

13/07/98

29431

Compra

Prólogo	xxi
---------------	-----

**PARTE I
EL LENGUAJE C**

1. Introducción al lenguaje C	3
Orígenes del lenguaje C	3
Un lenguaje de nivel medio	4
Un lenguaje estructurado	5
Un lenguaje para programadores	6
Compiladores frente a intérpretes	8
Formato de un programa en C	8
Bibliotecas y enlace	8
Compilación separada	10
Mapa de memoria de un programa en C	11
Repaso de terminología	12
2. Variables, constantes, operadores y expresiones	15
Nombres de identificadores	15
Tipos de datos	16
Modificadores de tipo	16
Modificadores de acceso	18

Declaración de variables	19
Variables locales	20
Parámetros formales	22
Variables globales	22
Especificadores de clase de almacenamiento	24
extern	24
Variables estáticas (static)	26
Variables locales estáticas (static)	26
Variables globales estáticas (static)	28
Variables register	29
Instrucciones de asignación	30
Asignaciones múltiples	30
Conversión de tipo en asignaciones	30
Inicialización de variables	32
Constantes	33
Constantes de caracteres de barra invertida	33
Operadores	33
Operadores aritméticos	34
Incremento y decremento	35
Operadores relacionales y lógicos	36
Operadores a nivel de bit	38
El operador ?	42
Los operadores de puntero & y *	43
El operador en tiempo de compilación sizeof	45
El operador coma	45
Los operadores . y ->	46
[] y ()	47
Resumen de precedencia	47
Expresiones	47
Conversión de tipo en expresiones	47
Conversiones de tipo	49
Espaciado y paréntesis	50
Abreviaturas de C	50
3. Instrucciones de control de programa	53
Verdadero y falso en C	53
Instrucciones de selección	54
if	54
Ifs anidados	55
if-else-if escalonados	56
La alternativa ?	57
switch	59
Instrucciones switch anidadas	63

Bucles	63
for	63
Variaciones del bucle for	65
El bucle infinito	68
Bucles for sin cuerpo	68
while	69
do/while	71
break	72
exit()	73
continue	75
Etiquetas y goto	76
4. Funciones	79
La instrucción return	80
Salidas de una función	80
Tipos de Valores devueltos	81
Devolución de valores desde main()	83
Reglas de ámbito de las funciones	83
Argumentos de las funciones	83
Llamada por valor y llamada por referencia	84
Creación de una llamada por referencia	85
Llamada a funciones con arrays	86
Argumentos de main()	90
Funciones que devuelven valores no enteros	95
Prototipos de funciones	97
Prototipos de funciones de biblioteca estándar	99
Prototipos de funciones sin parámetros	99
Devolución de punteros	101
Declaración de parámetros clásica y moderna	102
Recursividad	103
Punteros a funciones	105
Aspectos de implementación	107
Parámetros y funciones de propósito general	107
Eficiencia	108
5. Arrays	111
Arrays unidimensionales	111
Generación de un puntero a un array	113
Paso de arrays unidimensionales a funciones	113
Cadenas	115
Arrays bidimensionales	116
Arrays de cadenas	120
Arrays multidimensionales	121

Indexación de punteros	122
Asignación dinámica de arrays	124
Inicialización de arrays	128
Inicialización de arrays sin límites predefinidos	130
Un ejemplo de tres en raya	131
6. Punteros	135
Los punteros son direcciones	135
Variables puntero	136
Los operadores de punteros	136
Expresiones de punteros	138
Asignación de punteros	138
Aritmética de punteros	139
Comparación de punteros	140
Asignación dinámica y punteros	142
Introducción a los punteros const	144
Punteros y arrays	145
Punteros a arrays de caracteres	146
Arrays de punteros	148
Punteros a punteros: Direccionamiento indirecto múltiple	149
Inicialización de punteros	151
Punteros a funciones	152
Problemas con los punteros	155
7. Estructuras, uniones y tipos definidos por el usuario	157
Estructuras	157
Referencia a los miembros de una estructura	160
Asignaciones de estructuras	160
Arrays de estructuras	161
Un ejemplo de inventario	161
Paso de estructuras a funciones	168
Paso de miembros de estructura a funciones	168
Paso de estructuras completas a funciones	169
Punteros a estructuras	169
Declaración de un puntero a una estructura	170
Utilización de punteros a estructuras	170
Arrays y estructuras dentro de estructuras	173
Campos de bits	174
Uniones	177
Enumeraciones	179
Utilización de sizeof para asegurar la portabilidad	182
typedef	184

