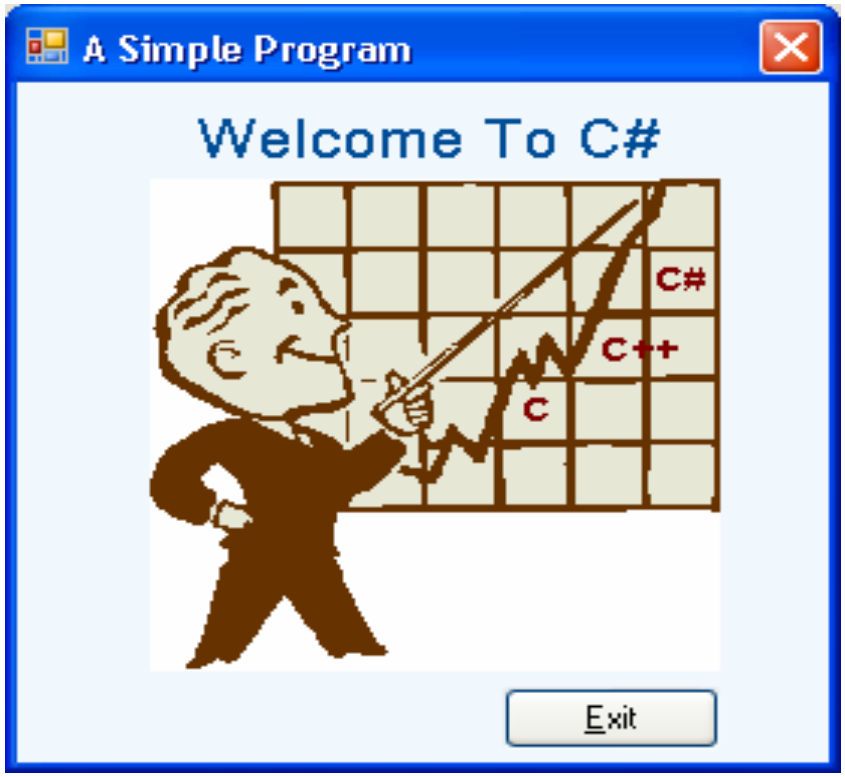
`Solution Explorer` **`Properties`** **`ToolBox`**

Môi trường lập trình .NET

- Một số thao tác cơ bản
  - Dịch **Project**
    - Ấn F6 hoặcchọn menu **Build → Build Solution**
  - Chạy **Project**
    - Ấn F5 hoặc chọn menu **Debug → Start Debugging**
  - Chú ý:
    - Khi chạy **Project** máy tự động ghi các thay đổi
    - Trong một **Project** có nhiều **Form**, để chạy **Form** nào cần vào cửa sổ **Solution Explorer** và chọn **program.cs** để khai báo lại **Form** cần chạy.
    - **Project** chỉ chạy được khi mọi Form đã hết lỗi cú pháp.

Ví dụ

- Xây dựng Form



Ví dụ

- Mã lệnh của chương trình

WindowsApplication1.Form1 | Form1()

