**BDRPC187**

**BỘ TRUYỀN THÔNG & THÔNG TIN**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH & VIỄN THÔNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**ĐỀ TÀI : PHẦN MỀM QUẢN LÝ NHÂN SỰ C#**

**Giảng viên hướng dẫn: Ths. Huỳnh Trung Trụ**

**Sinh viên thực hiện: Nguyễn Thành Nhơn**

**Lớp: D14CQIS01**

**MSSV: N14DCCN062**

***TP.HCM ,Ngày 09 Tháng 07 năm 2018***

LỜI MỞ ĐẦU

☺☺☺☺

Nhà trường mà cụ thể là khoa CNTT trường Học viện công nghệ Bưu Chính Viễn Thông cơ sở TP.HCM đã tạo điều kiện cho sinh viên được tiếp xúc với môi trường làm việc thực tế, thông qua quá trình thực tập kéo dài 1 tháng tại Công ty Saigonlab. Trong quá trình thực tập đã giúp em tìm hiểu, làm quen, và nắm được hoạt động của nhân viên trong Công ty cũng như sự phân công công việc cụ thể cho từng phòng ban và chức vụ cụ thể. Đồng thời biết được quá trình áp dụng những kiến thức đã học trong trường lớp vào thực tế công việc của Công ty.

Đối với quá trình hoàn thiện sản phẩn CNTT trong quá trình thực tập, cụ thể là phần mềm Quản lý nhân sự đã giúp em có những khái niệm cơ bản về công việc quản lý nhân sự trong Công ty. Kết hợp xây dựng phần mềm với việc ôn tập để thi tốt nghiệp, bổ sung những kiến thức đã có và tìm hiểu thêm những kiến thức mới thông qua ngôn ngữ lập trình C#.NET.

Qua đây em xin gửi lời cảm ơn chân thành đến các anh chị em Công ty Saigonlab đã hết lòng huớng dẫn, giúp đỡ và chỉ dạy cho tôi có thể hoàn thành tốt công việc được giao trong suốt quá trình thực tập cũng như lời cảm ơn sâu sắc đến thầy Huỳnh Trung Trụ đã tận tình huớng dẫn hoàn thành đồ án thực tập này.

## **Chương 1 C# và .Net Framework**

Mục tiêu của C# là cung cấp một ngôn ngữ lập trình đơn giản, an toàn, hiện đại, hướng đối tượng, đặt trọng tâm vào Internet, có khả năng thực thi cao cho môi trường .NET. C# là một ngôn ngữ mới, nhưng tích hợp trong nó những tinh hoa của ba thập kỷ phát triển của ngôn ngữ lập trình. Ta có thể dể dàng thầy trong C# có những đặc trưng quen thuộc của Java, C++, Visual Basic, …

Đề tài này đặt trọng tâm giới thiệu ngôn ngữ C# và cách dùng nó như là một công cụ lập trình trên nền tảng .NET. Với ngôn ngữ C++, khi học nó ta không cần quan tâm đến môi trường thực thi. Với ngôn ngữ C#, ta học để tạo một ứng dụng .NET, nếu lơ là ý này có thể bỏ lỡ quan điểm chính của ngôn ngữ này. Do đó, trong đề tài này xét C# tập trung trong ngữ cảnh cụ thể là nền tảng .NET của Microsoft và trong các ứng dụng máy tính để bàn và ứng dụng Internet.

Chương này trình bày chung về hai phần là ngôn ngữ C# và nền tảng .NET, bao gồm cả khung ứng dụng .NET (.NET Framework)

### Nền tảng của .NET

Khi Microsoft công bố C# vào tháng 7 năm 2000, việc khánh thành nó chỉ là một phần trong số rất nhiều sự kiện mà nền tảng .Net được công công bố. Nền tảng .Net là bô khung phát triển ứng dụng mới, nó cung cấp một giao diện lập trình ứng dụng (Application Programming Interface - API) mới mẽ cho các dịch vụ và hệ điều hành Windows, cụ thể là Windows 2000, nó cũng mang lại nhiều kỹ thuật khác nổi bật của Microsoft suốt từ những năm 90. Trong số đó có các dịch vụ COM+, công nghệ ASP, XML và thiết kế hướng đối tượng, hỗ trợ các giao thức dịch vụ web mới như SOAP, WSDL và UDDL với trọng tâm là Internet, tất cả được tích hợp trong kiến trúc DNA.

