

Contenido

1	Introducción al lenguaje de programación Java	1	2.2.3	El método toString	41
1.1	Introducción	2	2.2.4	El método equals	42
1.2	Comencemos a programar	2	2.2.5	Definir y "crear" objetos	43
1.2.1	El Entorno Integrado de Desarrollo (IDE)	3	2.2.6	El constructor	44
1.3	Estructuras de control	3	2.2.7	Un pequeño repaso de lo visto hasta aquí	45
1.3.1	Entrada y salida de datos por consola	3	2.2.8	Convenciones de nomenclatura	46
1.3.2	Definición de variables	4	2.2.9	Sobrecarga	47
1.3.3	Comentarios en el código	5	2.2.10	Encapsulamiento	50
1.3.4	Estructuras de decisión	6	2.2.11	Visibilidad de métodos y atributos	52
1.3.5	Estructuras iterativas	10	2.2.12	Packages (paquetes)	54
1.4	Otros elementos del lenguaje	12	2.2.13	La estructura de los paquetes y la variable CLASSPATH	54
1.4.1	Tipos de datos	12	2.2.14	Las APIs ("Application Programming Interface")	55
1.4.2	Algunas similitudes y diferencias con C y C++	12	2.2.15	Representación gráfica UML	55
1.4.3	Definición de constantes	13	2.2.16	Importar clases de otros paquetes	56
1.4.4	Arrays	14	2.3	Herencia y polimorfismo	57
1.4.5	Matrices	17	2.3.1	Polimorfismo	59
1.4.6	Literales de cadenas de caracteres	18	2.3.2	Constructores de subclases	61
1.4.7	Caracteres especiales	20	2.3.3	La referencia super	62
1.4.8	Argumentos en línea de comandos	21	2.3.4	La referencia this	64
1.5	Tratamiento de cadenas de caracteres	22	2.3.5	Clases abstractas	65
1.5.1	Acceso a los caracteres de un String	22	2.3.6	Constructores de clases abstractas	69
1.5.2	Mayúsculas y minúsculas	23	2.3.7	Instancias	72
1.5.3	Ocurrencias de caracteres	23	2.3.8	Variables de instancia	73
1.5.4	Subcadenas	24	2.3.9	Variables de la clase	75
1.5.5	Prefijos y sufijos	24	2.3.10	El Garbage Collector (recolector de residuos)	75
1.5.6	Posición de un substring dentro de la cadena	25	2.3.11	El método finalize	76
1.5.7	Concatenar cadenas	25	2.3.12	Constantes	77
1.5.8	La clase StringBuffer	26	2.3.13	Métodos de la clase	77
1.5.9	Conversión entre números y cadenas	27	2.3.14	Clases utilitarias	79
1.5.10	Representación numérica en diferentes bases	28	2.3.15	Referencias estáticas	79
1.5.11	La clase StringTokenizer	29	2.3.16	Colecciones (primera parte)	81
1.5.12	Usar expresiones regulares para particionar una cadena	30	2.3.17	Clases genéricas	86
1.5.13	Comparación de cadenas	31	2.3.18	Implementación de una pila (estructura de datos)	88
1.6	Operadores	33	2.3.19	Implementación de una cola (estructura de datos)	90
1.6.1	Operadores aritméticos	33	2.4	Interfaces	91
1.6.2	Operadores lógicos	33	2.4.1	Desacoplamiento de clases	93
1.6.3	Operadores relacionales	34	2.4.2	El patrón de diseño factory method	95
1.6.4	Operadores lógicos de bit	34	2.4.3	Abstracción a través de interfaces	95
1.6.5	Operadores de desplazamiento de bit	34	2.4.4	La interface comparable	95
1.7	La máquina virtual y el JDK	34	2.4.5	Desacoplar aún más	99
1.7.1	El JDK (Java Development Kit)	35	2.5	Colecciones	103
1.7.2	Versiones y evolución del lenguaje Java	35	2.5.1	Cambio de implementación	105
1.8	Resumen	35	2.6	Excepciones	106
1.9	Contenido de la página Web de apoyo	36	2.6.1	Excepciones declarativas y no declarativas	109
1.9.1	Mapa conceptual	36	2.6.2	El bloque try-catch-finally	111
1.9.2	Autoevaluación	36	2.7	Resumen	112
1.9.3	Videotutoriales	36	2.8	Contenido de la página Web de apoyo	112
1.9.4	Presentaciones*	36	2.8.1	Mapa conceptual	112
2	Programación orientada a objetos	37	2.8.2	Autoevaluación	112
2.1	Introducción	38	2.8.3	Videotutoriales	112
2.2	Clases y objetos	38	2.8.4	Presentaciones*	112
2.2.1	Los métodos	39	3	Acceso a bases de datos (JDBC)	113
2.2.2	Herencia y sobrescritura de métodos	41	3.1	Introducción	114