8. Entrada, salida, flujos y archivos	185
E/S en C y C++	186
Flujos (Streams) y archivos	186
Flujos (Streams)	187
Archivos	187
E/S por pantalla	189
Lectura y escritura de caracteres	189
Lectura y escritura de cadenas: gets() y puts()	192
E/S formateada por pantalla	193
printf()	194
scanf()	201
El sistema de archivos del estándar ANSI de C	208
El puntero a archivo	209
Apertura de un archivo	210
Escritura de un carácter	211
Lectura de un carácter	212
fclose()	212
Utilización de fopen(), getc(), putc() y fclose()	213
Utilización de feof()	214
Dos funciones adicionales: getw() y putw()	216
Manejo de cadenas: fgets() y fputs()	216
fread() y fwrite()	217
fseek() y E/S de acceso directo	219
fprintf() y fscanf()	221
Eliminación de archivos	224
ferror() y frewind()	225
Conexión con la pantalla	225
9. El preprocesador y los comentarios	227
#define	228
#error	231
#include	232
Directivas de compilación condicional	233
#if, #else, #elif y #endif	233
#ifdef e #ifndef	235
#undef	236
Utilización de defined	237
#line	237
#pragma	238
Los operadores del preprocesador # y ##	241
Nombres de macro predefinidas	243
Comentarios	245

**PARTE II
LA BIBLIOTECA DE FUNCIONES DE BORLAND**

10. Enlace, bibliotecas y archivos de cabecera	249
El enlazador	249
Archivos de biblioteca frente a archivos objeto	251
La biblioteca estándar ANSI de C frente a las extensiones de Borland	251
Archivos de cabecera	252
Macros en archivos de cabecera	253
11. Funciones de E/S	255
12. Funciones para el manejo de cadenas, memoria y caracteres	325
13. Funciones matemáticas	359
14. Funciones de fecha, de hora y del sistema	379
15. Asignación dinámica	447
16. Funciones de directorios	473
17. Funciones de control de procesos	491
18. Funciones gráficas y de texto	501
19. Funciones diversas	581

**PARTE III
BORLAND C++**

20. Introducción al C++	629
Orígenes del C++	629
¿Qué es la programación orientada a objetos?	630
Encapsulación	631
Polimorfismo	632
Herencia	632
Algunos fundamentos de C++	633
Compilación de un programa en C++	635
Introducción a las clases de C++	636
Sobrecarga de funciones	640
Sobrecarga de operadores	643
Herencia	643
Constructores y destructores	647
Palabras reservadas en C++	650
Dos nuevos tipos de datos	651

21. Estudio de clases y objetos	653
Constructores parametrizados	653
Funciones afines	657
Argumentos de funciones por omisión	661
Utilización correcta de los argumentos por omisión	665
Relación entre clases y estructuras	665
Relación entre clases y uniones	667
Uniones anónimas	668
Funciones tipo inline	669
Creación de funciones en línea dentro de una clase	671
Paso de objetos a funciones	672
Funciones que devuelven objetos	674
Asignación de objetos	675
Arrays de objetos	676
Inicialización de arrays de objetos	677
Creación de arrays inicializados frente a no inicializados	679
Punteros a objetos	679
22. Sobrecarga de funciones y operadores	683
Sobrecarga de funciones constructor	683
Ubicación de las variables	685
Ubicación de la creación de objetos	686
Sobrecarga de funciones y ambigüedad	688
Localización de la dirección de una función sobrecargada	691
El puntero this	692
Sobrecarga de operadores	693
Funciones operador afines	700
Referencias	704
Parámetros por referencia	704
Paso de referencias a objetos	707
Funciones que devuelven referencias	708
Referencias independientes	709
Utilización de una referencia para sobrecargar un operador unario ..	711
Sobrecarga de []	714
Aplicación de la sobrecarga de operadores	717
23. Herencia, funciones virtuales y polimorfismo	723
Herencia y especificadores de acceso	723
Definición de los especificadores de acceso	724
Control de acceso a la clase base	725
Constructores y destructores en clases derivadas	728
Herencia múltiple	731