Nền tảng .NET bao gồm bốn nhóm sau:

* + 1. Một tập các ngôn ngữ, bao gồm C# và Visual Basic .Net; một tập các công cụ phát triển bao gồm Visual Studio .Net; một tập đầy đủ các thư viện phục vụ cho việc xây dựng các ứng dụng web, các dịch vụ web và các ứng dụng Windows; còn có CLR - Common Language Runtime: (ngôn ngữ thực thi dùng chung) để thực thi các đối tượng được xây dựng trên bô khung này.
    2. Một tập các Server Xí nghiệp .Net như SQL Server 2000. Exchange 2000, BizTalk 2000, … chúng cung cấp các chức năng cho việc lưu trữ dữ liệu quan hệ, thư điện tử, thương mại điện tử B2B, …
    3. Các dịch vụ web thương mại miễn phí, vừa được công bố gần đậy như là dự án Hailstorm; nhà phát triển có thể dùng các dịch vụ này để xây dựng các ứng dụng đòi hỏi tri thức về định danh người dùng…
    4. .NET cho các thiết bị không phải PC như điện thoại (cell phone), thiết bị game

### .NET Framework

.Net hỗ trợ tích hợp ngôn ngữ, tức là ta có thể kế thừa các lớp, bắt các biệt lệ, đa hình thông qua nhiều ngôn ngữ. .NET Framework thực hiện được việc này nhờ vào đặc tả Common Type System - CTS (hệ thống kiểu chung) mà tất cả các thành phần

.Net đều tuân theo. Ví dụ, mọi thứ trong .Net đều là đối tượng, thừa kế từ lớp gốc System.Object.

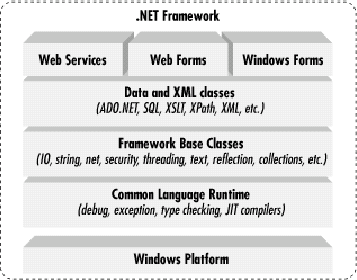
Ngoài ra .Net còn bao gồm Common Language Specification - CLS (đặc tả ngôn ngữ chung). Nó cung cấp các qui tắc cơ bản mà ngôn ngữ muốn tích hợp phải thỏa mãn. CLS chỉ ra các yêu cầu tối thiểu của ngôn ngữ hỗ trợ .Net. Trình biên dịch tuân theo CLS sẽ tạo các đối tượng có thể tương hợp với các đối tượng khác. Bộ thư viện lớp của khung ứng dụng (Framework Class Library - FCL) có thể được dùng bởi bất kỳ ngôn ngữ nào tuân theo CLS.

.NET Framework nằm ở tầng trên của hệ điều hành (bất kỳ hệ điều hành nào không chỉ là Windows). .NET Framework bao bao gồm:

* Bốn ngôn ngữ chính thức: C#, VB.Net, C++, và Jscript.NET
* Common Language Runtime - CLR, nền tảng hướng đối tượng cho phát triển

ứng dụng Windows và web mà các ngôn ngữ có thể chia sẻ sử dụng.

* Bộ thư viện Framework Class Library - FCL.



**Hình 1-1 Kiến trúc khung ứng dụng .Net**

Thành phần quan trọng nhất của .NET Framework là CLR, nó cung cấp môi trường cho ứng dụng thực thi, CLR là một máy ảo, tương tự máy ảo Java. CLR kích hoạt đối tượng, thực hiện kiểm tra bảo mật, cấp phát bộ nhớ, thực thi và thu dọn chúng.

Trong Hình 1-1 tầng trên của CLR bao gồm:

* Các lớp cơ sở
* Các lớp dữ liệu và XML
* Các lớp cho dịch vụ web, web form, và Windows form.

Các lớp này được gọi chung là FCL, Framework Class Library, cung cấp API hướng đối tượng cho tất cả các chức năng của .NET Framework (hơn 5000 lớp).