3.2	Conceptos básicos sobre bases de datos relacionales	114	5.2.1	Distribución de componentes (layouts)	163
3.2.1	Relaciones foráneas y consistencia de datos	115	5.2.2	AWT y Swing	163
3.2.2	Diagrama Entidad-Relación (DER)	115	5.3	Comenzando a desarrollar GUI	164
3.2.3	SQL – Structured Query Language	116	5.3.1	Distribuciones relativas	164
3.2.4	Ejecutar sentencias query	116	5.3.2	FlowLayout	164
3.2.5	Unir tablas (join)	117	5.3.3	BorderLayout	167
3.2.6	Ejecutar sentencias UPDATE	117	5.3.4	GridLayout	168
3.3	Conectar programas Java con bases de datos	118	5.3.5	Combinación de layouts	170
3.3.1	Invocar un query con un join	123	5.4	Capturar eventos	176
3.3.2	Updates	123	5.4.1	Tipos de eventos	180
3.3.3	Ejecutar un INSERT	123	5.4.2	Eventos de acción	181
3.3.4	Ejecutar un DELETE	125	5.4.3	Eventos de teclado	184
3.3.5	Ejecutar un UPDATE	125	5.5	Swing	186
3.3.6	El patrón de diseño “Singleton” (Singleton Pattern)	125	5.5.1	Cambiar el LookandFeel	190
3.3.7	Singleton Pattern para obtener la conexión	126	5.6	Model View Controller (MVC)	192
3.3.8	El shutdown hook	128	5.6.1	Ejemplo de uso: ListModel	192
3.3.9	Inner classes (clases internas)	128	5.6.2	Ejemplo de uso: TableModel	195
3.3.10	Manejo de transacciones	129	5.7	Resumen	197
3.4	Uso avanzado de JDBC	131	5.8	Contenido de la página Web de apoyo	198
3.4.1	Acceso a la metadata del resultSet (ResultSetMetaData)	131	5.8.1	Mapa conceptual	198
3.4.2	Definir el “Query Fetch Size” para conjuntos de resultados grandes	133	5.8.2	Autoevaluación	198
3.4.3	Ejecutar batch updates (procesamiento por lotes)	133	5.8.3	Videotutorial	198
3.5	Resumen	134	5.8.4	Presentaciones*	198
3.6	Contenido de la página Web de apoyo		6	Multithreading (Hilos)	199
3.6.1	Mapa conceptual	135	6.1	Introducción	200
3.6.2	Autoevaluación	135	6.2	Implementar threads en Java	201
3.6.3	Videotutorial	135	6.2.1	La interface Runnable	202
3.6.4	Presentaciones*	135	6.2.2	Esperar a que finalice un thread	203
4	Diseño de aplicaciones Java (Parte I)	137	6.2.3	Threads y la interfaz gráfica	204
4.1	Introducción	138	6.2.4	Sistemas operativos multitarea	206
4.2	Atributos de una aplicación	138	6.2.5	Ciclo de vida de un thread	207
4.2.1	Casos de uso	138	6.2.6	Prioridad de ejecución	209
4.3	Desarrollo de aplicaciones en capas	139	6.3	Sincronización de threads	209
4.3.1	Aplicación de estudio	140	6.3.1	Monitores y sección crítica	210
4.3.2	Análisis de los objetos de acceso a datos (DAO y DTO)	141	6.3.2	Ejemplo del Productor/Consumidor	210
4.3.3	Análisis del façade	145	6.4	Resumen	214
4.3.4	Diagrama de secuencias de UML	147	6.5	Contenido de la página Web de apoyo	215
4.4	Portabilidad entre diferentes bases de datos	148	6.5.1	Mapa conceptual	215
4.4.1	DAOs abstractos e implementaciones específicas para las diferentes bases de datos	150	6.5.2	Autoevaluación	215
4.4.2	Implementación de un factory method	153	6.5.3	Presentaciones*	215
4.4.3	Combinar el factory method con el singleton pattern	154	7	Networking	217
4.4.4	Mejorar el diseño de los DAOs abstractos	156	7.1	Introducción	218
4.5	Diseño por contratos	158	7.2	Conceptos básicos de networking	218
4.5.1	Coordinación de trabajo en equipo	158	7.2.1	TCP - “Transmission Control Protocol”	218
4.6	Resumen	160	7.2.2	UDP - “User Datagram Protocol”	218
4.7	Contenido de la página Web de apoyo	160	7.2.3	Puertos	219
4.7.1	Mapa conceptual	160	7.2.4	Dirección IP	219
4.7.2	Autoevaluación	160	7.2.5	Aplicaciones cliente/servidor	219
4.7.3	Presentaciones*	160	7.3	TCP en Java	219
5	Interfaz gráfica (GUI)	161	7.3.1	El socket	219
5.1	Introducción	162	7.3.2	Un simple cliente/servidor en Java	219
5.2	Componentes y contenedores	162	7.3.3	Serialización de objetos	222
			7.3.4	Implementación de un servidor multithread	223
			7.3.5	Enviar y recibir bytes	225
			7.3.6	Enviar y recibir valores de tipos de datos primitivos	228
			7.4	UDP en Java	228