Paso de parámetros a una clase base	734
Punteros y referencias a tipos derivados	735
Referencias a tipos derivados	738
Funciones virtuales	738
¿Por qué utilizar funciones virtuales?	743
Funciones virtuales puras y tipos abstractos	747
Vinculación a priori y a posteriori	749
24. La biblioteca de clases de E/S de C++	751
¿Por qué C++ tiene su propio sistema de E/S?	751
Flujos en C++	752
Los flujos predefinidos en C++	752
Las clases de flujos de C++	753
Creación de insertores y extractores personales	753
Creación de insertores	754
Sobrecarga de extractores	757
Formato de E/S	758
Formatos utilizando las funciones miembro de ios	758
Utilización de los manipuladores	763
Creación de funciones manipuladoras personales	765
Creación de manipuladores sin parámetros	765
Creación de manipuladores parametrizados	767
E/S en archivos	770
Apertura y cierre de un archivo	770
Lectura y escritura en archivos de texto	773
E/S binaria	774
Detección de EOF	777
Acceso aleatorio	777
25. E/S basada en arrays	781
Clases basadas en arrays	781
Creación de un flujo de salida basado en arrays	782
Utilización de un array como entrada	784
Utilización de las funciones miembro ios en flujos basados en arrays	785
Flujos de entrada/salida basados en arrays	786
Acceso aleatorio en arrays	787
Utilización de arrays dinámicos	788
Manipuladores y E/S basada en arrays	789
Extractores e insertores personalizados	789
Utilización del formateo basado en arrays	791

26. Plantillas, excepciones y RTTI	793
Funciones genéricas	793
Una función con dos tipos genéricos	795
Sobrecarga explícita de una función genérica	796
Limitaciones de las funciones genéricas	797
Clases genéricas	798
Un ejemplo con dos tipos de datos genéricos	801
Manejo de excepciones	802
Fundamento del manejo de excepciones	803
Utilización de varias instrucciones catch	807
Opciones para el manejo de excepciones	808
Captura de todas las excepciones	809
Restricción de las excepciones	810
Renotificación de una excepción	811
Aplicación del manejo de excepciones	812
Identificación de tipos en tiempo de ejecución	814
Nuevos operadores de conversión	816
27. Temas diversos de C++	819
Asignación dinámica usando new y delete	819
Asignación de memoria a objetos	822
Otra forma de controlar los fallos en la asignación de memoria	824
Sobrecarga de new y delete	825
Miembros static de una clase	826
Clases base virtuales	830
Funciones miembro const y volatile	834
Utilización de la palabra reservada asm	834
Especificación del enlace	835
Los operadores .* y ->*	836
Creación de funciones de conversión	838
Constructores de copia	840
Concesión de acceso	843
Utilización de espacios de nombres	844
Algunas modificaciones recientes	845
Nuevas cabeceras	845
Constructores explícitos	846
Utilización de mutable	847
typename	847
La biblioteca de plantillas estándar	847
Diferencias entre C y C++	850
Consideraciones finales	850

