ỦY BAN NHÂN DÂN TỈNH ĐỒNG THÁP
TRƯỜNG CAO ĐẲNG CỘNG ĐỒNG ĐỒNG THÁP

GIÁO TRÌNH

MÔN HỌC: LẬP TRÌNH TRỰC QUAN
NGÀNH, NGHỀ: QUẢN TRỊ MẠNG MÁY TÍNH
TRÌNH ĐỘ: CAO ĐẲNG

(Ban hành kèm theo Quyết định số /QĐ-CĐCĐ ngày tháng năm 20…
của Hiệu trưởng trường Cao đẳng Cộng đồng Đồng Tháp)

Đồng Tháp, năm 2017

TUYÊN BỐ BẢN QUYỀN

LỜI GIỚI THIỆU

Giáo trình Lập trình trực quan, nhằm cung cấp cho sinh viên kiến thức cơ bản về lập trình trực quab trong Visual Studio .Net.

Nội dung bài giảng gồm 5 chương sau:

- Tổng quan về Framework.Net.
- Ngôn ngữ lập trình c#
- Lập trình hướng đối tượng
- Form và Các điều khiển.

Bài giảng dành cho đối tượng học viên trung cấp tin học - ngành quản trị cơ sở dữ liệu.

Dù đã cố gắng để hoàn thành bài giảng này nhưng không thể tránh khỏi những sai sót. Rất mong sự đóng góp của quý đồng nghiệp để bài giảng ngày càng hoàn thiện hơn.

Tôi chân thành cảm ơn sự động viên và đóng góp ý kiến của các bạn đồng nghiệp trong quá trình biên soạn tài liệu.

Nhóm biên soạn

MỤC LỤC

LỜI GIỚI THIỆU

MỤC LỤC

CHƯƠNG 1 – TỔNG QUAN VỀ FRAMEWORK .NET

Mục tiêu:

- Trình bày tổng quan về .FrameWork. Net, các khái niệm cơ bản trong môi trường .Net
- Trình bày được các bước để xây dựng một ứng dụng trong môi trường .Net
- Tạo được ứng dụng đơn giản đầu tiên trong môi trường .Net
- Rèn luyện tính tích cực, chủ động, thích học hỏi.
- Thực hiện thao tác an toàn, chính xác trên máy tính.

1.1 Giới thiệu .Net

Microsoft .NET gồm 2 phần chính: **Framework** và **Integrated Development Environment (IDE)**. Framework cung cấp những gì cần thiết và căn bản, chữ Framework có nghĩa là khung hay khung cảnh trong đó ta dùng những hạ tầng cơ sở theo một qui ước nhất định để công việc được trôi chảy. IDE thì cung cấp một môi trường giúp chúng ta triển khai dễ dàng, và nhanh chóng các ứng dụng dựa trên nền tảng .NET. Nếu không có IDE chúng ta cũng có thể dùng một trình soạn thảo ví như Notepad hay bất cứ trình soạn thảo văn bản nào và sử dụng command line để biên dịch và thực thi, tuy nhiên việc này mất nhiều thời gian. Tốt nhất là chúng ta dùng IDE phát triển các ứng dụng, và cũng là cách dễ sử dụng nhất.

Thành phần Framework là quan trọng nhất .NET là cốt lõi và tinh hoa của môi trường, còn IDE chỉ là công cụ để phát triển dựa trên nền tảng đó thôi. Trong .NET toàn bộ các ngôn ngữ C#, Visual C++ hay Visual Basic.NET đều dùng cùng một IDE.

Tóm lại Microsoft .NET là nền tảng cho việc xây dựng và thực thi các ứng dụng phân tán thế hệ kế tiếp. Bao gồm các ứng dụng từ client đến server và các dịch vụ khác. Một số tính năng của Microsoft .NET cho phép những nhà phát triển sử dụng như sau:

- Một mô hình lập trình cho phép nhà phát triển xây dựng các ứng dụng dịch vụ web và ứng dụng client với Extensible Markup Language (XML).
- Tập hợp dịch vụ XML Web, như Microsoft .NET My Services cho phép nhà phát triển đơn giản và tích hợp người dùng kinh nghiệm.
- Cung cấp các server phục vụ bao gồm: Windows 2000, SQL Server, và BizTalk Server, tất cả điều tích hợp, hoạt động, và quản lý các dịch vụ XML Web và các ứng dụng.
- Các phần mềm client như Windows XP và Windows CE giúp người phát triển phân phối sâu và thuyết phục người dùng kinh nghiệm thông qua các dòng thiết bị.
- Nhiều công cụ hỗ trợ như Visual Studio .NET, để phát triển các dịch vụ Web XML, ứng dụng trên nền Windows hay nền web một cách dể dàng và hiệu quả.

