

**SÁCH
TỰ HỌC**

DƯƠNG QUANG THIÊN
biên soạn

.NET Toàn tập

Tập 1

C# căn bản

và Visual Studio .NET IDE

Lập trình Visual C# thế nào ?

C# căn bản

C# và
.NET Framework

GUI và user control

ADO .NET

ASP .NET

Crystal Report



NHÀ XUẤT BẢN TỔNG HỢP TP.HCM

C# căn bản

và Visual Studio .NET IDE

Lập trình Visual C# thế nào?

DƯƠNG QUANG THIÊN

biên soạn

.NET Toàn tập

Tập 1

C# căn bản

và Visual Studio .NET IDE

Lập trình Visual C# thế nào?

Chịu trách nhiệm xuất bản:

TRẦN ĐÌNH VIỆT

Biên tập:

TRUNG HIẾU

Sửa bản in:

HỒNG HUÊ

Bìa:

HOÀNG NGỌC GIAO

*

NHÀ XUẤT BẢN TỔNG HỢP TP. HỒ CHÍ MINH

62 Nguyễn Thị Minh Khai - Q.1

ĐT: 8225340 - 8296764 - 8220405 - 8296713 - 8223637

Fax: 84.8.8222726 - Email: nxbtphcm@bdvn.vnd.net

In 1000 cuốn, khổ 16 x 22 cm, tại Xí nghiệp Cơ khí ngành in.
Giấy phép xuất bản số 399-191/XB-QLXB ký ngày 11-4-2003
In xong và nộp lưu chiểu tháng 1-2005.

Mục Lục

LỜI MỞ ĐẦU	17
Chương 1: <i>Visual C# và .NET Framework</i>	
1.1 Sàn diễn .NET	25
1.2 Tổng quan về .NET Framework	26
1.3 Việc gì xảy ra khi bạn biên dịch và chạy một chương trình	30
1.3.1 Biên dịch mã nguồn	31
1.3.2 Thi hành chương trình biên dịch	32
1.3.3 Những lợi điểm của đoạn mã được giám quản (managed code)	33
1.4 Intermediate Language (IL)	34
1.4.1 Lập trình thiên đối tượng cổ điển	35
1.4.2 Dữ liệu kiểu trị và kiểu qui chiếu	37
1.4.3 Kiểm tra chặt chẽ kiểu dữ liệu	38
1.5 Các cấu kiện của .NET Framework	39
1.5.1 Assembly	39
1.5.1.1 <i>Metadata và Manifest</i>	39
1.5.1.2 <i>Assembly được chia sẻ sử dụng hoặc riêng tư</i>	40
1.5.2 Namespaces	41
1.5.2.1 <i>Một vòng rào qua .NET Namespace</i>	42
1.5.3 Application Domain	45
1.5.4 Trình biên dịch JIT	47
1.5.5 .NET Tools	48
1.5.6 Garbage Collector	48
1.5.7 Biệt lệ (exceptions)	50
1.5.8 Attributes	51
1.5.9 Reflection	52
1.6 .NET Framework	52
1.6.1 Common Language Runtime (CLR)	55
1.6.2 Common Type System (CTS)	56
1.6.2.1 <i>Ý nghĩa của CTS đối với sự hợp tác liên ngôn ngữ</i> <i>(language interoperability)</i>	56
1.6.2.2 <i>Cây đẳng cấp CTS (CTS Hierarchy)</i>	57
1.6.3 Common Language Specification (CLS)	63
1.6.4 .NET Base Class Library	64
1.7 Ngôn ngữ C#	65