Các lớp cơ sở tương tự với các lớp trong Java. Các lớp này hỗ trợ các thao tác nhập xuất, thao tác chuổi, văn bản, quản lý bảo mật, truyền thông mạng, quản lý tiểu trình và các chức năng tổng hợp khác …

Trên mức này là lớp dữ liệu và XML. Lớp dữ liệu hỗ trợ việc thao tác các dữ liệu trên cơ sở dữ liệu. Các lớp này bao gồm các lớp SQL (Structure Query Language: ngôn ngữ truy vấn có cấu trúc) cho phép ta thao tác dữ liệu thông qua một giao tiếp SQL chuẩn. Ngoài ra còn một tập các lớp gọi là ADO.Net cũng cho phép thao tác dữ liệu. Lớp XML hỗ trợ thao tác dữ liệu XML, tìm kiếm và diễn dịch XML.

Trên lớp dữ liệu và XML là lớp hỗ trợ xây dựng các ứng dụng Windows (Windows forms), ứng dụng Web (Web forms) và dịch vụ Web (Web services).

### Biên dịch ngôn ngữ trung gian (MSIL)

Với .NET chương trình không biên dịch thành tập tin thực thi, mà biên dịch thành ngôn ngữ trung gian (MSIL - Microsoft Intermediate Language, viết tắt là IL), sau đó chúng được CLR thực thi. Các tập tin IL biên dịch từ C# đồng nhất với các tập tin IL biên dịch từ ngôn ngữ .Net khác.

Khi biên dịch dự án, mã nguồn C# được chuyển thành tập tin IL lưu trên đĩa. Khi chạy chương trình thì IL được biên dịch (hay thông dịch) một lần nữa bằng trình *Just In Time* - JIT, khi này kết quả là mã máy và bộ xử lý sẽ thực thi.

Trình biên dịch JIT chỉ chạy khi có yêu cầu. Khi một phương thức được gọi, JIT phân tích IL và sinh ra mã máy tối ưu cho từng loại máy. JIT có thể nhận biết mã nguồn đã được biên dịch chưa, để có thể chạy ngay ứng dụng hay phải biên dịch lại.

CLS có nghĩa là các ngôn ngữ .Net cùng sinh ra mã IL. Các đối tượng được tạo theo một ngôn ngữ nào đó sẽ được truy cập và thừa kế bởi các đối tượng của ngôn ngữ khác. Vì vậy ta có thể tạo được một lớp cơ sở trong VB.Net và thừa kế nó từ C#.

### Ngôn ngữ C#

C# là một ngôn ngữ rất đơn giản, với khoảng 80 từ khoá và hơn mười kiểu dữ liệu dựng sẵn, nhưng C# có tính diễn đạt cao. C# hỗ trợ lập trình có cấu trúc, hướng đối tượng, hướng thành phần (component oriented).

Trọng tâm của ngôn ngữ hướng đối tượng là lớp. Lớp định nghĩa kiểu dữ liệu mới, cho phép mở rộng ngôn ngữ theo hướng cần giải quyết. C# có những từ khoá dành cho việc khai báo lớp, phương thức, thuộc tính (property) mới. C# hỗ trợ đầy đủ khái niệm trụ cột trong lập trình hướng đối tượng: đóng gói, thừa kế, đa hình.

Định nghĩa lớp trong C# không đòi hỏi tách rời tập tin tiêu đề với tập tin cài đặt như C++. Hơn thế, C# hỗ trợ kiểu sưu liệu mới, cho phép sưu liệu trực tiếp trong tập tin mã nguồn. Đến khi biên dịch sẽ tạo tập tin sưu liệu theo định dạng XML.

C# hỗ trợ khái niệm giao diện, *interfaces* (tương tự Java). Một lớp chỉ có thể kế thừa duy nhất một lớp cha nhưng có thế cài đặt nhiều giao diện.

C# có kiểu *cấu trúc,* struct (không giống C++). Cấu trúc là kiểu hạng nhẹ và bị giới hạn.Cấu trúc không thể thừa kế lớp hay được kế thừa nhưng có thể cài đặt giao diện.

C# cung cấp những đặc trưng lập trình hướng thành phần như property, sự kiện và dẫn hướng khai báo (được gọi là *attribute*). Lập trình hướng component được hỗ trợ bởi CLR thông qua siêu dữ liệu (metadata). Siêu dữ liệu mô tả các lớp bao gồm các phương thức và thuộc tính, các thông tin bảo mật ….

