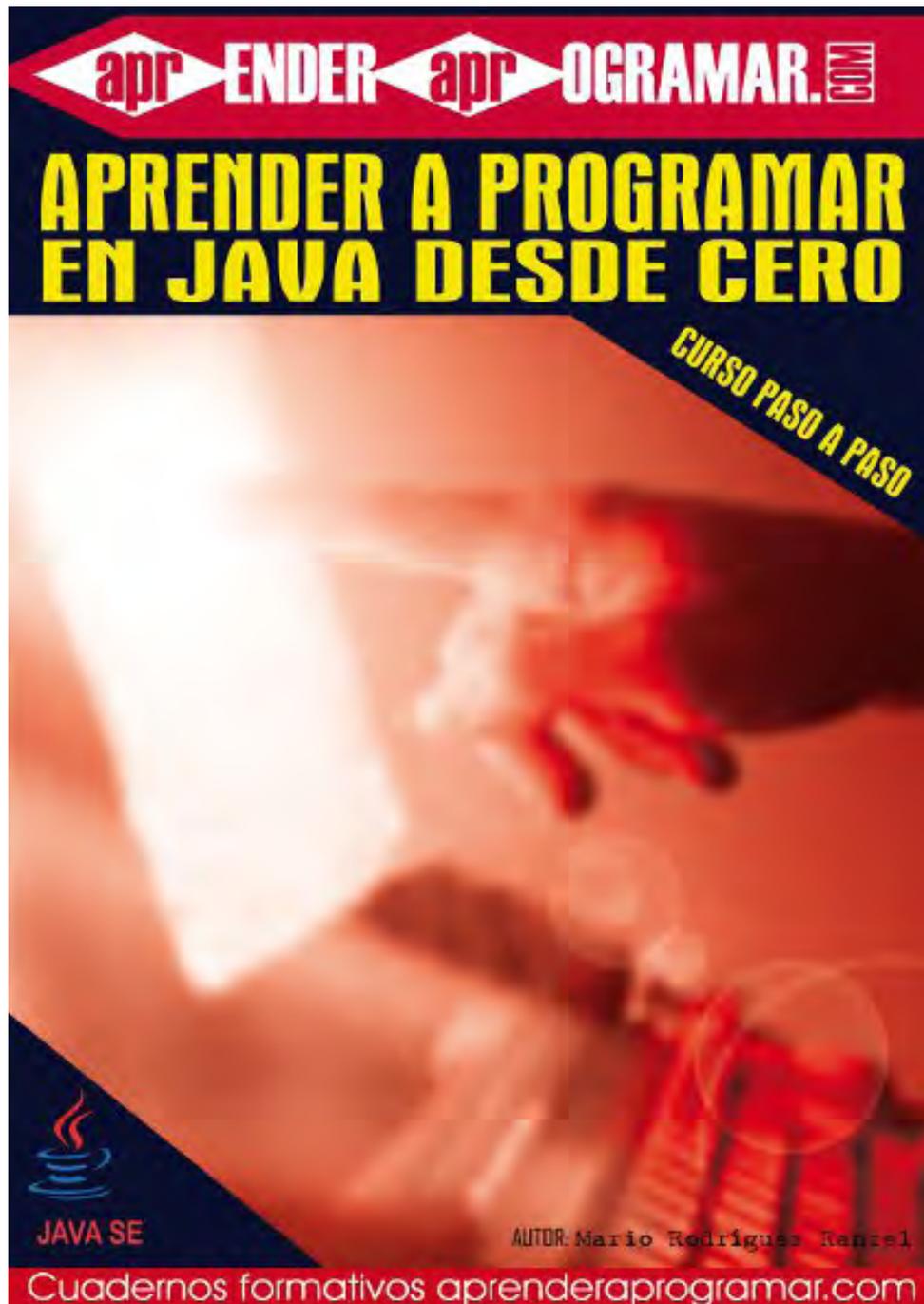


ÍNDICE DEL LIBRO Y DEL EBOOK “APRENDER A PROGRAMAR EN JAVA DESDE CERO. CURSO PASO A PASO.”