Chương 2: *Bắt đầu từ đây ta tiến lên!*

2.1 Chương trình C#, hình thù ra sao? -----	67
2.2 Lớp, Đối tượng và Kiểu dữ liệu -----	68
2.2.2 Các dòng chú giải (comments) -----	70
2.2.3 Các ứng dụng trên Console -----	71
2.2.4 Namespaces -----	72
2.2.5 Tác tử dấu chấm (dot operator) "." -----	73
2.2.6 Từ chốt using -----	73
2.2.7 Phân biệt chữ hoa chữ thường (case sensivity) -----	75
2.2.8 Từ chốt static -----	75
2.3 Các "biến tấu" khác nhau của hàm Main() -----	76
2.3.1 Xử lý các thông số của Command Line -----	77
2.3.2 Có nhiều hàm Main() -----	78
2.4 Giới thiệu cách định dạng chuỗi chữ C# -----	79
2.5 Triển khai "Xin Chào Bà Con!" -----	82
2.5.1 Hiệu đính "Xin Chào Bà Con!" -----	82
2.5.2 Cho biên dịch và chạy "Xin Chào Bà Con!" -----	85
2.6 Sơ liệu dựa trên XML -----	87
2.6.1 Hỗ trợ sơ liệu của Visual Studio .NET -----	91

Chương 3: *Sử dụng Debugger thế nào?*

3.1 Các điều cơ bản -----	93
3.1.1 Trắc nghiệm -----	94
3.1.2 Gỡ rối chương trình -----	95
3.1.2.1 Cơ bản về gỡ rối: Các chốt ngừng -----	96
3.1.3 Các công cụ gỡ rối dùng quan sát chương trình -----	98
3.1.3.1 DataTips -----	99
3.1.3.2 Các cửa sổ và khung đối thoại của Debugger -----	99
3.1.3.3 Sử dụng cửa sổ QuickWatch -----	100
3.1.3.4 Sử dụng cửa sổ Watch Window -----	102
3.1.3.5 Sử dụng cửa sổ Locals Window -----	104
3.1.3.6 Sử dụng cửa sổ Autos Window -----	105
3.1.3.7 Sử dụng cửa sổ This Window -----	107
3.1.3.8 Sử dụng cửa sổ Registers Window -----	108
3.1.3.9 Sử dụng cửa sổ Memory Window -----	111
3.1.3.10 Sử dụng cửa sổ Disassembly Window -----	115
3.1.3.11 Sử dụng cửa sổ Call Stack Window -----	117
3.2 Điều khiển việc thi hành chương trình -----	118

3.2.1	Bắt đầu gỡ rối	119
3.2.2	Ngắt thi hành (breaking execution)	120
3.2.3	Ngưng thi hành	121
3.2.4	Cho thi hành từng bước một (Stepping)	122
3.2.5	Cho chạy về một vị trí nhất định nào đó	123
3.2.6	Cho đặt để điểm thi hành	123
3.2.6.1	Nhảy về vị trí con nhảy	124
3.2.6.2	Chạy về một hàm được chỉ định	125
3.3	Chốt ngừng	126
3.3.1	Các loại chốt ngừng và thuộc tính	126
3.3.2	Cửa sổ Breakpoints Window	127
3.3.2.1	Sử dụng cửa sổ Breakpoints Window	129
3.3.3	Thuộc tính Hit Count	130
3.3.3.1	Khai báo hoặc thay đổi Hit Count	131
3.3.4	Thuộc tính Condition	132
3.3.4.1	Khai báo hoặc thay đổi điều kiện chốt ngừng	133
3.3.5	Chèn một chốt ngừng mới từ Debug	134
3.3.6	Gỡ bỏ tất cả các chốt ngừng	135
3.3.7	Các tác vụ chốt ngừng trên cửa sổ mã nguồn	135
3.3.8	Các tác vụ chốt ngừng trên cửa sổ Disassembly	137
3.3.9	Các tác vụ chốt ngừng trên cửa sổ Call Stack	139

