

# INDICE DE CONTENIDOS

Prólogo

## PARTE I: PROGRAMACIÓN ORIENTADA A OBJETOS

### 1. Introducción: la orientación a objetos

Conceptos previos

Programación estructurada

Programación modular

Abstracción

Ocultamiento de implementación

Hacia un nuevo paradigma

Calidad del software

Orientación a objetos

Reducción de la brecha entre el mundo de los problemas  
y el mundo de los modelos

Aumento del nivel de complejidad de los sistemas

Reutilización y extensión del código

Uso de prototipos

Programación en ambientes de interfaz de usuario gráfica

Programación por eventos

Programación orientada a objetos

Significado

Lenguajes de programación orientada a objetos

Herramientas visuales y de desarrollo rápido de aplicaciones

### 2. Programación orientada a objetos: generalidades y encapsulamiento

Objetos y clases

Objetos

Clases

Encapsulamiento

La abstracción en programación orientada a objetos

El ocultamiento en programación orientada a objetos

Encapsulamiento

Tipos especiales de atributos

Atributos de clase

Atributos con valor constante

Atributos con restricciones

Métodos de clase

Terminología de objetos

Estados, eventos y transiciones	38
Operaciones	39
Estado observable de un objeto	39
Pasaje por valor y por referencia en programación orientada a objetos	39
Constructores	40
Destruyores	40
Sobrecarga	40
Implementación de objetos y clases en distintos lenguajes	42
Objetos y clases en Object Pascal	42
Objetos y clases en C++	45
Objetos y clases en Java	48
Ejercicio resuelto: implementación de números racionales	50
Enunciado	50
Solución en Object Pascal	50
Solución en C++	50
Solución en Java	50
Ejercicio resuelto: implementación de conjuntos	51
Enunciado	51
Solución en Object Pascal	51
Solución en C++	51
Solución en Java	51
<b>3. Programación orientada a objetos: reutilización y extensión del código</b>	<b>53</b>
Reutilización con composición	53
Composición	53
Agregación	54
Reutilización con herencia	54
Definición y ejemplos	54
Utilización de herencia	56
Herencia múltiple	58
Herencia con excepciones	58
Especialización con menos atributos	59
Constructores, destructores, herencia y composición	59
Atributos y métodos protegidos	60
Generalización, especialización e instanciación	61
Interfaces	61
Concepto de interfaz	61
Un ejemplo de interfaces: la clase adaptadora	62
Elección de implementación: ¿composición, herencia o interfaces?	64
Implementación de reutilización del código en distintos lenguajes	65

Composición, herencia e interfaces en Object Pascal	65
Composición y herencia simple y múltiple en C++	67
Composición, herencia e interfaces en Java	68
Ejercicio resuelto: implementación de matrices	70
Enunciado	70
Solución en Object Pascal	70
Solución en C++	70
Solución en Java	70
Ejercicio resuelto: uso de la implementación de matrices	71
Enunciado	71
Solución en Object Pascal	71
Solución en C++	71
Solución en Java	71
<b>4. Programación orientada a objetos: polimorfismo</b>	<b>73</b>
Polimorfismo	73
Polimorfismo y vinculación tardía	73
Redefinición	73
Sobrecarga y redefinición	75
Objetos polimorfos o conversión de tipo automática	75
Métodos virtuales	76
Constructores y destructores virtuales	78
Grados de polimorfismo	78
Clases y métodos abstractos	79
Clases abstractas	79
Métodos abstractos	80
Clases utilitarias y clases sin estado	80
Uso de información de tipos en tiempo de ejecución	81
Transformación de tipos	81
Moldeo automático y moldeo explícito	81
Datos estructurados con elementos de varios tipos	83
Información de tipos en tiempo de ejecución	84
Implementación del polimorfismo en distintos lenguajes	85
Polimorfismo y otros aspectos en Object Pascal	85
Polimorfismo y otros aspectos en C++	86
Polimorfismo y otros aspectos en Java	87
Ejercicio resuelto: propiedades de una inmobiliaria	89
Enunciado	89
Solución en Object Pascal	89
Solución en C++	89

Solución en Java	89
Ejercicio resuelto: uso de la implementación de la inmobiliaria	89
Enunciado	89
Solución en Object Pascal	89
Solución en C++	89
Solución en Java	89
<b>5. Documentando la programación</b>	<b>91</b>
Documentación en general	91
UML para documentar la programación	92
UML	92
Diagramas de clases	94
Diagramas de secuencia	98
Diagramas de estados	101
Diagramas de actividades	102
Diagramas y programas	104
Documentación interna	105
Documentación interna de la programación	105
El caso de javadoc	105
Estándares de nomenclatura	107
Ejercicio resuelto: diagramas de actividades, clases y secuencia	109
Enunciado	109
Solución	109
Ejercicio resuelto: un diagrama de estados	112
Enunciado	112
Solución	112
<b>6. Temas misceláneos de programación orientada a objetos</b>	<b>113</b>
Colecciones en los lenguajes orientados a objetos	113
Colecciones y niveles de abstracción	113
Tipos genéricos y clases parametrizadas	114
Iteradores	115
Colecciones e iteradores en Java	116
Colecciones e iteradores en C++	118
Colecciones en Object Pascal (Delphi y Kylix)	119
Programación guiada por eventos	120
La programación secuencial y sus limitaciones	120
Eventos y programación	122
Mensajes y eventos	124
Metodologías de ruteo de mensajes	125
Manejo de los eventos	126