Kiến trúc .NET Framework

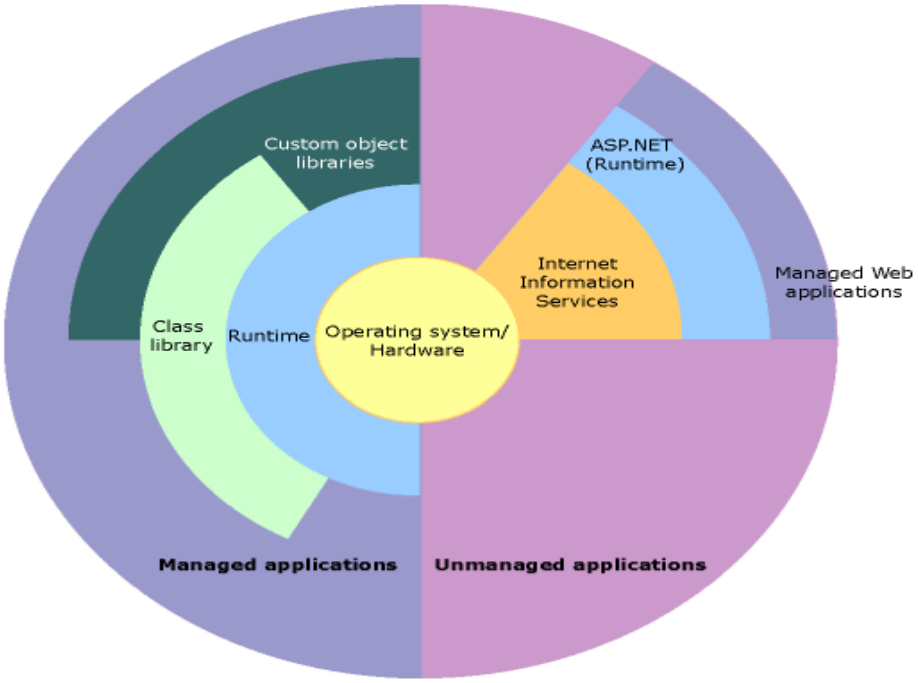
.NET Framework là một platform mới làm đơn giản việc phát triển ứng dụng trong môi trường phân tán của Internet. .NET Framework được thiết kế đầy đủ để đáp ứng theo quan điểm sau:

- Để cung cấp một môi trường lập trình hướng đối tượng vững chắc, trong đó mã nguồn đối tượng được lưu trữ và thực thi một cách cục bộ. Thực thi cục bộ nhưng được phân tán trên Internet, hoặc thực thi từ xa.

- Để cung cấp một môi trường thực thi mã nguồn mà tối thiểu được việc đóng gói phần mềm và sự tranh chấp về phiên bản.
- Để cung cấp một môi trường thực thi mã nguồn mà đảm bảo việc thực thi an toàn mã nguồn, bao gồm cả việc mã nguồn được tạo bởi hãng thứ ba hay bất cứ hãng nào mà tuân thủ theo kiến trúc .NET.
- Để cung cấp một môi trường thực thi mã nguồn mà loại bỏ được những lỗi thực hiện các script hay môi trường thông dịch.
- Để làm cho những người phát triển có kinh nghiệm vững chắc có thể nắm vững nhiều kiểu ứng dụng khác nhau. Như là từ những ứng dụng trên nền Windows đến những ứng dụng dựa trên web.
- Để xây dựng tất cả các thông tin dựa triên tiêu chuẩn công nghiệp để đảm bảo rằng mã nguồn trên .NET có thể tích hợp với bất cứ mã nguồn khác. .NET Framework có hai thành phần chính: Common Language Runtime (CLR) và thư viện lớp .NET Framework. CLR là nền tảng của .NET Framework. Chúng ta có thể hiểu runtime như là một agent quản lý mã nguồn khi nó được thực thi, cung cấp các dịch vụ cốt lõi như: quản lý bộ nhớ, quản lý tiểu trình, và quản lý từ xa. Ngoài ra nó còn thúc đẩy việc sử dụng kiểu an toàn và các hình thức khác của việc chính xác mã nguồn, đảm bảo cho việc thực hiện được bảo mật và mạnh mẽ. Thật vậy, khái niệm quản lý mã nguồn là nguyên lý nền tảng của runtime. Mã nguồn mà đích tới runtime thì được biết như là mã nguồn được quản lý (managed code). Trong khi đó mã nguồn mà không có đích tới runtime thì được biết như mã nguồn không được quản lý (unmanaged code).