Chương 4: *Căn bản Ngôn ngữ C#*

4.1	Kiểu dữ liệu (type)	141
4.1.1	Làm việc với kiểu dữ liệu bẩm sinh	144
4.1.1.1	Kiểu dữ liệu số nguyên (integer type)	145
4.1.1.2	Kiểu dữ liệu số dấu chấm di động (floating point number)	146
4.1.1.3	Kiểu dữ liệu số thập phân (decimal type)	146
4.1.1.4	Kiểu dữ liệu Bool	147
4.1.1.5	Kiểu dữ liệu ký tự	147
4.1.1.6	Chọn một kiểu dữ liệu bẩm sinh thế nào?	147
4.1.1.7	Chuyển đổi các kiểu dữ liệu bẩm sinh	149
4.2	Biến và Hằng	152
4.2.1	Gán rõ ràng (definite assignment)	154
4.2.2	Hằng (constant)	155
4.2.2.1	Một lớp hằng	157
4.2.3	Enumerations	158
4.2.3.1	Lớp cơ bản System.Enum	162
4.2.4	Các chuỗi chữ	163
4.2.5	Các diện tử (identifier)	164
4.2.6	Phạm vi hoạt động của biến (variable scope)	164
4.2.6.1	Vùng mục tin và Biến cục bộ	166
4.3	Biểu thức (expression)	167

4.4	Whitespace	167
4.5	Các câu lệnh (statements)	168
4.5.1	Các câu lệnh rẽ nhánh vô điều kiện	169
4.5.2	Các câu lệnh rẽ nhánh có điều kiện	170
4.5.2.1	Câu lệnh <i>If...else</i>	170
4.5.2.2	Các câu lệnh <i>if</i> lồng nhau	172
4.5.2.3	Câu lệnh <i>switch</i> : một phương án thay thế <i>if</i> nằm lồng	173
4.5.2.4	<i>Switch</i> trên các câu lệnh kiểu chuỗi chữ	176
4.5.3	Các câu lệnh rảo lặp (iteration)	176
4.5.3.1	Lệnh <i>goto</i>	176
4.5.3.2	Vòng lặp <i>while</i>	177
4.5.3.3	Vòng lặp <i>do ...while</i>	178
4.5.3.4	Vòng lặp <i>for</i>	179
4.5.3.5	Vòng lặp <i>foreach</i>	184
4.5.4	Các câu lệnh nhảy: <i>continue</i> , <i>break</i> và <i>return</i>	184
4.5.4.1	Câu lệnh <i>Continue</i>	184
4.5.4.2	Câu lệnh <i>Break</i>	184
4.5.4.3	Câu lệnh <i>Return</i>	187
4.6	Các tác tử (operators)	187
4.6.1	Tác tử gán (=)	187
4.6.2	Tác tử toán học	188
4.6.2.1	Tác tử số học đơn giản (+, -, *, /)	188
4.6.2.2	Tác tử <i>modulus</i> (%) để trả về số dư sau khi chia một số nguyên	188
4.6.3	Tác tử tăng giảm (++, --)	190
4.6.3.1	Tác tử tính toán và gán trở lại	190
4.6.3.2	Tác tử tiền tố và tác tử hậu tố (<i>prefix & postfix operator</i>)	191
4.6.4	Tác tử quan hệ	193
4.6.5	Sử dụng các tác tử lô gic với điều kiện	194
4.6.6	Các tác tử lô gic hoặc bitwise operator	194
4.6.7	Các tác tử kiểu dữ liệu (Type operator)	194
4.6.7.1	Tác tử <i>is</i>	194
4.6.7.2	Tác tử <i>sizeof</i>	195
4.6.7.3	Tác tử <i>typeof</i>	195
4.6.7.4	Tác tử <i>checked</i> và <i>unchecked</i>	196
4.6.8	Qui tắc "tôn ti trật tự" (Operator Precedence)	198
4.6.9	Tác tử tam phân (ternary operator)	200
4.7	Địa bàn hoạt động các tên (Namespaces)	201
4.7.1	Namespace Aliases	202
4.8	Các chỉ thị tiền xử lý (Preprocessor Directives)	203
4.8.1	Định nghĩa những diện từ	203
4.8.2	Không định nghĩa những diện từ	204
4.8.3	Các chỉ thị <i>#if</i> , <i>#elif</i> , <i>#else</i> , và <i>#endif</i>	204
4.8.4	Chỉ thị <i>#region</i>	205