Generalización del concepto de eventos en programación orientada a objetos	126
Componentes	127
Paquetes	127
Un soporte para la administración de clases	127
Organización de las clases en Object Pascal	128
Organización de las clases en C++	129
Organización de las clases en Java	130
Colaboraciones	131
Patrones de diseño	131
Patrones de diseño y de arquitectura	131
Caso I: Singleton	133
Caso II: recorridos de colecciones	134
Caso III: el patrón observador	136
Lenguajes orientados a objetos	139
Simula	139
Smalltalk	139
C++	140
Object Pascal / Delphi / Kylix	140
Java	141
Python	142
C#	142
Visual Basic 6	143
Javascript	143
Un cuadro comparativo de lenguajes	143
Ejercicio resuelto: un paquete de figuras geométricas	145
Enunciado	145
Solución en Object Pascal	145
Solución en C++	145
Solución en Java	145
Ejercicio propuesto: patrón observador	145
Enunciado	145
<b>7. Análisis global del paradigma de orientación a objetos</b>	<b>147</b>
Qué aporta de nuevo la programación orientada a objetos	147
Una serie de requisitos mínimos para ser "orientado a objetos"	149
Conceptos de desarrollo de software antes y después de la orientación a objetos	150
Definición de sistema	150
Cohesión y acoplamiento	151

Ciclo de vida tradicional del software	152
Desarrollo incremental	153
Desarrollo con prototipos completos	154
El Proceso Unificado de Desarrollo de Software	154
Extreme programming	155
Desarrollo orientado a objetos para proyectos pequeños	159
Algunas notas sobre mantenimiento	161
Una introducción a las pruebas en programación orientada a objetos	161
Las pruebas en el desarrollo de software	161
Las pruebas en las metodologías de desarrollo incremental	161
Las pruebas en extreme programming	162
Pruebas y programación orientada a objetos	164
El futuro de la orientación a objetos	165
Recomendaciones generales para diseño y programación orientada a objetos	166

## **PARTE II: ASPECTOS AVANZADOS**

<b>8. Tratamiento de problemas en tiempo de ejecución y excepciones</b>	<b>171</b>
Dependencia de estados	171
Introducción	171
Enfoques conservadores	171
Enfoques optimistas	172
Excepciones	172
Errores y excepciones	172
Distintas formas de tratar excepciones	174
Transacciones	175
Lanzamiento de excepciones	176
Captura de excepciones: manejadores	177
Atributos y métodos en excepciones creadas por el programador	178
Excepciones derivadas	179
Excepciones en Object Pascal	180
Excepciones en C++	181
Excepciones en Java	182
El patrón de moldes de métodos para invariantes	183
Algunas desventajas del modelo actual de excepciones	184
Ejercicio resuelto: implementación de fechas con excepciones	185
Enunciado	185
Solución en Object Pascal	185
Solución en C++	185

Solución en Java	185
Ejercicio propuesto: jerarquía de excepciones	185
Enunciado	185
<b>9. Distintos modelos de datos</b>	<b>187</b>
¿Objetos estáticos o referencias?	187
Inicialización, asignación y liberación de memoria	188
Copia y clonación	189
Asignación y copia en el modelo estático	189
La clonación en el modelo dinámico	190
Igualdad y desigualdad	192
Ejercicio resuelto: implementación de copia/clonación e igualdad	192
Enunciado	192
Solución en Object Pascal	192
Solución en C++	192
Solución en Java	192
Ejercicio propuesto: copia/clonación e igualdad	193
Enunciado	193
<b>10. Persistencia de objetos</b>	<b>195</b>
Persistencia y objetos persistentes	195
Conceptos básicos	195
Separación de la capa de acceso a datos	196
Normas de la persistencia	196
Persistencia y bases de datos relacionales	197
Bases de datos orientadas a objetos	198
Evolución del modelo de clases	200
Soluciones para el almacenamiento de objetos	200
Serialización y XML	200
Serialización en Java	202
Del modelo de objetos al modelo relacional	203
Consideraciones adicionales	203
Problemas con objetos complejos	203
Los lenguajes orientados a objetos y la persistencia	204
Ejercicio resuelto: serialización XML	205
Enunciado	205
Solución en Object Pascal	206
Ejercicio resuelto: serialización a una cadena de bytes	206
Enunciado	206
Solución en Java	206