7.5	Remote Method Invocation (RMI)	230	9.4.3	Presentaciones*	284
7.5.1	Componentes de una aplicación RMI	231	10	Parametrización mediante archivos XML	285
7.5.2	Ejemplo de una aplicación que utiliza RMI	231	10.1	Introducción	286
7.5.3	Compilar y ejecutar la aplicación RMI	233	10.2	XML - "Extensible Markup Language"	286
7.5.4	RMI y serialización de objetos	234	10.3	Estructurar y definir parámetros en un archivo XML	287
7.6	Resumen	234	10.3.1	Definición de la estructura de parámetros	287
7.7	Contenido de la página Web de apoyo	234	10.3.2	Leer y parsear el contenido de un archivo XML	289
7.7.1	Mapa conceptual	234	10.3.3	Acceder a la información contenida en el archivo XML	292
7.7.2	Autoevaluación	234	10.4	Resumen	300
7.7.3	Videotutorial	234	10.5	Contenido de la página Web de apoyo	300
7.7.4	Presentaciones*	234	10.5.1	Mapa conceptual	300
8	Diseño de aplicaciones Java (Parte II)	235	10.5.2	Autoevaluación	300
8.1	Introducción	236	10.5.3	Presentaciones*	300
8.2	Repaso de la aplicación de estudio	236	11	Introspección de clases y objetos	301
8.3	Capas lógicas vs. capas físicas	238	11.1	Introducción	302
8.3.1	Desventajas de un modelo basado en dos capas físicas	238	11.2	Comenzando a introspectar	303
8.3.2	Modelo de tres capas físicas	239	11.2.1	Identificar métodos y constructores	303
8.4	Desarrollo de la aplicación en tres capas físicas	240	11.2.2	Acceso al prototipo de un método	305
8.4.1	Desarrollo del servidor	240	11.3	Annotations	307
8.4.2	Desarrollo de un cliente de prueba	244	11.4	Resumen	310
8.4.3	El service locator (o ubicador de servicios)	246	11.5	Contenido de la página Web de apoyo	310
8.4.4	Integración con la capa de presentación	251	11.5.1	Mapa conceptual	310
8.5	Implementación del servidor con tecnología RMI	253	11.5.2	Autoevaluación	310
8.5.1	El servidor RMI	253	11.5.3	Presentaciones*	310
8.5.2	El ServiceLocator y los objetos distribuidos	255	12	Generalizaciones y desarrollo de frameworks	311
8.5.3	Desarrollo de un cliente de prueba	256	12.1	Introducción	312
8.5.4	Integración con la capa de presentación	257	12.2	¿Qué es un framework?	312
8.5.5	El bussiness delegate	259	12.2.1	¿Frameworks propios o frameworks de terceros?	313
8.6	Concurrencia y acceso a la base de datos	259	12.2.2	Reinventar la rueda	313
8.6.1	El pool de conexiones	260	12.3	Un framework para acceder a archivos XML	314
8.6.2	Implementación de un pool de conexiones	260	12.3.1	Diseño de la API del framework	315
8.6.3	Integración con los servidores TCP y RMI	265	12.3.2	Análisis del elemento a generalizar	317
8.7	Resumen	266	12.3.3	Parsear el archivo XML y cargar la estructura de datos	318
8.8	Contenido de la página Web de apoyo	266	12.4	Un framework para acceder a bases de datos	324
8.8.1	Mapa conceptual	266	12.4.1	Identificación de la tarea repetitiva	325
8.8.2	Autoevaluación	266	12.4.2	Diseño de la API del framework	326
8.8.3	Presentaciones*	266	12.4.3	Java Beans	327
9	Estructuras de datos dinámicas	267	12.4.4	Transacciones	338
9.1	Introducción	268	12.4.5	Mappeo de tablas usando annotations	342
9.2	Estructuras dinámicas	268	12.5	El bean factory	345
9.2.1	El nodo	268	12.6	Integración	347
9.2.2	Lista enlazada (LinkedList)	269	12.6.1	Los objetos de acceso a datos	347
9.2.3	Pila	273	12.6.2	El façade	348
9.2.4	Cola	274	12.6.3	El archivo de configuración	349
9.2.5	Implementación de una cola sobre una lista circular	275	12.6.4	El cliente	350
9.2.6	Clases LinkedList, Stack y Queue	277	12.8	Resumen	351
9.2.7	Tablas de dispersión (Hashtable)	278	12.9	Contenido de la página Web de apoyo	351
9.2.8	Estructuras de datos combinadas	280	12.9.1	Mapa conceptual	351
9.2.9	Árboles	282	12.9.2	Autoevaluación	351
9.2.10	Árbol binario de búsqueda	283	12.9.3	Presentaciones*	351
9.2.11	La clase TreeSet	284	13	Entrada/Salida	353
9.3	Resumen	284	13.1	Introducción	354
9.4	Contenido de la página Web de apoyo	284	13.2	I/O streams (flujos de entrada y salida)	354
9.4.1	Mapa conceptual	284			
9.4.2	Autoevaluación	284			