Thư viện lớp, một thành phần chính khác của .NET Framework là một tập hợp hướng đối tượng của các kiểu dữ liệu được dùng lại, nó cho phép chúng ta có thể phát triển những ứng dụng từ những ứng dụng truyền thống command-line hay

những ứng dụng có giao diện đồ họa (GUI) đến những ứng dụng mới nhất được cung cấp bởi ASP.NET, như là Web Form và dịch vụ XML Web.



Hình 1.1: Mô tả các thành phần trong .NET Framework.

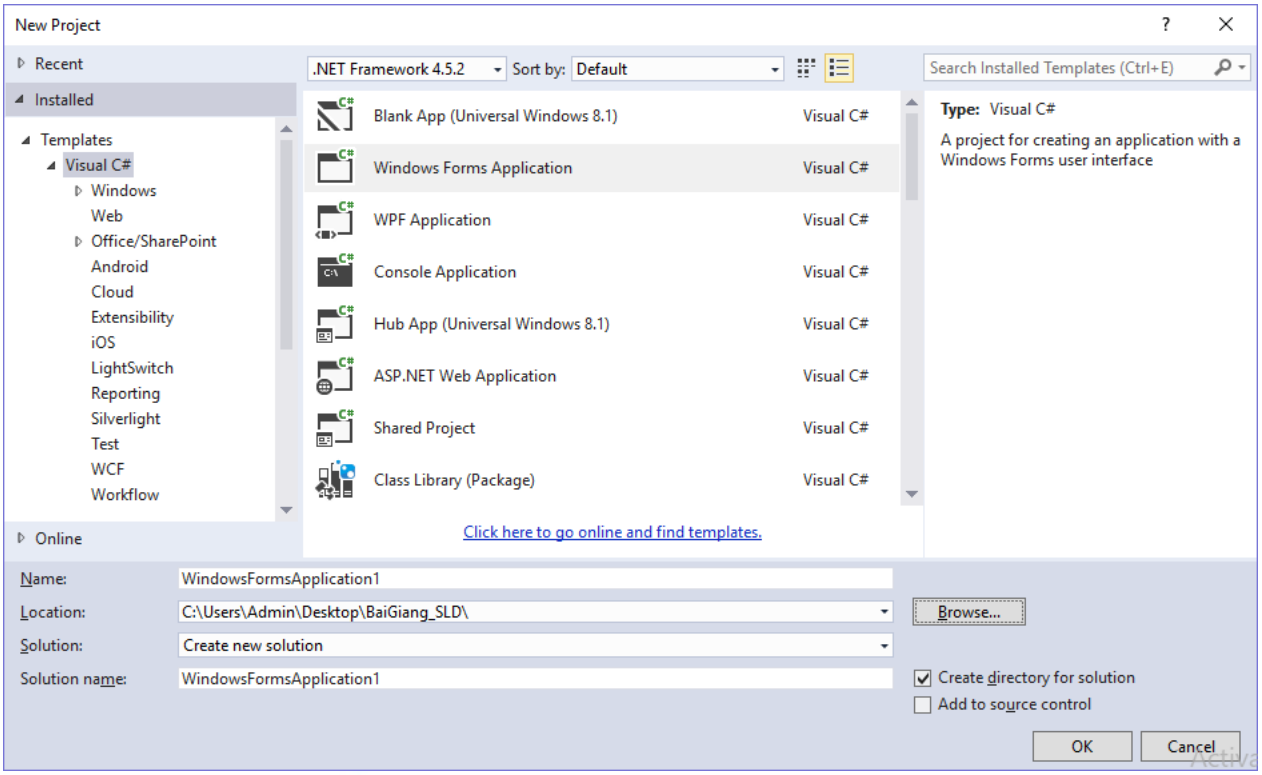
1.2 Các loại ứng dụng

Ứng dụng Console: ứng dụng này giao tiếp với người dùng thông quan bàn phím và không có giao diện người dùng (UI), giống như các ứng dụng thường thấy trong Windows.

