

Introducción a la programación con C#

Este texto es una introducción a la programación de ordenadores, usando el lenguaje C#.

Está organizado de una forma ligeramente distinta a los libros de texto "convencionales", procurando incluir ejercicios prácticos lo antes posible, para evitar que un exceso de teoría en los primeros temas haga el texto pesado de seguir.

Este texto ha sido escrito por Nacho Cabanes. Si quiere conseguir la última versión, estará en mi página web:

www.nachocabanes.com

Este texto es de **libre distribución** ("gratis"). Se puede distribuir a otras personas libremente, siempre y cuando no se modifique. Si le gustan los formalismos, esto sale a equivaler a una licencia **Creative Commons** BY-NC-ND: reconocimiento del autor, no se puede hacer uso comercial (no se puede "vender" este curso), no se puede crear obras derivadas.

Este texto se distribuye "tal cual", sin garantía de ningún tipo, implícita ni explícita. Aun así, mi intención es que resulte útil, así que le rogaría que me comunique cualquier error que encuentre.

Para cualquier sugerencia, no dude en contactar conmigo a través de mi web.

Contenido

0. Conceptos básicos sobre programación	7
0.1. Lenguajes de alto nivel y de bajo nivel.	7
0.2. Ensambladores, compiladores e intérpretes	9
0.3. Pseudocódigo	10
1. Toma de contacto con C#	11
1.1. Escribir un texto en C#	11
1.2. Cómo probar este programa	13
1.2.1. Cómo probarlo con Mono	13
1.2.2. Otros editores más avanzados	20
1.3. Mostrar números enteros en pantalla	21
1.4. Operaciones aritméticas básicas	21
1.4.1. Operadores	21
1.4.2. Orden de prioridad de los operadores	22
1.4.3. Introducción a los problemas de desbordamiento	22
1.5. Introducción a las variables: int	22
1.5.1. Definición de variables: números enteros	23
1.5.2. Asignación de valores	23
1.5.3. Mostrar el valor de una variable en pantalla	23
1.6. Identificadores	25
1.7. Comentarios	25
1.8. Datos por el usuario: ReadLine	26
1.9. Pequeñas mejoras	27
2. Estructuras de control	30
2.1. Estructuras alternativas	30
2.1.1. if	30
2.1.2. if y sentencias compuestas	31
2.1.3. Operadores relacionales: <, <=, >, >=, ==, !=	32
2.1.4. if-else	33
2.1.5. Operadores lógicos: &&, , !	35
2.1.6. El peligro de la asignación en un "if"	35
2.1.7. Introducción a los diagramas de flujo	36
2.1.8. Operador condicional: ?	38
2.1.10. switch	40
2.2. Estructuras repetitivas	43
2.2.1. while	43
2.2.2. do ... while	45
2.2.3. for	47
2.3. Sentencia break: termina el bucle	51
2.4. Sentencia continue: fuerza la siguiente iteración	52
2.5. Sentencia goto	54
2.6. Más sobre diagramas de flujo. Diagramas de Chapin.	55
2.7. El caso de "foreach"	57
2.8. Recomendación de uso para los distintos tipos de bucle	57
2.9. Una alternativa para el control errores: las excepciones	59
3. Tipos de datos básicos	62
3.1. Tipo de datos entero	62
3.1.1. Tipos de datos para números enteros	62
3.1.2. Conversiones de cadena a entero	62
3.1.3. Incremento y decremento	63

3.1.4. Operaciones abreviadas: +=	64
3.2. Tipo de datos real	64
3.2.1. Simple y doble precisión	65
3.2.2. Pedir y mostrar números reales	65
3.2.3. Formatear números	67
3.2.4. Cambios de base	68
3.3. Tipo de datos carácter	70
3.3.1. Leer y mostrar caracteres	70
3.3.2. Secuencias de escape: \n y otras	71
3.4. Toma de contacto con las cadenas de texto	73
3.5. Los valores "booleanos"	74
4. Arrays, estructuras y cadenas de texto	76
4.1. Conceptos básicos sobre arrays o tablas	76
4.1.1. Definición de un array y acceso a los datos	76
4.1.2. Valor inicial de un array	77
4.1.3. Recorriendo los elementos de una tabla	78
4.1.4. Datos repetitivos introducidos por el usuario	79
4.1.5. Operaciones habituales: buscar, añadir, insertar, borrar	80
4.2. Tablas bidimensionales	83
4.3. Estructuras o registros	85
4.3.1. Definición y acceso a los datos	85
4.3.2. Arrays de estructuras	87
4.3.3. Estructuras anidadas	88
4.4. Cadenas de caracteres	89
4.4.1. Definición. Lectura desde teclado	89
4.4.2. Cómo acceder a las letras que forman una cadena	90
4.4.3. Longitud de la cadena.	90
4.4.4. Extraer una subcadena	90
4.4.5. Buscar en una cadena	91
4.4.6. Otras manipulaciones de cadenas	92
4.4.7. Comparación de cadenas	94
4.4.8. Una cadena modificable: StringBuilder	96
4.4.9. Recorriendo con "foreach"	97
4.5 Ejemplo completo	98
4.6 Ordenaciones simples	101
5. Introducción a las funciones	107
5.1. Diseño modular de programas: Descomposición modular	107
5.2. Conceptos básicos sobre funciones	107
5.3. Parámetros de una función	109
5.4. Valor devuelto por una función. El valor "void".	110
5.5. Variables locales y variables globales	112
5.6. Los conflictos de nombres en las variables	114
5.7. Modificando parámetros	115
5.8. El orden no importa	118
5.9. Algunas funciones útiles	118
5.9.1. Números aleatorios	118
5.9.2. Funciones matemáticas	120
5.9.3. Pero hay muchas más funciones...	121
5.10. Recursividad	121
5.11. Parámetros y valor de retorno de "Main"	123
6. Programación orientada a objetos	126
6.1. ¿Por qué los objetos?	126