13.2.1	Entrada y salida estándar	354
13.2.2	Redireccionar la entrada y salidas estándar	355
13.2.3	Cerrar correctamente los streams	356
13.2.4	Streams de bytes (InputStream y OutputStream)	357
13.2.5	Streams de caracteres (readers y writers)	358
13.2.6	Streams bufferizados	359
13.2.7	Streams de datos (DataInputStream y DataOutputStream)	360
13.2.8	Streams de objetos (ObjectInputStream y ObjectOutputStream)	361
13.3	Resumen	363
13.4	Contenido de la página Web de apoyo	364
13.4.1	Mapa conceptual	364
13.4.2	Autoevaluación	364
13.4.3	Presentaciones*	364
14	Consideraciones finales	365
14.1	Introducción	366
14.2	Consideraciones sobre multithreading y concurrencia	366
14.2.1	Clases con o sin métodos sincronizados	366
14.2.2	El singleton pattern en contextos multithreaded	366
14.3	Consideraciones sobre clases "legacy"	367
14.3.1	La clase StringTokenizer y el método split	367
14.4	Resumen	368
14.5	Contenido de la página Web de apoyo	368
14.5.1	Mapa conceptual	368
14.5.2	Autoevaluación	368
14.5.3	Presentaciones*	368
15	Object Relational Mapping (ORM) y persistencia de datos	369
15.1	Introducción	370
15.2	Hibernate framework	371
15.2.1	El modelo de datos relacional	371
15.2.2	ORM (Object Relational Mapping)	372
15.2.3	Configuración de Hibernate	372
15.2.4	Mappeo de tablas	373
15.2.5	La sesión de Hibernate	375
15.3	Asociaciones y relaciones	375
15.3.1	Asociación many-to-one	375
15.3.2	Asociación one-to-many	377
15.3.3	P6Spy	378
15.3.4	Lazy loading vs. eager loading	379
15.4	Recuperar colecciones de objetos	381
15.4.1	Criterios de búsqueda vs. HQL	381
15.4.2	Named queries	383
15.4.3	Ejecutar SQL nativo	384
15.4.4	Queries parametrizados	384
15.5	Insertar, modificar y eliminar filas	384
15.5.1	Transacciones	384
15.5.2	Insertar una fila	385
15.5.3	Estrategia de generación de claves primarias	385
15.5.4	Modificar una fila	386
15.5.5	Múltiples updates y deletes	386
15.6	Casos avanzados	387
15.6.1	Análisis y presentación del modelo de datos	387
15.6.2	Asociaciones many-to-many	388
15.6.3	Claves primarias compuestas (Composite Id)	390
15.7	Diseño de aplicaciones	392
15.7.1	Factorías de objetos	393
15.8	Resumen	399
15.9	Contenido de la página Web de apoyo	400
15.9.1	Mapa conceptual	400
15.9.2	Autoevaluación	400
15.9.3	Presentaciones*	400
16	Inversión del control por inyección de dependencias	401
16.1	Introducción	402
16.2	Spring framework	402
16.2.1	Desacoplar el procesamiento	404
16.2.2	Conclusión y repaso	408
16.3	Spring y JDBC	409
16.4	Integración Spring + Hibernate	412
16.5	Resumen	415
16.6	Contenido de la página Web de apoyo	416
16.6.1	Mapa conceptual	416
16.6.2	Autoevaluación	416
16.6.3	Presentaciones*	416
17	Actualización a Java7	417
17.1	Introducción	418
17.2	Novedades en Java 7	418
17.2.1	Literales binarios	418
17.2.2	Literales numéricos separados por "_" (guion bajo)	418
17.2.3	Uso de cadenas en la sentencia switch	419
17.2.4	Inferencia de tipos genéricos	419
17.2.5	Sentencia try con recurso incluido	420
17.2.6	Atrapar múltiples excepciones dentro de un mismo bloque catch	420
17.2.7	Nuevos métodos en la clase File	420
17.3	Contenido de la página Web de apoyo	421
17.3.1	Mapa conceptual	421
17.3.2	Auto evaluación	421
17.3.3	Presentaciones*	421
Apéndice A Nociones básicas de programación		423
A.1	Introducción	424
A.2	Conceptos iniciales	424
A.2.1	El lenguaje de programación	424
A.2.2	El compilador	424
A.2.3	Los intérpretes	424
A.2.4	Las máquinas virtuales	425
A.2.5	Java y su máquina virtual	425
A.3	Recursos de programación	425
A.3.1	Las variables	425
A.3.2	Tipos de datos	425
A.3.3	Operadores aritméticos	426
A.3.4	Estructuras de decisión	427
A.3.5	Estructuras de repetición	427
Apéndice B Applets		429
B.1	Introducción	430
B.2	Comenzando con Applets	430
B.2.1	El ciclo de vida de un applet	432
B.2.2	El contexto del applet	433
B.2.3	Pasarle parámetros a un applet	433
Bibliografía		435

