



Lenguaje de programación con JAVA

Autor: Eric Gustavo Coronel Castillo

© Derechos de autor registrados:

Empresa Editora Macro EIRL

© Derechos de edición, arte gráfico y diagramación reservados:

Empresa Editora Macro EIRL

Jefe de edición:

Cynthia Arestegui Baca

Coordinación de edición:

Magaly Ramon Quiroz

Diseño de portada:

Alessandra Bonilla Zapata

Corrección de estilo:

Hassel Ortiz Huamán

Diagramación:

Verónica Calderón Cáceres

Edición a cargo de:

© Empresa Editora Macro EIRL

Av. Paseo de la República N.° 5613, Miraflores, Lima, Perú

☎ Teléfono: (511) 748 0560

✉ E-mail: proyectoeditorial@editorialmacro.com

🌐 Página web: www.editorialmacro.com

Segunda edición: mayo de 2015

Tiraje: 1000 ejemplares

Impresión

Talleres gráficos de la Empresa Editora Macro EIRL

Jr. San Agustín N.° 612-624, Surquillo, Lima, Perú

ISBN N.° 978-612-304-288-2

Hecho el depósito legal en la Biblioteca Nacional del Perú N.° 2015-05711

Prohibida la reproducción parcial o total, por cualquier medio o método, de este libro sin previa autorización de la Empresa Editora Macro EIRL.

Índice

Capítulo 1: Fundamentos de la Programación

1.1 Historia de Java	25
1.1.1 Orígenes.....	25
1.1.2 Aspectos cronológicos	26
a. Java 1.....	26
b. Java 2.....	26
1.2 Características de Java.....	27
1.2.1 Plataforma de ejecución	27
1.2.2 Características del Lenguaje.....	28
a. Simple.....	28
b. Orientado a objetos	28
c. Distribuido	29
d. Robusto	29
e. Arquitectura neutral.....	30
f. Seguro	31
g. Portable.....	33
h. Interpretado.....	34
i. Multihilo.....	38
j. Dinámico	38
1.2.3 Ejecución de programas Java	39
1.3 Productos & Tecnología relacionada con Java.....	40
1.3.1 J2SE (Java2 Standard Edition)	40
a. JRE – Java2 <i>Runtime Enviroment</i>	41
b. JDK ¹ – Java2 <i>Development Kip</i>	41
1.3.2 J2EE (Java2 Enterprise Edition)	41
1.3.3 J2ME (Java2 Micro Edition).....	42

1.3.4 Java Card Technology	42
1.3.5 Java Web Services	43
1.3.6 Java Business Integration (JBI)	43
1.3.7 Java XML	43

Capítulo 2: Software e Instalación

2.1 Software	45
2.2 Instalación del JDK.....	46
2.3 Herramientas del JDK	51
2.3.1 Herramientas básicas del JDK	52
2.3.2 Herramientas de seguridad	53
2.3.3 Herramientas de internacionalización	53
2.3.4 Herramientas RMI (Remote Method Invocation)	54
2.4 Probando Java	54
2.4.1 Carpeta de trabajo	54
2.4.2 Verificando la variable de entorno PATH.....	54
2.5 Ejecutando el primer programa	56
2.6 Configurando la variable PATH	57

Capítulo 3: Herramientas de desarrollo

3.1 Historia de Java	61
3.2 EditPlus.....	62
3.3 JDeveloper.....	63
3.4 NetBeans	64
3.4.1 Inicio de un nuevo proyecto	66
3.5 JCreator Pro	68
3.5.1 Inicio de un nuevo proyecto	70
3.5.2 Agregar una nueva clase	71
3.6 Eclipse.....	72
3.6.1 Inicio de un nuevo proyecto	75
3.6.2 Agregar una nueva clase	75