Ứng dụng WindowsForms: là ứng dụng có giao diện đồ họa chạy trên hệ điều hành Windows.

1.3 Các bước xây dựng trong môi trường .Net

Bước 1: Tạo dự án mới



Các thông tin cần xác định cho một dự án mới:

- Name: tên
- Location: nơi lưu trữ

Bước 2: Thiết kế giao diện

Đưa các điều khiển vào form để thiết kế giao diện

Bước 3: Đặt các thuộc tính cho các điều khiển trên giao diện

Bước 4: Viết code

Bước 5: Thi hành chương trình – Nhấn F5

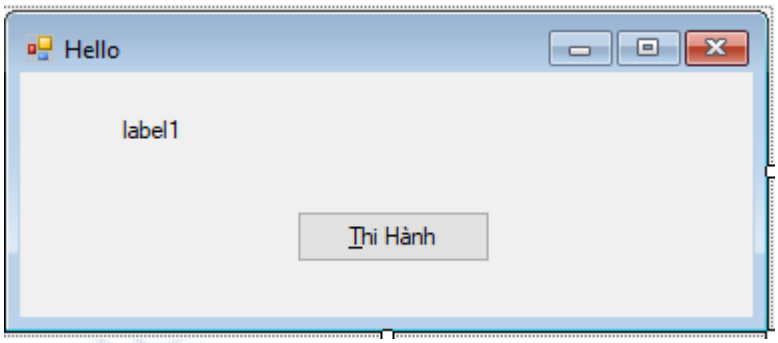
1.4 Ứng dụng đầu tiên

Bước 1: Tạo dự án mới

Bước 2: Thiết kế giao diện

Đưa các điều khiển Label1 và Button1 vào form.

Chỉnh vị trí và kích thước của các điều khiển cho hợp lý.



Bước 2: Đặt các thuộc tính cho các điều khiển

Form

Name	frmHello
Autosize	F
BackColor	255, 192, 192
Text	Rỗng

Label1

Name	lblHello
Autosize	False

BackColor	255, 192, 192
Font	Font Microsoft Sans Serif, 15.75pt, style=Bold
ForeColor	ForeColor Blue
Text	Rỗng
TextAlign	TextAlign MiddleCenter

Button1

Name	btnThiHanh
Text	&Thi Hành

Bước 4: Viết code

```
private void btnThiHanh_Click(object sender, EventArgs e)
        {
            lblHello.Text = "Hello World!";
        }
```

Bước 5: Gõ F5 để thi hành chương trình

CHƯƠNG 2 – NGÔN NGỮ LẬP TRÌNH C#

Mục tiêu:

- Trình bày được các thành phần trong ngôn ngữ lập trình c#.
- Trình bày và sử dụng thành thạo các cấu trúc điều khiển để viết chương trình.
- Trình bày và sử dụng thành thạo các kiểu dữ liệu có cấu trúc.
- Xây đựng được hàm để áp dụng cho viết chương trình.
- Sử dụng được cấu trúc try … catch để bẩy lỗi.

Rèn luyện tính tích cực, chủ động trong tư duy, thao tác cận thận, an toàn trên máy tính.

2.1 Giới thiệu

Ngôn ngữ C# khá đơn giản, chỉ khoảng 80 từ khóa và hơn mười mấy kiểu dữ liệu được xây dựng sẵn. Tuy nhiên, ngôn ngữ C# có ý nghĩa cao khi nó thực thi những khái niệm lập trình hiện đại. C# bao gồm tất cả những hỗ trợ cho cấu trúc, thành phần component, lập trình hướng đối tượng. Những tính chất đó hiện diện trong một ngôn ngữ lập trình hiện đại. Và ngôn ngữ C# hội đủ những điều kiện như vậy, hơn nữa nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Ngôn ngữ C# được phát triển bởi đội ngũ kỹ sư của Microsoft, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth. Cả hai người này điều là những người nổi tiếng, trong đó Anders Hejlsberg được biết đến là tác giả của Turbo Pascal, một ngôn ngữ lập trình PC phổ biến. Và ông đứng đầu nhóm thiết kế Borland Delphi,

một trong những thành công đầu tiên của việc xây dựng môi trường phát triển tích hợp (IDE) cho lập trình client/server.

