

INDICE

Reconocimientos	XVII
Prefacio	XIX
Parte I. Introducción a la Programación Orientada a Objetos	1
1. ¿Por qué Aprender Programación Orientadas a Objetos?	3
La arquitectura del software	3
Evolución del diseño	4
Las posibilidades de la arquitectura orientada a objetos	5
Aprendizaje de lo Básico	8
Los objetos por si mismo no proporcionan una estructura	9
Construcción de sistemas a partir de objetos	9
Compresión, diseño y construcción de sistemas complejos	9
2. Aspectos Básicos de la Programación Orientada a Objetos	11
¿Qué es la Programación orientada a objetos?	11
Polimorfismo, herencia y encapsulado	12
Estructura de un programa	13
Objetos	19
Clases	21
Comunicación	21
Procesamiento	28
¿Por qué es importante la programación orientada a objetos?	30
Puntos principales del capítulo	31
3. Herencia, Polimorfismo y Programación Orientada a Objetos	33
Colecciones polimórficas	34
Clasificación	37
Herencia	44
Polimorfismo	52
Diferencias entre polimorfismo y herencia	59
El polimorfismo generaliza el método de despacho	60
Nexos dinámicos	66
Rotación de objetos Shape polimórficos	77
Puntos principales del capítulo	86
Clasificación	86
Herencia	87
Polimorfismo y nexos dinámicos	87
4. Hablemos de Smalltalk	89
Smalltalk Comparado Con C++	89
El ambiente de programación de Smalltalk	90
Uso del intérprete interactivo de Smalltalk	91
Introducción y selección de texto de programa para evaluación	93
Evaluación de texto seleccionados	94
Programas de proposiciones múltiples	95
Variables temporales	96
Valores resultantes o de regreso	97
Trabajo con múltiples ventanas	100
Concatenación de mensajes enviados al mismo objeto	106
Ambientes de programación para lenguajes compilados	107
Apoyos durante la ejecución, para programas ejecutables	108

5. Bibliotecas de Clase	109
La biblioteca de clase Smalltalk	111
Colecciones y recipientes	112
Bolsas y conjuntos	112
Diccionarios	123
Diccionarios e iteradores	133
Colecciones ordenadas	137
Colecciones ordenadas y colecciones clasificadas	137
Colecciones ordenadas y colecciones clasificadas	145
Arreglos	146
Conversión de colecciones	150
Conteo y selección de los elementos de una colección	153
Colecciones que contiene colecciones	157
6. Encapsulado, Herencia y Polimorfismo en Acción	165
Cómo minimizar el impacto del cambio	165
Diseño e instrumentación de una clase abstracta screen	167
La responsabilidad del usuario	172
Un archivo make para construir los programas para rotar formas	178
La responsabilidad del constructor	184
Instrumentación de Screen para zortech flash graphics	190
Instrumentación de la pantalla de Microsoft Windows	194
Programación de aplicaciones en Microsoft Windows	195
Aplicaciones Windows	196
Instancia Windows	196
La función WinMain para aplicaciones Microsoft Windows	196
Manejo de mensajes	197
Asas de ventana y asas de instancia	203
Asas para contextos de exhibición	204
Puntos principales del capítulo	221
Parte II. Herramientas de la Programación Orientada a Objetos	
7. Apoyo del Lenguaje a Tipos Definidos por el Usuario	225
¿Qué hace que in lenguaje sea orientado a objetos?	226
Abstracción de datos	226
Clasificación jerárquica de objetos	227
Tipos definidos por el usuario en C++ y en Pascal Orientado a Objetos	228
Tipos internos	229
Tipos definidos por el usuario (abstracción de datos)	231
Ocultamiento de datos	237
Inicialización automática de objetos (constructores)	244
Limpiando automático de objetos (destrutores)	250
Sobrecarga de operadores	257
¿Qué tan bien C++ y el Pascal orientado a objetos apoyan la programación con tipos definidos por el usuario?	261
Tipos definidos por el usuario en Smalltalk y actor	263
Tipos internos	265
Tipos definidos por el usuario	265
Ocultamiento de datos	276
Inicialización automática de objetos	281