Chương 5: Lớp và Đối tượng

5.1 Định nghĩa lớp -----	208
5.1.1 Từ chốt hướng dẫn truy xuất (Access Modifiers)-----	211
5.1.2 Các đối mục hàm hành sự (Method Arguments)-----	212
5.2 Tạo các đối tượng -----	213
5.2.1 Hàm khởi dựng (constructor) -----	213
5.2.2 Bộ khởi gán (Initializers) -----	216
5.2.3 Copy Constructors -----	218
5.2.4 Từ chốt this -----	219
5.3 Sử dụng các thành viên static-----	220
5.3.1 Triệu gọi các hàm static-----	221
5.3.2 Sử dụng hàm static constructor -----	222
5.3.3 Sử dụng Private Constructor -----	223
5.3.4 Sử dụng các vùng mục tin static-----	223
5.4 Hủy các đối tượng -----	224
5.4.1 C# Destructor-----	225
5.4.2 Hàm Finalize() đối nghịch với hàm Dispose()-----	225
5.4.3 Thiết đặt hàm hành sự Close -----	226
5.4.4 Sử dụng lệnh using -----	226
5.5 Trao thông số cho hàm-----	227
5.5.1 Trao thông số theo qui chiếu -----	228
5.5.2 Trao các thông số out với cách gán rõ ràng-----	230
5.6 Nạp chồng các hàm hành sự và hàm constructor-----	232
5.7 Gói ghém dữ liệu thông qua các thuộc tính -----	235
5.7.1 get Accessor -----	237
5.7.2 set Accessor-----	238
5.8 Các vùng mục tin read-only -----	238
5.9 Cuộc sống bên trong của các đối tượng -----	240
5.9.1 Thật sự một biến đối tượng (object variable) là gì? -----	240
5.9.2 Hàm hành sự instance và static -----	245
5.9.3 Truy xuất các thành viên static và instance-----	245
5.9.4 Các instance và static method được thiết đặt thế nào-----	246
5.9.5 Dưới căn lều thiên đối tượng-----	247
5.9.5.2 Sử dụng đối tượng-----	250
5.9.5.3 Trở lại vấn đề dữ liệu kiểu trị và kiểu qui chiếu -----	251

Chương 6: *Kế thừa và Đa hình*

6.1 Chuyên hóa (specialization) và Tổng quát (generalization)-----	254
6.2 Tính kế thừa (Inheritance) -----	256
6.2.2 Triệu gọi các hàm constructor của lớp cơ bản -----	259
6.2.3 Triệu gọi các hàm hành sự của lớp cơ bản -----	259
6.2.4 Kiểm soát việc truy xuất -----	260
6.2.5 Hướng dẫn sử dụng lớp cơ bản -----	261
6.3 Tính đa hình (polymorphisme)-----	261
6.3.1 Tạo những kiểu dữ liệu đa hình-----	261
6.3.2 Tạo các hàm hành sự đa hình -----	262
6.4 Các lớp trừu tượng (Abstract classes) -----	267
6.4.1 Những hạn chế của abstract -----	270
6.4.2 Các lớp "vô sinh" (Sealed class) -----	271
6.5 Nguồn gốc của tất cả các lớp: Object -----	271
6.6 Boxing và Unboxing các kiểu dữ liệu -----	274
6.6.1 Boxing được hiểu ngầm -----	274
6.6.2 Unboxing bắt buộc phải tường minh -----	275
6.7 Lớp lồng nhau-----	276

Chương 7: *Nạp chồng tác tử*

7.1 Sử dụng từ chốt operator -----	279
7.2 Hỗ trợ các ngôn ngữ .NET khác-----	280
7.3 Tạo những tác tử hữu ích -----	281
7.4 Cặp tác tử lô gic -----	281
7.5 Tác tử Equals -----	281
7.6 Chuyển đổi các tác tử -----	282

Chương 8: *Cấu trúc Struct*

8.1 Struct được khai báo thế nào? -----	289
8.1.1 Hàm khởi dựng và kế thừa-----	290