I. C# CƠ BẢN

Thông tin về Trung Tâm Lập Trình TP

Thông tin về khóa học

Đối tượng

Nền tảng cần có

C# là gì? Vì sao học c#?

1. C# là gì?

C# là một ngôn ngữ lập trình đơn giản, hiện đại, mục đích tổng quát, hướng đối tượng được phát triển bởi Microsoft và được phê chuẩn bởi European Computer Manufacturers Association (ECMA) và International Standards Organization (ISO).

C# được phát triển bởi Anders Hejlsberg và team của ông trong khi phát triển .Net Framework.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

1. Dưới đây là các lý do làm C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:

* C# là ngôn ngữ lập trình hiện đại, mục đích tổng quát.
* C# là ngôn ngữ hướng đối tượng tốt nhất(OOP).
* Dễ dàng để học.
* Là một ngôn ngữ được cấu trúc rõ ràng, có nhiều tài liệu hướng dẫn.
* Cung cấp môi trường để tạo ra các chương trình có hiệu quả.
* Có thể được biên dịch trên nhiều nền tảng máy tính khác nhau(Cross platforms).
* Là một phần của .Net Framework.

Các đặc điểm lập trình mạnh mẽ của C#

Cấu trúc C# khá gần với các ngôn ngữ high-level truyền thống, C và C++, và là một ngôn ngữ lập trình hướng đối tượng. Nó có sự giống nhau mạnh mẽ với Java, nó có nhiều đặc điểm lập trình mạnh mẽ mà làm cho nó trở nên ưa thích với các lập trình viên trên toàn thế giới.

Dưới đây là các đặc điểm quan trọng của C#:

* Điều kiện Boolean
* Tự động dọn rác bởi Garbage-Collector (GC)
* Thư viện chuẩn (Standard Library)
* Assembly
* Property và sự kiện (Event)
* Delegate Quản lý sự kiện
* Dễ dàng để sử dụng Generic
* Indexer
* Biên dịch có điều kiện (Conditional Compilation)
* Đa luồng dễ dàng (Multithreading)
* LINQ và Lambda Expression
* Tích hợp với Windows

Cài đặt môi trường (Enviroment)

**Giới thiệu .Net Framework**

.Net Framework là một nền tảng mang tính cách mạng giúp bạn viết các kiểu ứng dụng sau:

* Windows application
* Web application
* Dịch vụ Web

Các ứng dụng .Net Framework là các ứng dụng đa nền tảng. Nền tảng đã được thiết kế theo cách như vậy có thể được sử dụng từ bất kỳ ngôn ngữ lập trình nào: C#, C++, Visual Basic, Jscript, COBOL… Tất cả các ngôn ngữ này có thể truy cập tới nền tảng này cũng như giao tiếp với nhau.

.Net Framework gồm một thư viện code khổng lồ được sử dụng bởi các ngôn ngữ client như C#. Dưới đây là một số thành phần của .Net Framework.

* Common Language Runtime (CLR)
* .Net Framework Class Library
* Common Language Specification
* Common Type System
* Metadata và Assembly
* Windows Form
* ASP.Net và ASP.Net AJAX
* ADO.Net
* Windows Workflow Foundation (WF)
* Windows Presentation Foundation
* Windows Communication Foundation (WCF)
* LINQ

**Integrated Development Environment (IDE) cho C#**

Microsoft cung cấp các công cụ phát triển sau cho lập trình C#:

* Visual Studio 2015(Bảng đề nghị)
* Visual Studio 2013
* Visual Studio 2012

Cú pháp cơ bản (Basic Syntax)

C# là một ngôn ngữ lập trình hướng đối tượng. Trong phương pháp lập trình hướng đối tượng, một chương trình gồm các đối tượng đa dạng mà tương tác với nhau các cách thức của action. Các action mà một đối tượng có thể nhận được gọi là các phương thức. Các đối tượng cùng loại được xem như là có cùng kiểu hoặc, được xem như là trong cùng lớp.

Ví dụ, xét đối tượng Rectangle. Nó có các thuộc tính như length và width. Phụ thuộc vào thiết kế trên. Nó có thể cần những cách để cấp nhận các giá trị của thuộc tính này, tính toán diện tích và hiển thị chi tiết.

Chúng ta cùng theo dõi ví dụ sau triển khai một lớp Rectangle và cú pháp C# cơ bản của chương trình này. Để minh họa rõ ràng tính hướng đối tượng của C#, mình sẽ tạo hai class trong hai file riêng rẽ nhau.

Ví dụ lớp Rectangle: chứa các thuộc tính, phương thức chính

class Rectangle

{

// cac bien thanh vien

double \_length;

double \_width;

//phuong thuc

public void Acceptdetails()

{

\_length = 4.5;

\_width = 3.5;

}

//phuong thuc

public double GetArea()

{

return \_length \* \_width;

}

//phuong thuc

public void Display()

{

Console.WriteLine("Chieu dai: {0}", \_length);

Console.WriteLine("Chieu rong: {0}", \_width);

Console.WriteLine("Dien tich: {0}", GetArea());

}

}

Lớp ExecuteRectangle: là lớp chứa phương thức main() để tiến hành thao tác trên đối tượng Rectangle của lớp Rectangle. Đến đây, có lẽ bạn chưa hiểu gì, nhưng không sao vì đây chỉ là chương giúp bạn làm quen với C# thôi.

using System;

namespace TPCenter

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Chuong trinh minh hoa tinh huong doi tuong trong C#");

Console.WriteLine("--------------------------------------------------\n");

//tao doi tuong Rectangle

Rectangle r = new Rectangle();

//goi cac phuong thuc cua doi tuong nay

r.Acceptdetails();

r.Display();

Console.ReadLine();

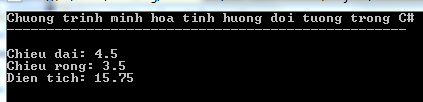
Console.ReadKey();

}

}

}

Nhấn phím **F5** để biên dịch và chạy chương trình C# trên sẽ cho kết quả sau:



Kiểu dữ liệu(Data Types)

Chuyển đổi kiểu dữ liệu(Type Conversion)

Biến(Variables)

Hằng(Constant/Literal)

Toán tử(Operators)

Lệnh If, Else, Switch(Decision Making)

Vòng lặp(Loop)

Tính đóng gói(Encapsulation)

Phương thức(Method)

Nullable

Mảng(Array)

Chuỗi(String)

Cấu trúc(Struct)

Enum

Lớp(Classes)

Tính kế thừa(Inheritance)

Tính đa hình(Polymorphism)

Nạp chồng toán tử(Operator Overloading)

Interface

Namespaces

Chỉ thị tiền xử lý(Preprocessor Directives)

Regular Expression

Exception Handling

File I/O

Attributes

Reflection

Properties

Indexer

Delegate

Event

Collection

Generic

Phương thức nặc danh(Anonymous Methods)

Unsafe Code

Multithreading

Tổng kết