3.7 JBuilder.....	77
3.7.1 Inicio de un nuevo proyecto	79
3.7.2 Agregar una nueva clase	79

Capítulo 4: Fundamentos de la Programación Orientada a Objetos

4.1 Introducción	83
4.2 Conceptos básicos sobre POO	84
4.2.1 Definición de clase	84
4.2.2 Definición de objeto	86
4.2.3 Mensaje	88
4.3 Encapsulación.....	89
4.4 Herencia	92
4.4.1 This.....	94
4.4.2 Super.....	94
4.5 Polimorfismo	95
4.6 Planteamiento de la implementación	97
4.7 Diseño de la estructura y el comportamiento de un objeto.....	98
4.7.1 Clase.....	98
4.7.2 Diferencia entre operación y método	99

Capítulo 5: Fundamentos de lenguaje

5.1 Esquema de un programa Java	101
5.1.1 Instanciación de una clase	102
5.1.2 Argumentos del método Main.....	104
5.2 Elementos básicos del lenguaje.....	107
5.2.1 Identificadores	107
5.2.2 Palabras clave e identificadores reservados	107
5.2.3 Comentarios	109
5.2.4 Punto y coma; bloques y espacios en blanco	109
5.2.5 Tipos de datos primitivos.....	111
a. Tipo boolean	112

b. Tipo char	113
c. Tipos byte, short, int y long	113
d. Tipos flot y double.....	115
5.2.6 Variables y operadores	116
a. Variables.....	116
b. Operadores aritméticos	116
c. Expresión lógica.....	119
5.2.7 Transformación de tipos: casting	126
5.3 Impresión en la consola.....	127
5.3.1 Método print()	127
5.3.2 Método println()	128
5.3.3 Método printf()	128

Capítulo 6: Instrucciones de control

6.1 Introducción	131
6.2 Instrucciones selectivas.....	131
6.2.1 Selectiva simple: if	131
6.2.2 Selectiva doble: if - else	133
6.2.3 Anidamiento de instrucciones if	134
6.2.4 Selectiva múltiple: switch	137
6.3 Instrucciones repetitivas	140
6.3.1 Bucle while.....	140
6.3.2 Bucle do – while.....	144
6.3.3 Bucle for.....	145
6.3.4 Bucle for con arreglos	149
6.3.5 Bucle for con colecciones.....	151
6.4 Instrucciones: break y continue	152

Capítulo 7: Arreglos

7.1 ¿Qué es un arreglo?	153
7.2 Arreglos unidimensionales – vectores.....	154

7.2.1 Declaración de arreglos	154
7.2.2 Creación de arreglos	154
7.2.3 Acceso a elementos de un arreglo.....	155
7.2.4 Inicialización de un arreglo	155
7.2.5 Averiguar el tamaño de un arreglo	156
7.3 Aplicación de arreglos unidimensionales	156
7.3.1 Generación de números aleatorios.....	156
7.4 Arreglos bidimensionales – matrices.....	162
7.4.1 Declaración de arreglos bidimensionales	162
7.4.2 Creación de arreglos	163
7.4.3 Acceso a los elementos de una matriz.....	163
7.4.4 Inicialización de una matriz.....	164
7.5 Aplicación de arreglos bidimensionales	164
Capítulo 8: Clases y objetos	
8.1 Clases versus objetos.....	169
8.1.1 Encapsulación	170
8.1.2 Herencia.....	170
8.1.3 Polimorfismo.....	170
8.2 Definición de una clase.....	171
8.3 Declaración de objetos.....	171
8.3.1 Operador new	172
8.3.2 Asignación de objetos.....	172
8.4 Definición de variables o campos	173
8.5 Definición de métodos	174
8.6 Ocultando los datos.....	178
8.7 Sobrecarga.....	181
8.8 Constructores	182
8.8.1 Constructores con parámetros	184
8.8.2 Sobrecarga de constructores	186

Capítulo 9: Herencia

9.1 Fundamentos	191
9.2 Implementando la herencia	194
9.2.1 Palabra clave extends	194
9.2.2 Herencia de variables o campos	195
9.2.3 Herencia de métodos.....	196
9.3 Palabras claves: this y super	199
9.3.1 Palabra clave this	199
9.3.2 Palabra clave super	201
a. Invocar constructores de la superclase	201
b. Acceder a miembros de la superclase.....	201
9.4 Palabras claves: abstract, final y static.....	204
9.4.1 Palabra clave abstract	204
a. Sintaxis: Clases abstractas	205
b. Sintaxis: Métodos abstractos	205
9.4.2 Palabra clave final	207
a. Constantes.....	207
b. Métodos finales	208
c. Clases finales	208
9.4.3 Palabra clave static.....	213
a. Variables estáticas	213
b. Métodos estáticos.....	213