**PARTE IV
EL ENTORNO INTEGRADO DE DESARROLLO
DE BORLAND C++**

28. El entorno integrado de desarrollo	855
La ventana principal del EID	856
La barra de menús	856
Exploración de la barra de menús	857
File (Archivo).....	857
Edit (Edición).....	859
Search (Buscar)	859
View (Ver)	859
Project (Proyecto)	859
Script.....	860
Tool (Herramienta)	860
Debug (Depurar)	860
Options (Opciones).....	860
Window (Ventana).....	861
Help (Ayuda).....	862
Utilización de la ayuda sensible al contexto	863
La barra de herramientas (SpeedBar)	863
La barra de estado	863
Utilización de los menús rápidos (SpeedMenus).....	863
Utilización de scripts	864
Consideraciones sobre la creación de proyectos y la compilación de programas	864
 29. Utilización del editor	 867
Entrada en el editor e introducción de texto	867
Supresión de caracteres, palabras y líneas	869
Desplazamiento, copia y supresión de bloques de texto	869
Utilización del portapapeles	871
Desplazamiento avanzado del cursor.....	872
Buscar y sustituir	872
Almacenamiento y lectura de archivos.....	875
Descripción de la sangría automática	875
Escritura y lectura de bloques de texto en archivos de disco	876
Correspondencia de delimitadores	876
Ordenes varias	876
Utilización del menú rápido (SpeedMenu)	877
Modificación de la configuración del editor	877
Resumen de órdenes de teclado	877

30. Utilización del entorno integrado de depuración de Borland C++	881
Preparación de los programas para su depuración	881
¿Que es un depurador a nivel de código fuente?	882
Fundamentos de la depuración	882
Ejecución paso a paso	883
Puntos de ruptura	884
Definición de puntos de ruptura incondicionales en el código fuente (tipo Source)	885
Definición de puntos de ruptura condicionales en el código fuente (tipo Source)	887
Visualización de variables	888
Códigos de formato de las expresiones a visualizar	889
Cualificación del nombre de una variable	892
Inspección de la pila	893
Evaluación de una expresión	894
Modificación de una variable	894
Inspección de una variable	895
Interrupción de un programa	895
Utilización de la ventana CPU	895
Algunos consejos para la depuración	896

**PARTE V
INTRODUCCIÓN A LA PROGRAMACIÓN
EN WINDOWS 95**

31. Introducción a Windows 95	899
¿Qué es Windows 95?	899
Utilización de la multitarea basada en elementos de proceso de Windows 95	900
La interfaz basada en llamadas de Windows 95	900
Bibliotecas de enlace dinámico (DLL, Dynamic Link Libraries)	901
Windows 95 frente a Windows 3.1	901
Diferencias a nivel de usuario	902
Diferencias a nivel de programación	903
La conexión NT	906
Los programas de Windows 95 son únicos	906
Interacción entre el programa y Windows 95	907
Win32: La API de Windows 95	907
Componentes de una ventana	908

Conceptos básicos sobre las aplicaciones de Windows 95	909
WinMain()	909
La función de ventana	910
Clases de ventana	910
Bucle de mensajes	911
Tipos de datos en Windows	911
Estructura de un programa en Windows 95	911
Compilación de la plantilla	914
Estudio detallado de la estructura básica de un programa	914
Definición de la clase de ventana	916
Creación de una ventana	918
Bucle de mensajes	920
La función de ventana	921
¿Qué sucede con los archivos de definición?	922
Convenios de nomenclatura	922
32. Conceptos básicos sobre aplicaciones: Mensajes y E/S básica	925
Cuadros de mensajes	925
Descripción de los mensajes de Windows 95	928
Respuesta a la pulsación de una tecla	929
Escritura de texto en una ventana	932
Contextos de dispositivo	937
Procesamiento del mensaje WM_PAINT	938
Generación de un mensaje WM_PAINT	943
Respuesta a mensajes del ratón	946
Más información sobre los mensajes del ratón	949
33. Utilización de menús	951
Introducción a los menús	951
Utilización de recursos	952
Compilación de archivos .RC	953
Creación de un menú sencillo	953
Inclusión de un menú en el programa	956
Respuesta a las selecciones del menú	956
Ejemplo de programa con un menú	957
Descripción del editor de recursos de Borland	960
Definición de abreviaturas de teclado para menús	961
Carga de la tabla de abreviaturas	963
Gestión dinámica de un menú	966
Incorporación de un elemento a un menú	966
Eliminación de un elemento de un menú	967
Obtención de un descriptor de un menú	968
Obtención del tamaño de un menú	968