Phần cốt lõi hay còn gọi là trái tim của bất cứ ngôn ngữ lập trình hướng đối tượng là sự hỗ trợ của nó cho việc định nghĩa và làm việc với những lớp. Những lớp thì định nghĩa những kiểu dữ liệu mới, cho phép người phát triển mở rộng ngôn ngữ để tạo mô hình tốt hơn để giải quyết vấn đề. Ngôn ngữ C# chứa những từ khóa cho việc khai báo những kiểu lớp đối tượng mới và những phương thức hay thuộc tính của lớp, và cho việc thực thi đóng gói, kế thừa, và đa hình, ba thuộc tính cơ bản của bất cứ ngôn ngữ lập trình hướng đối tượng.

Trong ngôn ngữ C# mọi thứ liên quan đến khai báo lớp điều được tìm thấy trong phần khai báo của nó. Định nghĩa một lớp trong ngôn ngữ C# không đòi hỏi phải chia ra tập tin header và tập tin nguồn giống như trong ngôn ngữ C++. Hơn thế nữa, ngôn ngữ C# hỗ trợ kiểu XML, cho phép chèn các tag XML để phát sinh tự động các document cho lớp.

C# cũng hỗ trợ giao diện interface, nó được xem như một cam kết với một lớp cho những dịch vụ mà giao diện quy định. Trong ngôn ngữ C#, một lớp chỉ có thể kế thừa từ duy nhất một lớp cha, tức là không cho đa kế thừa như trong ngôn ngữ C++, tuy nhiên một lớp có thể thực thi nhiều giao diện. Khi một lớp thực thi một giao diện thì nó sẽ hứa là nó sẽ cung cấp chức năng thực thi giao diện.

Trong ngôn ngữ C#, những cấu trúc cũng được hỗ trợ, nhưng khái niệm về ngữ nghĩa của nó thay đổi khác với C++. Trong C#, một cấu trúc được giới hạn, là kiểu dữ liệu nhỏ gọn, và khi tạo thể hiện thì nó yêu cầu ít hơn về hệ điều hành và bộ nhớ so với một lớp. Một cấu trúc thì không thể kế thừa từ một lớp hay được kế thừa nhưng một cấu trúc có thể thực thi một giao diện.

Ngôn ngữ C# cung cấp những đặc tính hướng thành phần (component-oriented), như là những thuộc tính, những sự kiện. Lập trình hướng thành phần được hỗ trợ bởi CLR cho phép lưu trữ metadata với mã nguồn cho một lớp. Metadata mô tả cho một lớp, bao gồm những phương thức và những thuộc tính của nó, cũng như những sự bảo mật cần thiết và những thuộc tính khác. Mã nguồn chứa đựng những logic cần thiết để thực hiện những chức năng của nó.. Do vậy, một lớp được biên dịch như là một khối self-contained, nên môi trường hosting biết được cách đọc metadata của một lớp và mã nguồn cần thiết mà không cần những thông tin khác để sử dụng nó.

Một lưu ý cuối cùng về ngôn ngữ C# là ngôn ngữ này cũng hỗ trợ việc truy cập bộ nhớ trực tiếp sử dụng kiểu con trỏ của C++ và từ khóa cho dấu ngoặc [] trong toán tử. Các mã nguồn này là không an toàn (unsafe). Và bộ giải phóng bộ nhớ tự động của CLR sẽ không thực hiện việc giải phóng những đối tượng được tham chiếu bằng sử dụng con trỏ cho đến khi chúng được giải phóng.

2.2 Các thành phần cơ bản

2.2.1 Biến

Một biến là một vùng lưu trữ với một kiểu dữ liệu. Biến có thể được gán giá trị và cũng có thể thay đổi giá trị khi thực hiện các lệnh trong chương trình. Để tạo một biến chúng ta phải khai báo kiểu của biến và gán cho biến một tên duy nhất.