*Assembly* là một tập hợp các tập tin mà theo cách nhìn của lập trình viên là các thư viện liên kết động (DLL) hay tập tin thực thi (EXE). Trong .NET một assembly là một đon vị của việc tái sử dụng, xác định phiên bản, bảo mật, và phân phối. CLR cung cấp một số các lớp để thao tác với assembly.

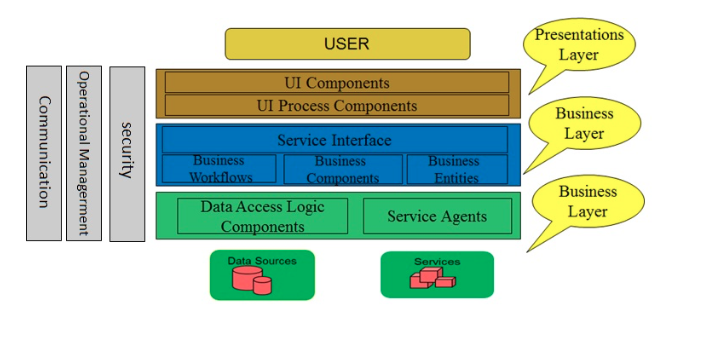
C# cũng cho truy cập trực tiếp bộ nhớ dùng con trỏ kiểu C++, nhưng vùng mã đó được xem như không an toàn. CLR sẽ không thực thi việc thu dọn rác tự động các đối tượng được tham chiếu bởi con trỏ cho đến khi lập trình viên tự giải phóng.

CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG TRONG ĐỀ TÀI

1. DevExpress:

* DevExpress là bộ control  rất hữu ích cho việc thiết kế và phát triển phần mềm, website,  đối với .NET nó thay thế hầu hết các control của bộ visual studio, nó không những giúp thiết kế được form đẹp hơn mà còn giúp cho công việc lập trình nhẹ nhàng hơn, nhất là trong việc tương tác dữ liệu.
* DevExpress được phát triển bởi Developer Express Inc (DevExpress) là công ty phát triển phần mềm được thành lập năm 1998, có trụ sở tại Glendale, California. Thời gian đầu, DevExpress phát triển UI Controls cho Borland Delphi/C++ Builder và ActiveX Controls cho Microsoft Visual Studio. Hiện nay thì sản phẩm của DevExpress hướng tới những lập trình viên sử dụng Delphi/c++ Builder, Visual Studio và HTML5/Javascript.
* Một số tính năng:
* Rating control: Rating control là một control mới được giới thiệu trong Version này, Rating Control giúp người dùng cuối tạo những feedback và những đánh giá
* TimeSpan Editor: TimeSpan Editor là thiết kế touch-friendly rất đơn giản và dễ dùng để người dùng có thể chọn thời gian chính xác tới đơn vị giây.
* SQL Data Access Component: Với phiên bản này bạn có thể sử dụng Winforms SqlDatasource mới để tương tác với database Sql, cấu hình kết nối dữ liệu và đưa dữ liệu vào các control như Gridcontrol, TreeList…
* Workspace Manager: Với Workspace Manager giúp người dùng cuối sao lưu và phục hồi không gian làm việc, layouts theo cách riêng của mình bao gồm Bar Managers, Ribbon Controls, Navigation Bars, Document và Dock Managers
* XtraGrid control: Gridcontrol có lẻ là control mà tôi ấn tượng nhất của DevExpress, không những nó giúp hiển thị dữ liệu kiểu bảng như DataGrid control của bộ Visual mà nó còn có nhiều tùy chọn khác khá hay
* Ngoài ra còn một số tính năng khác như:
* New Windows® 8 Inspired Grid TileView
* Excel® Inspired Conditional Formatting
* Excel Inspired Conditional Formatting
* New Excel® Data Export Engine
* Controls within Grid Cells

1. Sử dụng mô hình 3 layer



Mô hình 3-layer gồm có 3 phần chính :

– Presentation Layer (GUI) : Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liêu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

– Business Logic Layer (BLL) : Layer này phân ra 2 thành nhiệm vụ :

* Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
* Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

– Data Access Layer (DAL) : Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).

### Ưu điểm

* Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
* Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
* Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại  khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.
* Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
* Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.