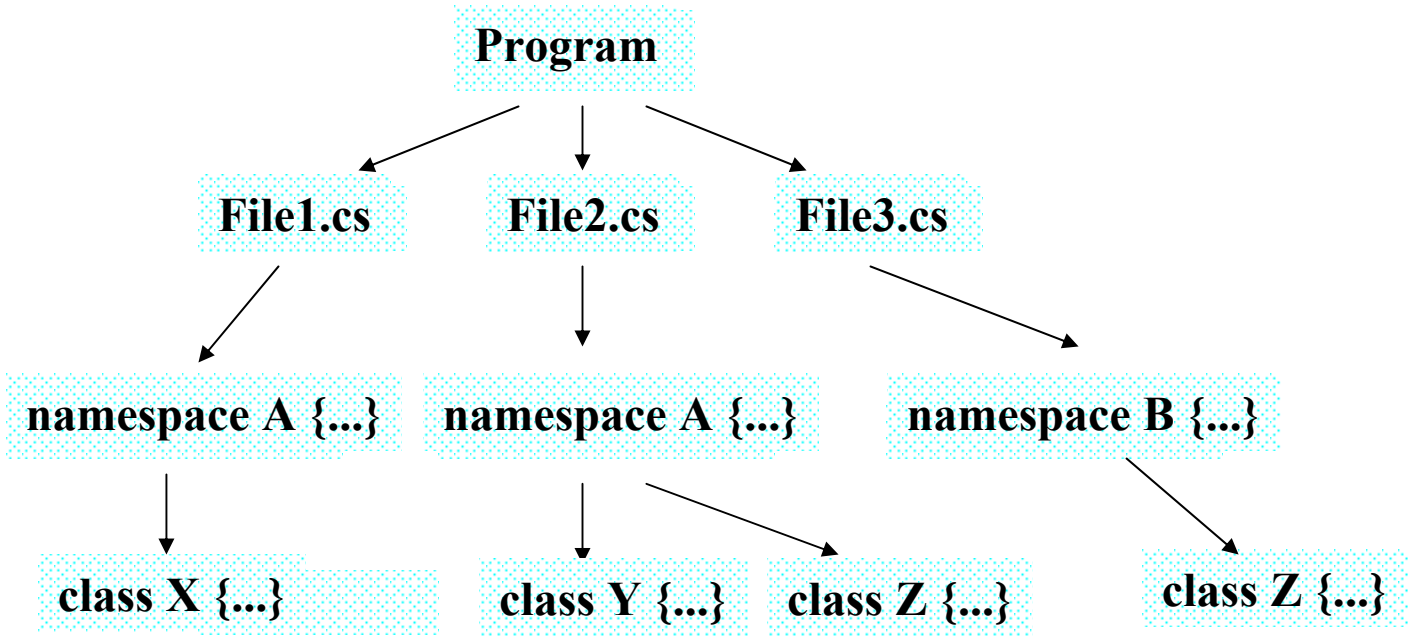
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnExit_Click(object sender, Eve
        {   //Exit simple program
            Application.Exit();
        }
    }
```

Một số khái niệm

- Cấu trúc của một chương trình Visual C#



Một số khái niệm

- Không gian tên - namespaces
  - Các lớp tổ chức trong các thư viện
  - Cho phép sử dụng lại mã lệnh
  - Phải được mở khi sử dụng các lớp trong namespace
  - Ví dụ
    - `using System.Text;`
    - `using System.Windows.Forms;`
    - `namespace WindowsApplication1`

Một số khái niệm

- Một số không gian tên trong .NET
  - **`System`**: Chứa các lớp và các kiểu dữ liệu cơ sở (**`int`**, **`double`**, **`char`**,...). Mọi chương trình đều phải mở thư viện này.
  - **`System.Data`**: Chứa các lớp của ADO .NET dùng cho truy nhập và thao tác với cơ sở dữ liệu.
  - **`System.Drawing`**: Chứa các lớp làm việc với đồi hoạc.
  - **`System.IO`**: Chứa các lớp làm việc với tệp và thư mục

Một số khái niệm

- Một số không gian tên trong .NET
  - **`System.Windows.Forms`**: Chứa các lớp làm việc với giao diện đồ hoạ
  - **`System.Xml`**: Chứa các lớp xử lý dữ liệu XML.

Một số khái niệm

- Cấu trúc của lớp

```
class <classname> {

    ... fields, constants ...
    ... methods ...
    ... constructors, destructors ...

    ... properties ...
    ... events ...

    ... indexers ...
    ... overloaded operators ...
    ... nested types (classes, structs, enums,)...
}
```

Một số khái niệm

- Cấu trúc của lớp
  - Constructors – Hàm thiết lập của lớp
    - Tự động thực hiện khi tạo đối tượng của lớp
    - Có quyền public
    - Có tên trùng với tên lớp
    - Không có giá trị trả về
    - Có thể định nghĩa các tham số
    - Có thể định nghĩa nhiều hơn một hàm thiết lập trong lớp
    - Một hàm thiết lập có thể gọi một hàm thiết lập khác với từ khoá *this*.
  - Tạo đối tượng dựa trên hàm thiết lập của lớp

Một số khái niệm

- Cấu trúc của lớp
  - Hàm thiết lập ngầm định
    - Nếu một lớp không định nghĩa hàm thiết lập, chương trình dịch tự sinh ra ham thiết lập ngầm định
    - Nếu có một hàm thiết lập được khai báo, hàm thiết lập ngầm định không được sinh ra.
  - Destructors - hàm huỷ bỏ
    - Tự động thực hiện khi giải phóng đối tượng khỏi bộ nhớ
    - Không có từ khoá public hoặc private
    - Tên hàm bắt đầu bằng dấu ~ và tiếp đến là tên lớp
    - Mỗi lớp chỉ có một hàm huỷ bỏ

Một số khái niệm

- Ví dụ

```
class rectangle{
    private float a, b;                          //fields
    public rectangle(float x-0, float y=0){ //Constructor
        a = x; b = y;
    }
    public void init(float x, float y){      //Method
        a = x; b = y;
    }
    public float area(){                         //Method
        returb a*b;
    }
}
```

Một số khái niệm

- Đối tượng (object)
  - Đối tượng là sự đóng gói của dữ liệu và phương thức (Object = Data + Method)
  - Dữ liệu: Mô tả đối tượng
  - Phương thức: Các hàm xử lý dữ liệu của đối tượng
  - Truy nhập các thành phần của đối tượng
    - **`public`**: Truy nhập được ở mọi nơi.
    - **`private:`** Chỉ truy nhập được trong định nghĩa lớp
  - Đối tượng phải được tạo bằng từ khoá new