INDICE

1. INTRODUCCIÓN A JAVA

- 1.1. ¿Qué es Java?
- 1.2. ¿Es Java un lenguaje ideal para aprender a programar?
- 1.3. ¿Es Java software libre?
- 1.4. ¿Cuáles son las versiones y distribuciones Java y cuál usar?
- 1.5. ¿Qué sistema operativo debo usar para programar Java?
- 1.6. ¿Qué son las actualizaciones de Java?

2. INSTALACIÓN Y PRIMEROS PASOS CON JAVA

- 2.1. Descargar (download) Java: obtener el instalador
- 2.2. Instalación de Java en Windows (en nuestro pc)
- 2.3. Configuración de Java en Windows: JAVA_HOME, PATH
- 2.4. Máquina virtual JVM, compilador e intérprete Java. Bytecode.
- 2.5. NetBeans, Eclipse, JCreator, JBuilder... ¿Cuál es el mejor entorno de desarrollo (IDE) Java?
- 2.6. Descargar (download) BlueJ. Instalación en Windows.
- 2.7. ¿Qué es un proyecto Java? Creación del primer proyecto.
- 2.8. La ventana del editor de BlueJ
- 2.9. Escribir código (una clase) en Java con un IDE. Primer programa.
- 2.10. Comentarios en lenguaje Java y bloques en Java

3. OBJETOS Y CLASES EN JAVA. INSTANCIAS.

- 3.1. Visualizar clases y objetos con BlueJ e invocar métodos
- 3.2. Tipos de datos (variables) en Java
- 3.3. Ejemplos de uso de tipos de datos (variables). Declaración y asignación de contenido.
- 3.4. ¿Qué es una clase? Atributos, constructor y métodos
- 3.5. Métodos tipo procedimiento (...void) y tipo función (...return).
- 3.6. Métodos con y sin parámetros
- 3.7. Métodos consultores o de acceso (getters) y métodos modificadores (setters)
- 3.8. Estado de un objeto
- 3.9. Parámetros formales y parámetros actuales
- 3.10. Comprender la filosofía de métodos y clases en Java
- 3.11. Signatura de un método. Interfaz o interface.
- 3.12. Guardar los proyectos Java. Copias de seguridad.

4. SINTAXIS BÁSICA Y CONDICIONALES EN JAVA

- 4.1. Imprimir por pantalla con System.out. Concatenar cadenas. Notación de punto.
- 4.2. Operadores aritméticos en Java.
- 4.3. Operadores lógicos principales en Java.
- 4.4. Sentencia de asignación. Asignación compuesta.
- 4.5. Condicional if else / if else if en Java
- 4.6. Condicional de selección con switch en Java

5. ORGANIZAR EL CÓDIGO EN JAVA

- 5.1 Variables locales a un método o constructor. Sobrecarga de nombres.
- 5.2 Cómo crear constructores en Java. Ejemplos.
- 5.3 Clases con dos o más constructores. Sobrecarga de constructores o métodos.
- 5.4 Clases que utilizan objetos preexistentes: relación de uso entre clases. Diagrama de clases.
- 5.5 Paso de objetos como parámetros a un método o constructor
- 5.6 La sentencia new como invocación de un constructor.