Limpiando automático de objetos	289
Nombre de método y sobrecarga de operadores en Smalltalk	290
¿Qué tan bien apoyan Smalltalk y actor a la programación por el usuario?	294
Puntos principales del capítulo	295
8. Apoyo del Lenguaje para la Programación Orientada a Objetos	297
La programación basada en objetos comparada con la programación orientada a objetos	298
Técnicas de la programación orientada a objetos	299
Clasificación jerárquica de objetos	301
Las clases son tipos	304
Paso de mensaje polimórficos	304
Características de un lenguaje orientado a objetos	310
La herencia específica jerarquía	310
Los nexos dinámicos habilitan el polimorfismo	311
Clasificación con herencia	311
Búsqueda de propiedades compartidas	312
Especificación de herencia	316
Programación con polimorfismo	335
Especificación de nexos dinámicos	335
Las colecciones polimórficas responden a mensajes polimórficos	336
Puntos principales del capítulo	341
Parte III. Bibliotecas de Clase y Marcos de Aplicación	
9. Características Generales de un Marco de Aplicación	345
Los marcos de aplicación son bibliotecas de clase	346
Programa = Aplicación = Modelo	348
Componentes de programa	351
Ventanas y vistas	351
Vistas de datos	361
Estructuras y organización de un programa	361
Separación de la aplicación de vistas y controles	361
Modelo – vistas – controlador	362
Mecanismos de control comparados con componentes de control	364
Control manejado por eventos	365
Agrupamiento de eventos	365
Ciclo principal de eventos	368
Manejo de eventos	368
Hola otra vez, Mundo	368
Puntos Principales del Capítulo	372
10. Turbo Visión: Construcción de menús para un marco de Aplicación Orientado a Caracteres	375
Creación de un menú simple	377
Menús múltiples con múltiples opciones	381
Menús y eventos	383
Despachos de eventos	385
Manejo de eventos	386
Características Especiales de Menús	389
Teclas abreviadas	390
Teclas aceleradoras	391

Nombres de cadena de teclas aceleradoras	392
Uso de constantes simbólicas para eventos de comando	393
Puntos principales del capítulo	397
11. C++ Views: Menús y Eventos de Comando	399
Construcción de un marco de menú	400
Creación de un menú simple	404
Menús múltiples con múltiples opciones	411
Menús y eventos	414
Despacho de eventos	414
Manejo de eventos	415
Características especiales de Menús	415
Teclas abreviadas	416
Teclas aceleradoras	417
Nombres de cadena de teclas aceleradoras	418
Separadores de línea de menú	419
Puntos principales del capítulo	426
12. C++ Views: Diálogo y Vistas de Control	429
Un marco para probar casillas de diálogo	430
El menú, los elementos de menú y los apuntadores de método	432
Métodos para probar elementos de menú	434
Uso de diálogos que proporciona C++ Views	435
About	436
Report	438
YesNo	439
YesNoCancel	441
Input	444
ListSelect	446
File Select	449
Puntos principales del capítulo	457
13. Diálogo y Datos	459
Construcción de Diálogos para necesidades Específicas	461
Vista de un diálogo representativa	462
Creación de una ventana de diálogo	465
Adición de controles de diálogo	466
Control modal	468
Métodos de retrollamada (callback)	469
Diálogos y Datos	470
Consideraciones modelo – vista – controlador	471
Conexiones entre los componentes modelo – vista – controlador	473
Creación de una vista de diálogo para el modelo MODEM	475
Uso de modemdialog	478
Acceso directo al modelo	480
Puntos principales del capítulo	485
14. Objectwindows: Un Marco de Aplicación Independiente del Lenguaje	493
Un programa Objectwindows simple	494
Cómo agregar un título de ventana	497
Creación de un menú simple	498

Especificación de métodos de retrollamada	502
Diálogos, Menús y Recursos	507
Menús anidados	509
Métodos de retrollamada que manejan eventos Windows	511
Manejo de eventos de ratón con métodos de retrollamada	520
Los programas turbo pascal para Windows pueden utilizar rutinas de entrada / salida compatibles con DOS	523
Datos y Diálogos	530
Transferencia de datos hacia y desde casillas de diálogo	535
Diálogos de selección de archivo	539
Puntos principales del capítulo	546
IV. Apéndices	549
A. Rotación de Formas en pascal, Smalltalk y Actor	551
Rotación de formas en pascal orientado a objetos	551
Unidades turbo pascal	553
Instrucciones para compilar y enlazar en turbo pascal	555
Listado de archivo fuente en pascal	556
La clase Tscreen abstracta para Smalltalk	567
Las clases Tdemowindow y Tscreen para actor	571
B. Programa C ++ para el capítulo 14	573
Índice	583