Cómo activar y desactivar un elemento de un menú	969
Ejemplo de la gestión dinámica de menús	969
Creación de menús dinámicos	975
34. Cuadros de diálogo	981
Utilización de controles en cuadros de diálogo	981
Cuadros de diálogo modales y no modales	982
Recepción de mensajes en cuadros de diálogo	982
Cómo activar un cuadro de diálogo	983
Cómo desactivar un cuadro de diálogo	983
Creación de un cuadro de diálogo simple	984
El archivo de recursos de cuadros de diálogo	984
La función de ventana del cuadro de diálogo	987
Primer ejemplo de programa con un cuadro de diálogo	988
Incorporación de un cuadro de lista	992
Conceptos básicos sobre cuadros de lista	993
Inicialización del cuadro de lista	995
Procesamiento de una selección	996
Incorporación de un cuadro de edición	997
Programa completo de un cuadro de diálogo modal	1000
Utilización de un cuadro de diálogo no modal	1005
Creación de un cuadro de diálogo no modal	1007
Un comentario	1014

**PARTE VI
INTRODUCCIÓN AL ENTORNO JAVA**

35. Introducción a Java	1017
¿Qué es Java?	1017
¿Por qué utilizar Java?	1018
Seguridad	1018
Portabilidad	1019
La magia de Java: los bytecodes	1019
Ventajas claves de Java:	1020
Sencillez	1021
Orientado a objetos	1021
Robustez	1021
Entrelazado	1021
Neutralidad frente a la arquitectura	1022
Interpretado y con alto rendimiento	1022
Entorno distribuido	1022
Dinámico	1022

Diferencias entre Java y C++	1023
Elementos de C++ ausentes en Java	1023
Nuevas características incorporadas por Java	1024
Características diferentes	1025
Aplicaciones y applets (pequeñas aplicaciones) en Java	1026
Métodos frente a funciones	1026
Una aplicación sencilla en Java	1026
Escritura del programa	1027
Compilación y ejecución de un programa Java	1027
Examen detallado a la primera aplicación	1028
Un segundo ejemplo	1030
Algunos fundamentos de Java	1032
Java es un lenguaje con tipos estrictos	1032
Tipos simples predefinidos en Java	1033
Literales de cadena	1034
Conversiones de tipo	1034
Operadores	1035
Instrucciones de control	1036
Conceptos básicos sobre clases	1036
Ejemplo de clase sencilla en Java	1038
Declaración de objetos	1039
Asignación de variables de referencia a objetos	1040
Incorporación de un método y un constructor	1041
Arrays	1042
Arrays unidimensionales	1043
Arrays multidimensionales	1045
Recuperación de espacio (Garbage collection)	1046
El método finalize()	1047
36. Herencia, paquetes e interfaces	1049
Herencia	1050
Llamadas a los constructores	1052
Utilización de la palabra reservada super	1052
Utilización de super para llamar a constructores de una super- clase	1053
Otra utilización de super	1058
Anulación de métodos y gestión dinámica	1059
Gestión dinámica	1060
Métodos y clases abstractas	1062
Utilización de la palabra reservada final	1063
Utilización de final para evitar la anulación	1063
Utilización de final para evitar la herencia	1064
Utilización de final para crear constantes con nombre	1064

La clase Object.....	1065
Paquetes	1065
Definición de un paquete.....	1065
Descripción de CLASSPATH.....	1066
Ejemplo de un paquete sencillo	1067
Importación de paquetes.....	1068
Los paquetes estándar	1069
Control de acceso y paquetes	1070
Interfaces	1071
Definición de una interfaz	1071
Implementación de una interfaz	1072
Ampliación de una interfaz	1073
Acceso a las implementaciones mediante referencias a la interfaz	1074
Aplicación de interfaces	1074
37. Introducción a las applets (pequeñas aplicaciones).....	1079
Un ejemplo de applet.....	1079
Compilación y visualización de una applet.....	1080
La clase Applet	1083
La arquitectura de una applet	1083
Plantilla de una applet	1084
Orden de inicialización y finalización de las applets	1085
El método update()	1087
Solicitud para volver a dibujar la pantalla	1087
Utilización de la ventana de estado	1088
Control de sucesos	1089
La clase Event	1089
Procesamiento de los sucesos del ratón.....	1090
Control de los sucesos del teclado	1093
Otros sucesos.....	1095
Temas a explorar.....	1095
Más información sobre Java.....	1096
Índice	1097