6. EL API DE JAVA. BIBLIOTECAS DE CLASES.

- 6.1 Otra definición de clase: un paquete de código. Objetos del mundo real y objetos abstractos.
- 6.2 ¿Qué es y para qué sirve el API de Java?
- 6.3 Organización y forma de nombrar las librerías en el API de Java
- 6.4 Importar y usar clases del API de Java. Ejemplo: clase Math
- 6.5 Interface de clase. Ejemplo: clase String84
- 6.6 Explorar un método. El método substring de la clase String.
- 6.7 Usar métodos para evitar errores. Ejemplo método length de la clase String

7. CREAR UN PROGRAMA CON JAVA

- 7.1. Concepción de programas mediante abstracción y modularización
- 7.2 Un ejemplo de código Java muy elemental. Crear clases con campos, constructor y métodos.
- 7.3 Definición de método interno y método externo en Java
- 7.4 La palabra clave this. Contenido null por defecto de un objeto.
- 7.5 Clase con el método main: clase principal, iniciadora o “programa” principal.
- 7.6 Sintaxis y código ejemplo de uso del método main.
- 7.7 Pedir datos por consola (teclado) en java. Escape con barra invertida backslash. Print y salto de línea con \n.

8. ESTRUCTURAS DE REPETICIÓN O BUCLES, COLECCIONES DE OBJETOS Y RECORRIDOS

- 8.1 Concepto general de bucle
- 8.2 Bucle con instrucción for. Operador ++ y --. Sentencia break
- 8.3 Bucle con instrucción while. Ejemplo de uso de break.
- 8.4 Bucle con instrucción do ... while.
- 8.5 El debugger de BlueJ. Detener un programa en ejecución.
- 8.6 Pensar en objetos. Pensar una entrada de teclado como objeto.
- 8.7 El método equals. Diferencia entre igualdad e identidad entre objetos.
- 8.8 Asignación de igualdad con tipos primitivos y con objetos.
- 8.9 Repaso y ejemplos sobre igualdad, identidad y método equals.
- 8.10 Colecciones de objetos de tamaño flexible. Contenedores de objetos.
- 8.11 La clase ArrayList del API de Java.
- 8.12 Concepto de clase genérica (clase parametrizada)
- 8.13 El for extendido o bucles for each en Java.
- 8.14 Recorrer colecciones con objetos Iterator. Evitar errores tipo java.util.ConcurrentModificationException.

- 8.15 Resumen de tipos de bucles (ciclos) en Java.
- 8.16 Objetos con referencia null y excepciones tipo `java.lang.NullPointerException`.
- 8.17 Autoboxing y unboxing. Conversión automática de tipos envoltorio a primitivos y viceversa.
- 8.18 Objetos anónimos.
- 8.19 Colecciones de tamaño fijo: arrays, arreglos o formaciones.
- 8.20 Campo `length` para saber el número de elementos de un array
- 8.21 Uso de ciclos `for each` con arrays
- 8.22 Resumen de colecciones

9. AVANZAR EN EL USO Y CONOCIMIENTO DE JAVA

- 9.1 Conversión de tipos en java
- 9.2 Método `valueOf` para conversión de tipos
- 9.3 Ejemplo de conversión de tipos. Tipo obtenido frente a tipo requerido. Métodos `get` y `remove` de `ArrayList`
- 9.4 Generar números aleatorios en Java. Clase `Random`.
- 9.5 Variables de clase o estáticas y constantes. Palabras clave `static` y `final`.
- 9.6 Organizar un proyecto Java en paquetes (packages). Cláusulas `package` e `import`
- 9.7 Formas de nombrar packages, jerarquización y visibilidad de clases. Los packages en BlueJ.
- 9.8 Copiar arrays y comparar arrays. Identidad e igualdad entre arrays.
- 9.9 La clase `Arrays` del API de Java. Métodos `equals` y `copyOf`.
- 9.10 Rellenar un array con un valor u objeto. Método `fill` de la clase `Arrays`.
- 9.11 Interfaz o interface en Java. Ampliación del concepto.
- 9.12 Polimorfismo en Java. Primera aproximación.
- 9.13 Transformar un array en una lista con el método `asList` de la clase `Arrays`. Constructores que usan colecciones.
- 9.14 Documentar un proyecto con `javadoc`. Comentarios, símbolos, tags.
- 9.15 Tipos enumerados (`enum`) en Java.
- 9.16 Método `values`. Enumerados clases con campos y constructores.
- 9.17 Métodos de clase o `static` frente a métodos de instancia. Comprender el método `main`.