Capítulo 10: Paquetes y clases

10.1 Paquetes.....	217
10.1.1 Definición de paquete.....	217
10.2 Paquetes y CLASSPATH	220
10.3 Importación de paquetes	221
10.4 Protección de acceso.....	225

Capítulo 11: Clases e interfaces

11.1 ¿Qué es una interfaz?.....	231
11.2 Uso de interfaces.....	234
11.2.1 Definición de una interfaz.....	234
11.2.2 Implementación de una interfaz.....	235
11.2.3 Variables e interfaces.....	236
11.2.4 Interfaces pueden extenderse	238
11.3 Ejemplo ilustrativo.....	239

Capítulo 12: Excepciones

12.1 Fundamentos	243
12.1.1. Tipos de errores	243
a. Errores de sintaxis	243
b. Errores de lógica.....	244
c. Errores de ejecución.....	245
d. Errores típicos	246
12.1.2 ¿Qué es una excepción?	247
12.1.3 Gestión de excepciones	248
12.1.4 Esquema general	248
12.1.5 Tipos de excepciones	249
12.2 Gestionando excepciones.....	250
12.2.1 Excepciones no capturadas.....	250
12.2.2 Captura de errores.....	251
12.2.3 Descripción de la excepción.....	254
12.2.4 Cláusulas catch múltiples.....	254
12.2.5 Instrucción throw.....	257
12.2.6 Palabra clave throws	259
12.2.7 Palabra clave finally	261
12.2.8 Creación de excepciones personalizadas	263
12.2.9 Excepciones encadenadas	265

Capítulo 13: Tipos enumerados

13.1 Fundamentos	269
13.2 Usar datos enumerados	270
13.2.1 Palabra clave enum.....	270
13.2.2 Uso de tipos enumerados.....	270
13.2.3 Métodos values() y valueOf().....	273
13.2.4 Tipos enumerados en línea.....	274
13.3 Los tipos enumerados son clases	276
13.3.1 Constructores y métodos.....	276
13.3.2 Implementación de interfaces	279

Capítulo 14: Autoboxing y Unboxing

14.1 Clases envolventes	283
14.1.1 Clase number	284
14.1.2 Double y Float.....	284
14.1.3 Métodos isInfinite() e isNaN().....	286
14.1.4. Byte, Short, Integer y Long.....	287
14.1.5 Character	288
14.1.6 Boolean.....	289
14.2 Autoboxing	290
14.3 Unboxing	291
14.4 Operaciones con tipos envolventes.....	292
14.4.1 Incremento y decremento con tipos envolventes	292
14.4.2 Boolean versus boolean.....	292
14.4.3 Condicionales y Unboxing.....	293
14.4.4 Resolución de métodos sobrecargados	294

Capítulo 15: Hilos (Threads)

15.1 ¿Qué son hilos?	297
15.2 Implementación de hilos.....	301
15.2.1 La clase thread	301

15.2.2 El hilo principal	304
15.2.3 La interfaz runnable	306
15.3 Sincronización.....	311
15.3.1 Palabra clave synchronized	314
15.3.2 Instrucción synchronized	317

Capítulo 16: I/O (Entrada/Salida)

16.1 Flujos (streams)	321
16.1.1 Flujos de bytes	322
16.1.2 Flujos de caracteres	322
16.1.3 Flujos predefinidos	322
16.2 Entrada / Salida por consola.....	324
16.2.1 Métodos: print() y println().....	324
16.2.2 Método read().....	324
16.3 Trabajando con flujos de caracteres	326
16.3.1 Clase reader	326
16.3.2. Entrada por consola	327
16.3.3 Clase writer	329
16.3.4 Salida por consola.....	330
16.3.5 Lectura y escritura de archivos	330

Capítulo 17: Manejo de cadenas

17.1 Fundamentos	337
17.2 Constructores de la clase String	338
17.3 Operaciones con cadenas.....	340

Fundamentos de la programación

CAPÍTULO

1

Capacidad: En este capítulo veremos un panorama general de Java. Esto nos permitirá tener una idea acerca del alcance de este extraordinario lenguaje.