Información del contenido de la página Web

El material marcado con asterisco (*) solo está disponible para docentes.

Capítulo 1

Introducción al lenguaje de programación Java

- Mapa conceptual
- Autoevaluación
- Videotutoriales:
 - Instalar Java y *Eclipse*.
 - Crear y ejecutar nuestro primer programa en *Eclipse*.
 - Pasar argumentos en línea de comandos en *Eclipse*.
 - Compilar y ejecutar un programa Java desde la línea de comandos (sin utilizar *Eclipse*).
- Presentaciones*

Capítulo 2

Programación orientada a objetos

- Mapa conceptual
- Autoevaluación
- Videotutoriales:
 - Utilizar la herramienta "javadoc" para documentar nuestro código fuente.
 - Empaquetar clases utilizando la herramienta "jar".
- Presentaciones*

Capítulo 3

Acceso a bases de datos (JDBC)

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Usar *Eclipse* como cliente SQL.
- Presentaciones*

Capítulo 4

Diseño de aplicaciones Java (Parte I)

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 5

Interfaz gráfica (GUI)

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Utilizar herramientas visuales para diseño y desarrollo de interfaz gráfica.
- Presentaciones*

Capítulo 6

Multithreading (Hilos)

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 7

Networking

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Compilar y ejecutar una aplicación RMI.
- Presentaciones*

Capítulo 8

Diseño de aplicaciones Java (Parte II)

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 9

Estructuras de datos dinámicas

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 10

Parametrización mediante archivos XML

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 11

Introspección de clases y objetos

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 12

Generalizaciones y desarrollo de frameworks

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 13

Entrada/Salida

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 14

Consideraciones adicionales

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 15

Object Relational Mapping (ORM) y persistencia de datos

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 16

Inversión del control por inyección de dependencias

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 17

Actualización a Java 7

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Código fuente de cada capítulo

Hipervínculos de interés

Fe de erratas

Guía para el docente de las competencias específicas que se desarrollan con este libro