10. HERENCIA EN JAVA

- 10.1 ¿Qué es la herencia en programación orientada a objetos?
- 10.2 Jerarquías de herencia. Organización y acceso entre clases.
- 10.3 Ejemplo de herencia y uso de palabras clave `extends` y `super`. Constructores con herencia.
- 10.4 Ejemplo de herencia descendente o herencia simple.
- 10.5 Jerarquía de tipos. Subtipos. Polimorfismo y variables polimórficas.
- 10.6 Conversión de tipos (enmascaramiento). Hacer casting y `ClassCastException`. 166
- 10.7 Determinación del tipo de variables con `instanceof`.
- 10.8 Tipo estático y tipo dinámico de variables. Sobreescritura (redefinición) de métodos. Métodos polimórficos.
- 10.9 Ejercicio ejemplo de código con herencia, polimorfismo de variables y métodos, y sobreescritura de métodos.
- 10.10 Uso de la palabra clave `super` para llamar a métodos.
- 10.11 Modificadores de acceso `public`, `private` y `protected`.
- 10.12 Sobrecribir métodos de la clase `Object`: método `toString()`.

- 10.13 Sobreescribir métodos de la clase Object: método equals.
- 10.14 Clases y métodos abstractos.
- 10.15 Clases abstractas en el API de Java
- 10.16 Herencia múltiple. Interfaces en Java.
- 10.17 Para qué sirven las interfaces en Java.
- 10.18 Ejemplo sencillo de interface en Java.
- 10.19 Implementar una interface del API de Java.
- 10.20 Resumen de herencia en Java.

11. **PROGRESAR COMO PROGRAMADORES JAVA: SWING, GESTIÓN DE ERRORES Y MÁS ALLÁ.**

- 11.1 ¿Qué hemos aprendido y qué no hemos aprendido con este curso?

12. **INDICE DE TÉRMINOS**

A

abstracción, 15, 39, 48, 75, 85, 87, 91, 158, 161, 182
abstract (palabra clave), 177, 183
 acoplamiento, 190
 actualizaciones Java, 19
 add (método de la clase ArrayList), 115
 aleatorio. *Véase* números aleatorios
 algoritmia, 13
 and (operador de conjunción &&), 59
 API (Application Programming Interface), 19
 api de Java, 19, 77, 80, 81, 118, 136, 145, 152, 161
 aprenderaprogramar.com, 13, 191
ArrayList, 114, 115, 118, 128, 146, 148, 165, 186
 Arrays (clase del api Java), 141, 144, 147, 156
 arrays (tipo de datos), 43, 124, 126, 139, 141, 144
 arreglo. *Véase* arrays (tipo de datos)
 asignación, 45, 59, 61, 96, 111, 125, 165
 asignación compuesta, 59
 asList (método de la clase Arrays), 147
 atributos, 133
 atributos de clase (estáticos), 133
 atributos de instancia, 39, 46, 52, 65, 68, 82, 84, 106
 atributos de objeto. *Véase* atributos de instancia
 autoboxing, 123
 AutoLayout, 37
 awt (biblioteca api java), 79, 145, 191

B

backslash. *Véase* barra invertida

banco de objetos (object bench), 41, 89
 barra invertida, 101
 bibliotecas. *Véase* api de Java
 bloque, 39
 BlueJ, 31, 32, 35, 38, 55, 70, 88, 105, 156
 boolean / Boolean (tipos), 43
 botones, 79
 break, 61, 103, 119
 breakpoint, 105
 bucles, 102
 bug (error), 105
 búsqueda dinámica de métodos, 171
 byte / Byte (tipos), 43
 bytecode, 26, 33