<b>11. Concurrencia y programación orientada a objetos</b>	<b>207</b>
Conceptos generales	207
Haciendo varias cosas a la vez	207
Concurrencia	208
Procesos e hilos	210
Concurrencia y programación orientada a objetos	211
Representación de concurrencia en UML	212
Escenarios concurrentes	213
Comunicación entre objetos activos	213
Objetos activos comunicándose con objetos pasivos	214
Prioridades	215
Problemas y soluciones	215
Exclusión mutua y sincronización de métodos y bloques	215
Inconvenientes de la sincronización de métodos	218
Objetos inmutables	218
Llaves de lectura y escritura	220
Confinamiento	220
Llaves de exclusión mutua (mutex)	221
Semáforos	222
Actualizaciones optimistas	222
Otras soluciones para evitar la sincronización de métodos	223
Bloqueos transitorios y bloqueos patológicos	223
Interbloqueo	224
Dependencia de estados, excepciones y constructores	225
Problemas con los compiladores y las optimizaciones	226
Cuándo usar concurrencia y cuándo no	226
Concurrencia y lenguajes de programación	228
Concurrencia en Java	229
Concurrencia en Object Pascal (Delphi y Kylix)	233
Concurrencia en C++	236
Temas afines	237
Programación en contextos de multiprocesamiento	237
Computación distribuida	238
Ejercicio resuelto: exclusión en el acceso a una cuenta bancaria	240
Enunciado	240
Solución en Java I (con sincronización total)	240
Solución en Java II (con llaves de lectura y escritura)	240
Solución en Object Pascal con llaves de lectura y escritura	240
Ejercicio resuelto: múltiples hilos en la multiplicación de grandes matrices	241

Enunciado	241
Solución en Object Pascal	241
Solución en Java	241
<b>PARTE III: ALGUNAS APLICACIONES</b>	<b>243</b>
<b>12. Programación en ambientes de interfaz de usuario gráfica</b>	<b>245</b>
programación en interfaces gráficas y objetos	245
Implementación de la programación por eventos en los lenguajes orientados a objetos	245
WIMP y programación por eventos	245
Eventos en Delphi y Kylix	246
Eventos en Java Swing	248
¿Eventos en C++?	249
Separación de capas de interfaz de usuario y modelo	250
Interfaces de usuario y HCI	250
Independencia de diálogo	251
MVC	252
Consecuencias de la independencia de diálogo	254
<b>13. Programación Web</b>	<b>257</b>
Conceptos básicos	257
Redes e Internet	257
Aplicaciones y World Wide Web	257
HTML y contenido dinámico del lado del cliente	258
Generación de páginas en el servidor	259
Sitios Web, Aplicaciones Web y Servicios Web	259
La programación de aplicaciones y servicios Web	261
Aplicando la orientación a objetos	261
Otras consideraciones	261
Separación en capas	262
<b>APÉNDICES</b>	<b>263</b>
<b>Apéndice A: soluciones de ejercicios en Object Pascal</b>	<b>265</b>
Implementación de números racionales	265
Implementación de conjuntos	266
Implementación de matrices	269
Uso de la implementación de matrices	277
Propiedades de una inmobiliaria	278
Uso de la implementación de la inmobiliaria	283

Paquete de figuras geométricas	283
Implementación de fechas con excepciones	288
Implementación de copia/clonación e igualdad	292
Serialización XML	294
Exclusión en el acceso a una cuenta bancaria: llaves de lectura y escritura	297
Múltiples hilos en la multiplicación de grandes matrices	299
<b>Apéndice B: soluciones de ejercicios en C++</b>	<b>301</b>
Implementación de números racionales	301
Implementación de conjuntos	302
Implementación de matrices	304
Uso de la implementación de matrices	311
Propiedades de una inmobiliaria	311
Uso de la implementación de la inmobiliaria	316
Paquete de figuras geométricas	316
Implementación de fechas con excepciones	319
Implementación de copia/clonación e igualdad	323
<b>Apéndice C: soluciones de ejercicios en Java</b>	<b>327</b>
Implementación de números racionales	327
Implementación de conjuntos	328
Implementación de matrices	329
Uso de la implementación de matrices	336
Propiedades de una inmobiliaria	336
Uso de la implementación de la inmobiliaria	341
Paquete de figuras geométricas	342
Implementación de fechas con excepciones	344
Implementación de copia/clonación e igualdad	348
Serialización a una cadena de bytes	350
Exclusión en el acceso a una cuenta bancaria I: sincronización total	351
Exclusión en el acceso a una cuenta bancaria II: llaves de lectura y escritura	352
Múltiples hilos en la multiplicación de grandes matrices	356
<b>Bibliografía</b>	<b>357</b>
Bibliografía usada	357
Libros recomendados	358
Otros libros del autor	358