6.2. Objetos y clases en C#	127
6.3. La herencia. Visibilidad	132
6.4. ¿Cómo se diseñan las clases?	136
6.5. Constructores y destructores.	137
6.6. Polimorfismo y sobrecarga	140
6.7. Orden de llamada de los constructores	140
6.8. Proyectos a partir de varios fuentes: SharpDevelop, Visual Studio y MonoDevelop	142
7. Utilización avanzada de clases	150
7.1. La palabra "static"	150
7.2. Arrays de objetos	151
7.3. Funciones virtuales. La palabra "override"	154
7.4. Llamando a un método de la clase "padre"	159
7.5. La palabra "this": el objeto actual	161
7.6. Sobrecarga de operadores	162
8. Manejo de ficheros	163
8.1. Escritura en un fichero de texto	163
8.2. Lectura de un fichero de texto	164
8.3. Lectura hasta el final del fichero	165
8.4. Añadir a un fichero existente	166
8.5. Ficheros en otras carpetas	167
8.6. Saber si un fichero existe	167
8.7. Más comprobaciones de errores: excepciones	168
8.8. Conceptos básicos sobre ficheros	171
8.9. Leer datos básicos de un fichero binario	171
8.10. Leer bloques de datos de un fichero binario	173
8.11. La posición en el fichero	174
8.12. Escribir en un fichero binario	176
8.13. Ejemplo: leer información de un fichero BMP	180
8.14. Leer y escribir en un mismo fichero binario	184
9. Persistencia de objetos	187
9.1. ¿Por qué la persistencia?	187
9.2. Creando un objeto "serializable"	188
9.3. Empleando clases auxiliares	190
9.4. Volcando a un fichero de texto	196
10. Acceso a bases de datos relacionales	201
10.1. Nociones mínimas de bases de datos relacionales	201
10.2. Nociones mínimas de lenguaje SQL	201
10.2.1. Creando la estructura	201
10.2.2. Introduciendo datos	202
10.2.3. Mostrando datos	202
10.3. Acceso a bases de datos con SQLite	203
10.4. Un poco más de SQL: varias tablas	207
10.4.1. La necesidad de varias tablas	207
10.4.2. Las claves primarias	208
10.4.3. Enlazar varias tablas usando SQL	209

10.4.4. Varias tablas con SQLite desde C#	211
10.5. Borrado y modificación de datos	214
10.6. Operaciones matemáticas con los datos	214
10.7 Grupos	215
10.8 Un ejemplo completo con C# y SQLite	216
10.9. Nociones mínimas de acceso desde un entorno gráfico	219
11. Punteros y gestión dinámica de memoria	221
11.1. ¿Por qué usar estructuras dinámicas?	221
11.2. Una pila en C#	222
11.3. Una cola en C#	223
11.4. Las listas	225
11.4.1. ArrayList	225
11.4.2. SortedList	227
11.5. Las "tablas hash"	229
11.6. Los "enumeradores"	231
11.7. Cómo "imitar" una pila usando "arrays"	233
11.8. Introducción a los "generics"	234
11.9. Los punteros en C#.	235
11.9.1. ¿Qué es un puntero?	235
11.9.2. Zonas "inseguras": unsafe	236
11.9.3. Uso básico de punteros	237
11.9.4. Zonas inseguras	238
11.9.5. Reservar espacio: stackalloc	238
11.9.6. Aritmética de punteros	239
11.9.7. La palabra "fixed"	241
12. Algunas bibliotecas adicionales de uso frecuente	243
12.1. Fecha y hora. Temporización	243
12.2. Más posibilidades de la "consola"	245
12.3. Lectura de directorios	248
12.4. El entorno. Llamadas al sistema	249
12.5. Datos sobre "el entorno"	250
12.6. Algunos servicios de red.	251
12.7. Nociones básicas de entornos gráficos	254
12.7.1 Contacto con Visual Studio 2010 Express	254
12.7.2 Botones, etiquetas, casillas de texto	258
12.7.3. Usando ventanas predefinidas	261
12.7.4. Una aplicación con dos ventanas	263
12.7.5. Otros componentes visuales	266
12.7.6. Dibujando con Windows Forms	267
12.8. Juegos con Tao.SDL	268
12.8.1. Mostrar una imagen estática	269
12.8.2. Una imagen que se mueve con el teclado	271
12.8.3. Simplificando con clases auxiliares	273
12.8.4. Un fuente más modular: el "bucle de juego"	277
12.8.5. Escribir texto	279
12.8.6. Colisiones simples	281
12.8.7. Imágenes PNG y JPG	285
12.8.8. ¿Por dónde seguir?	285
13. Otras características avanzadas de C#	287
13.1. Espacios de nombres	287
13.2. Operaciones con bits	289

Esta es una muestra, haga clic en el enlace de descarga para obtener el tutorial completo