C

Calendar, 79
 campos. *Véase* atributos de instancia
 capacidad de una colección, 115
 case (palabra clave para switch), 61
 casting, 167
 char / Character (tipos), 43
 clase, 39, 46, 67, 75, 91, 160
 clase enumerada. *Véase* enumerados (tipo)
 clase genérica o parametrizada, 118
 clases abstractas, 177, 182, 184, 189
 clases internas, 190
 clases sin campos o atributos, 74
 ClassCastException, 167
 clone (método), 113
 codepad, 89

código fuente, 17, 23, 26, 33, 36, 78, 85, 170
 código máquina, 26, 30, *Véase* bytecode
 cohesión, 190
 colas, 114
 colecciones de objetos, 114, 120, 124, 126, 147
 Collection (interface), 148
 Collections (clase), 190
 comentarios en código Java, 38, 93, 149
 Comparable (interface), 188
 comparar arrays, 141
 comparar objetos, 110, 175, *Véase* equals
 compareTo (método), 189
 compilación, 25, 28, 35, 170, *Véase* compilador
 compilador, 17, 23, 25, 28, 31, 38, 77, 82, 169
 concat (método de la clase String), 86, 175, 179
 concatenación de cadenas de texto (Strings), 57, *Véase*
 concat (método de la clase String)
 ConcurrentModificationException, 120
 condicionales, 60, 61
 configuración de Java en Windows, 24
 conjunto, 114, 123
 constantes, 132, 152, 183, 187
 constructor, 46, 51, 63, 65, 73, 75, 79, 82, 85, 91, 108, 113
 constructores con colecciones como parámetro, 148
 constructores con herencia, 163
 constructores de subclases. *Véase* constructores con
 herencia
 constructores múltiples o sobrecargados, 67
 contains (método de la clase String), 86, 121
 conversión de tipos, 127, 128, 166, *Véase* casting
 copiar arrays, 139
 copiar objetos, 113
 copias de seguridad, 55
 cuerpo, 47, 157, 169, 178, 183

D

Date, 79
 debugger, 30, 105
 declaración de ámbito. *Véase* modificadores de acceso
 declaración de campos. *Véase* declaración de variables
 declaración de constantes, 133
 declaración de variables, 46, 64, 67, 118, 123, 133
 default (cláusula en switch), 62
 depurador. *Véase* debugger
 desbordamiento, 116
 descarga de Java (download), 21
 diagrama de clases, 70, 94, 117, 162, 164, 166, 170, 179, 184
 diseño de clases, 161, 190
 diseño de programas, 190
 distribuciones Java, 17, 19
 divide y vencerás, 91
 do ... while, 104

documentar proyectos y código Java, 36, 79, 148, *Véase*
 javadoc
 double / Double (tipos), 43

E

Eclipse, 30, 38, 134
 ejecutar línea a línea, 106
 else, 60
 else if, 60, 69
 else vacío, 60, 116
 encapsulamiento, 173, 174
 enmascaramiento de tipos. *Véase* casting
 enteros. *Véase* tipos de datos Java
 entorno de desarrollo, 30, 37, 56
 entrada de teclado, 100, 106
 enum. *Véase* enumerados (tipo)
 enumerados (tipo), 62, 152
 envoltorio (tipos envoltorio), 43, 123
 equals (método), 108, 110, 113, 141, 161, 175
 error. *Véase* bug
 errores de compilación, 35, 37, 56, 61
 errores en tiempo de ejecución, 123, 125, 167
 escape. *Véase* barra invertida
 estado de un objeto, 52
 eventos, 191
 excepción. *Véase* error, bug
 extends (palabra clave), 159, 161, 184, 189

F

fill (método de la clase Arrays), 144
 filosofía Java, 16, 32, 53
 final (palabra clave), 133, 183, 185
 float / Float (tipos), 43
 for, 102, 118, 120, 125
 for each. *Véase* for extendido
 for extendido (for each), 118, 126, 188
 formación. *Véase* arrays (tipo de datos)
 función. *Véase* métodos tipo función

G

genérica. *Véase* clase genérica o parametrizada
 get (método de la clase ArrayList), 115, 129
 get (prefijo para métodos), 51
 getters. *Véase* métodos consultores o de acceso
 Graphical User Interface (GUI), 145
 GUI. *Véase* Graphical User Interface

H

HashMap, 78, 79, 115, 118, 190
 HashSet, 79, 114, 190
 hasNext (método de Iterator), 121
 herencia, 157
 herencia múltiple, 183, 185, 189
 herencia simple, 158, 164
 historia (evolución) de Java, 18
 HTML, 148, 150

I

IDE. *Véase* entorno de desarrollo
 identidad entre objetos, 109, 113, 139
 if (condicional), 60, 92, 96
 igualdad entre objetos, 110, 113, 140, 141, *Véase* equals
 implements (palabra clave), 183, 188, 189
 import, 79, 82, 84, 115, 135, 138, 141
 incompatibilidad de tipos, 111, 148, 167
 IndexOutOfBoundsException, 116
 inicialización, 46, 65, 97, 164
 inspector de objetos, 52, 68
 instalar Java, 20, 23
 instanceof (palabra clave), 168, 181
 instancias. *Véase* objetos
 int (tipo), 43
 Integer (tipo), 43, 113, 127, 189
 interface (como signatura), 55, 85, 145
 interface Java, 114, 145, 147, 183, 186, 189
 Interfaz Gráfica de Usuario. *Véase* GUI
 intérprete Java, 23, 25, 28
 interrupción de bucles. *Véase* break
 iterator (método), 120
 Iterator (tipo para recorrer colecciones), 120

J

J2SE, J2EE, J2ME, 17
 java.lang, 79, 81
 JAVA_HOME, 24
 javadoc, 38, 148
 JBuilder, 31
 JCreator, 31
 JDK, 13, 19, 20, 31, 77
 jerarquía de herencia, 160
 jerarquía de tipos, 165
 JRE, 15, 23
 JVM. *Véase* máquina virtual Java

L

length (campo de un array), 126
 length (método de la clase String), 86, 89
 librerías. *Véase* api de Java
 ligadura dinámica. *Véase* búsqueda dinámica de métodos
 LinkedList, 79, 114, 124, 146, 147, 166, 186, 190
 Linux, 13
 List (interface), 146, 166, 184, 186
 lista enlazada. *Véase* LinkedList
 long / Long (tipos), 43

M

Macintosh, 13
 main (método iniciador), 98, 133, 153, 156
 máquina virtual Java, 17, 23, 25, 31, 34, 77, 136, 170
 Math (clase), 81, 130, 156
 método externo, 95
 método interno, 95, 108
 método main. *Véase* main (método iniciador)
 método principal. *Véase* main (método iniciador)
 métodos, 49, 53, 63, 87, 95, 173
 métodos abstractos, 177, 182, 183, 187, 189
 métodos consultores o de acceso, 51
 métodos de clase (estáticos), 156
 métodos modificadores, 51
 métodos observadores. *Véase* métodos consultores o de acceso
 métodos polimórficos, 170
 métodos tipo función, 49, 54, 58, 64, 128, 150
 métodos tipo procedimiento, 50, 55
 modificadores de acceso, 174
 modularización, 91

N

NetBeans, 30, 38
 new, 73, 82, 113, 122, 155, 177
 next (método de Iterator), 121
 nombre completo de una clase, 79
 not (operador de negación !), 59
 notación de punto, 57
 null, 65, 96, 122, 144
 NullPointerException, 122
 números aleatorios, 130, 151

O

Object (clase), 158, 161, 166, 169, 174, 188
 objetos, 39, 51, 68, 106, 114
 objetos anónimos, 124, 169

ocultamiento de la información, 54, 85
 operadores aritméticos, 58
 operadores lógicos, 59
 or (operador de disyunción ||), 59
 orden natural, 189
 ordenar, 19, 78, 141
 ordinal, 62
 orientación a objetos, 13, 16, 28, 39, 54, 70, 87, 98, 106, 158

P

packages, 134, 136, 145
 parametrización. *Véase* clase genérica o parametrizada
 parámetro actual, 52
 parámetro formal, 52
 parámetros, 50, 55, 65, 73, 87, 91, 96, 149, 157, 163
 PATH, 25
 polimorfismo, 147, 166, 170, 188
 print, 101
 println, 57
 private, 45, 48, 54, 64, 174
 procedimiento. *Véase* métodos tipo procedimiento
 programa, 37, 91, 98, 106, 157, 170, 186
 programa principal. *Véase* main (método iniciador)
 protected, 45, 54, 174
 prototipos, 55
 pruebas, 30, 105, 106
 pseudoaleatorio, 130
 public, 45, 48, 54, 64, 136, 174
 punto de interrupción. *Véase* breakpoint

R

Random (clase), 79, 130
 read (método de System.in), 127
 recursión, 75
 redefinición de métodos. *Véase* sobrescritura
 relación de uso entre clases, 68
 rellenar un array. *Véase* fill (método de la clase Arrays)
 remove (método de Iterator), 121
 remove (método de la clase ArrayList), 129
 retorno de carro, 101
 return, 49, *Véase* métodos tipo función

S

salto de línea, 101
 Scanner (clase), 100
 set (método de la clase ArrayList), 115
 set (prefijo para métodos), 51
 setters. *Véase* métodos modificadores
 short / Short (tipos), 43

signatura, 55, 78, 85, 89, 145, 156, 169, 177
 Singleton (patrón de diseño), 191
 sistema de archivos, 56, 136
 sistema operativo, 13, 19, 24, 26, 32, 145
 size (método de la clase ArrayList), 115
 sobrecarga de constructores o métodos, 68
 sobrecarga de nombres, 64
 sobrescritura, 169, 175, 178
 Stack, 146
 static, 133, 156, 183
 String (clase), 44, 49, 62, 71, 77, 79, 84, 96, 110, 112, 121
 StringIndexOutOfBoundsException, 89
 StringTokenizer, 79
 subclase, 160, 163, 165, 169, 174, 178, 182, 189
 substring (método de la clase String), 87, 110
 subtipo. *Véase* subclase
 super (palabra clave), 163, 173
 superclase, 160, 161, 165, 169, 173, 177, 182, 189
 supertipo. *Véase* superclase
 sustitución, 165
 Swing (biblioteca gráfica), 18, 79, 145, 191
 switch, 61
 System (clase), 77
 System.in, 101, 127
 System.out, 57
 System.out.print, 101
 System.out.println, 57

T

tabulador (carácter de escape \t), 154
 tags, 150
 this (palabra clave), 91, 95
 tiempo de ejecución, 169
 tipo dinámico, 169, 172, 178
 tipo enumerado. *Véase* enumerados (tipo)
 tipo estático, 169, 179
 tipo requerido, 51, 128
 tipos de datos Java, 42
 tipos envoltorio. *Véase* envoltorio
 tipos primitivos, 42, 70, 96, 108, 111, 123, 128, 161
 toString (método), 153, 161, 170, 174, 182, 188
 TreeSet, 79

U

unboxing, 123
 Unix, 27

V

valueOf (método para conversión de tipos), 127

values (método para enumerados), 154
variables, 42, 63, 132, 165
variables de clase, 133
variables de entorno, 24
variables de instancia. *Véase* atributos de instancia
variables locales, 63, 96, 106
variables polimórficas, 166, *Véase* polimorfismo
variables que referencian a objetos. *Véase* objetos
vector. *Véase* array (tipo de datos)
Vector (clase api Java), 190
versiones de Java, 17
visibilidad de clases, 134, 138
visibilidad de métodos y campos. *Véase* modificadores de
 acceso
void, 50

W

while, 103, *Véase* do ... while
Windows, 13, 19, 20, 22, 24, 32