1.1 HISTORIA DE JAVA

1.1.1 ORÍGENES

En los primeros años de la década de los noventa, *Sun Microsystems* decidió intentar introducirse en el mercado de la electrónica de consumo y desarrollar programas para pequeños dispositivos electrónicos. Posteriormente, decidió crear una filial, denominada *FirstPerson Inc.*, para dar margen de maniobra al equipo responsable del proyecto.

La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar códigos de escaso tamaño; por ello, James Gosling diseñó Java que originalmente se llamo *Oak* (roble).

Debido a la existencia de distintos tipos de CPUs y los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada. Se desarrolló un código «neutro» que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una «máquina hipotética o virtual» denominada **Java Virtual Machine (JVM)**. Era la JVM que interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se convirtió en el principal lema del lenguaje: «Write once, Run *everywhere*». A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje.

Como lenguaje de programación para computadoras, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa *Netscape Navigator*, produciendo una verdadera revolución en Internet. Java 1.0 oficialmente es presentado en enero de 1996. Poco tiempo después a principios de 1997 apareció Java 1.1, mejorando sustancialmente la primera versión del lenguaje. Al año siguiente, Java 1.2 se rebautizó como Java 2.

♦ Java 2

Java 2, antes llamado Java 1.2 o JDK 1.2 es la tercera versión importante del lenguaje de programación Java. Todas las versiones posteriores a Java 1.2 se conocen como Java 2. Actualmente estamos en Java 2 Versión 5.

1.1.2 ASPECTOS CRONOLÓGICOS

a. Java 1

♦ Java 1.0 (enero, 1996)

Compuesto por 8 paquetes, 212 clases. Es la primera versión pública. La presión mediática hizo que se publicara demasiado pronto; lo cual significó que el diseño del lenguaje no sea bueno y existiera numerosos errores. Respecto a la seguridad, es restrictivo por defecto.

♦ Java 1.1 (marzo, 1997)

Compuesto de 23 paquetes, 504 clases. Tiene mejoras de rendimiento en el JVM, nuevo modelo de eventos en AWT, clases anidadas, serialización de objetos, API de JavaBeans, archivos *jar*, internacionalización, API Reflection (*Reflexión*), JDBC (*Java Data base Connectivity*), RMI (*Remote Method Invocation*). Se añade la firma del código y la autenticación. Es la primera versión lo suficientemente estable.

b. Java 2

♦ Java 1.2 (diciembre, 1998)

Compuesto de 59 paquetes, 1520 clases, JFC (*Swing*), Drag and Drop, Java2D, Corba, API Collections. Se producen mejoras notables en todos los niveles. Para enfatizar esto, Sun lo renombra como «Java 2». El JDK (Java Development Kit) se renombra como SDK (Software Development Kit). Se divide en J2SE, J2EE y J2ME.

♦ Java 1.3 (abril, 2000)

Compuesto de 77 paquetes, 1595 clases. Está orientada sobre todo a la resolución de errores y a la mejora del rendimiento. Se producen algunos cambios menores como la inclusión de JNDI (*Java Naming and Directory Interface*) y la API Java Sound. También incluye un nuevo compilador de alto rendimiento JIT (*Just In Time*).

♦ Java 1.4 (2002)

Compuesto de 103 paquetes, 2175 clases. También conocido como Merlin. Mejora notablemente el rendimiento y añade, entre otros, soporte de expresiones regulares, una nueva API de entrada/salida de bajo nivel (NIO, New I/O), clases para el trabajo con Collections, procesado de XML; y mejoras de seguridad como el soporte para la criptografía mediante las *Java Cryptography Extension* (JCE), la inclusión de la *Java Secure Socket Extension* (JSSE) y el *Java Authentication and Authorization Service* (JAAS).