Biến có thể được khởi tạo giá trị ngay khi được khai báo, hay nó cũng có thể được gán một giá trị mới vào bất cứ lúc nào trong chương trình.

Cú pháp khai báo biến

```
<type> <tên biến>
```

hoặc

<type> <tên biến> = <giá trị>;

Ví dụ:

```
int a, b, c, d, e;

int i=100;
```

Chú ý:

- C# đòi hỏi các biến phải được khởi tạo trước khi được sử dụng
- Việc sử dụng biến khi chưa được khởi tạo là không hợp lệ trong C#
- Tuy nhiên không nhất thiết lúc nào chúng ta cũng phải khởi tạo biến. Nhưng để dùng được thì bắt buộc phải gán cho chúng một giá trị trước khi có một lệnh nào tham chiếu đến biến đó. Điều này được gọi là gán giá trị xác định cho biến và C# bắt buộc phải thực hiện điều này.

2.2.2 Các kiểu dữ liệu

Ngôn ngữ C# đưa ra các kiểu dữ liệu xây dựng sẵn rất hữu dụng, phù hợp với một ngôn ngữ lập trình hiện đại, mỗi kiểu dữ liệu được ánh xạ đến một kiểu dữ liệu được hỗ trợ bởi hệ thống xác nhận ngôn ngữ chung (Common Language Specification: CLS) trong MS.NET. Việc ánh xạ các kiểu dữ liệu nguyên thuỷ của C# đến các kiểu dữ liệu của .NET sẽ đảm bảo các đối tượng được tạo ra trong C# có thể được sử dụng đồng thời với các đối tượng được tạo bởi bất cứ ngôn ngữ khác được biên dịch bởi .NET, như VB.NET.

Mỗi kiểu dữ liệu có một sự xác nhận và kích thước không thay đổi, không giống như C++, int trong C# luôn có kích thước là 4 byte bởi vì nó được ánh xạ từ kiểu Int32 trong .NET.

Kiểu c#	Số byte	Kiểu .NET	Mô tả
byte	1	Byte	Số nguyên dương không dấu từ 0-255
char	2	Char	Ký tự Unicode
bool	1	Boolean	Giá trị logic true/false
sbyte	1	Sbyte	Số nguyên có dấu (từ -128 đến 127)
short	2	Int16	Số nguyên -32.768 đến 32.767
ushort	2	Uint16	Số nguyên từ 0 đến 65.535
int	4	Int32	Số nguyên -214.7483.647 đến 2.147.483.647
uint	4	Uint32	Số nguyên từ 0 đến 4.294.967.295
float	4	Single	Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-38 đến 3,4E+38, với 7 chữ số có nghĩa
double	8	Double	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ 1.7E-308 đến 1,7E+308, với 15, 16 chữ số có nghĩa
decimal	8	Decimal	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính kiểu này đòi hỏi phải có hậu tố “m” hoặc “M” theo sau giá trị

long	8	Int64	Kiểu nguyên -9.223.370.036.854.775.808 đến 9.223.370.036.854.775.807
ulong	8	Uint64	Số nguyên từ 0 đến 0xffffffffffffffff

Chú ý:

Kiểu giá trị logic chỉ có thể nhận được giá trị là true hay false mà thôi. Một giá trị nguyên không thể gán vào một biến kiểu logic trong C# và không có bất cứ chuyển đổi ngầm định nào. Điều này khác với C/C++, cho phép biến logic được gán giá trị nguyên, khi đó giá trị nguyên 0 là false và các giá trị còn lại là true.

2.2.3 Hằng

Hằng cũng là một biến nhưng giá trị của hằng không thay đổi. Biến là công cụ rất mạnh, tuy nhiên khi làm việc với một giá trị được định nghĩa là không thay đổi, ta phải đảm bảo giá trị của nó không được thay đổi trong suốt chương trình

Cú pháp khai báo hằng:

<const> <type> <tên hằng> = <giá trị>;

Ví dụ:

```
const int ChuNhat = 0;

const int ThuHai = 1;

const int ThuBa=2;
```

Chú ý: