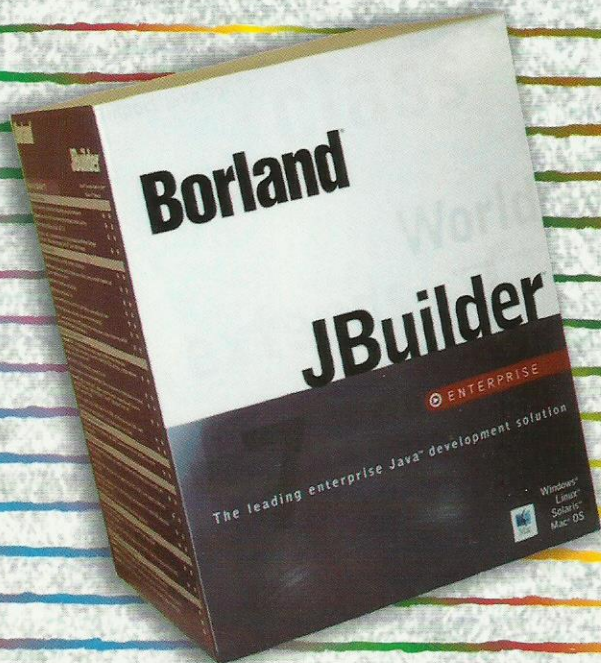


*Guía Práctica
para usuarios*

JBuilder 7

Francisco Charte



ANAYA
MULTIMEDIA

Las **Guías Prácticas** de Anaya Multimedia son los manuales más útiles que usted podrá encontrar. En esta colección hallará lo último sobre ordenadores personales y la información más actual sobre las últimas versiones en lenguajes de programación, hojas de cálculo, bases de datos, procesadores de textos, programas de diseño, etc.

En definitiva, estas **Guías** pondrán a su disposición una fuente útil e inagotable de datos, con la información más novedosa sobre los productos más importantes y de mayor relevancia en el mercado informático.

OTROS TEMAS DE LA COLECCIÓN

3ds max 4	Open Source
Access 2002	PageMaker 7
Acrobat 5	Photoshop 6
C# y Microsoft .NET	PHP 4
Comercio electrónico	Plug-ins de Reyes para 3ds max 4
ContaPlus 2001	Pocket PC
CorelDRAW 10	PowerPoint 2002
Defensa contra hackers	Premiere 6
Div X	Principios del diseño Web
Excel 2002	Red Hat Linux 7
FileMaker Pro 5.5	Redes Locales
Flash MX	Softimage XSI 2
FreeHand 10	Técnicas de retoque con Photoshop
Internet Explorer 6	Telefonía móvil digital
J2ME	Visual Basic .NET
JSP	Visual Studio .NET
LightWave 3D 7	Windows XP Home Edition
Mac OS X	Windows XP Professional
Messenger	Word 2002
MP3	Windows XP. Registro y configuración
Netscape communicator 6	XML

NIVELES	Iniciación	TIPO DE LIBRO
	✓ Básico	
	✓ Medio	TEMÁTICA
	Avanzado	
Profesional/Experto		



<http://www.AnayaMultimedia.es>

2335289

ISBN 84-415-1324-4



9 788441 151324

Guía práctica JBuilder 7

© Francisco Charte

Introducción

- Rápida historia de Java
- Ediciones de Java 2
- JBuilder 7
- Objetivos de este libro

Cómo usar este libro

- Estructura de la guía
- Uso de los ejemplos
- Convenciones tipográficas

1. Instalación

- 1.1. Introducción
- 1.2. Requisitos para la instalación
- 1.3. Paquetes de instalación
- 1.4. Inicio de la instalación
 - 1.4.1. Proceso de instalación
- 1.5. Activación del producto

2. Toma de contacto

- 2.1. Introducción
- 2.2. Generalidades sobre el entorno
- 2.3. Configuración del entorno
- 2.4. Inicio de un nuevo proyecto
 - 2.4.1. Añadimos una clase al proyecto
 - 2.4.2. Edición del código
 - 2.4.3. Compilación y ejecución
 - 2.4.4. Ejecución desde la línea de comandos
- 2.5. Una interfaz gráfica de usuario
 - 2.5.1. Elementos de la aplicación
 - 2.5.2. Diseño de la interfaz
 - 2.5.3. Código de implementación
 - 2.5.4. Compilación y ejecución
- 2.6. Resumen

3. Familiarizándonos con el entorno

- 3.1. Introducción
- 3.2. Gestión de proyectos
 - 3.2.1. El Panel de proyectos
 - 3.2.2. Manipulación de un elemento
 - 3.2.3. Propiedades del proyecto
- 3.3. Estructura de un elemento
- 3.4. El Panel de contenidos
 - 3.4.1. Diseño visual de interfaces
 - 3.4.2. Edición de propiedades
 - 3.4.3. Edición de código
 - 3.4.4. Esquemas de código
- 3.5. Asistentes y herramientas diversas
 - 3.5.1. Empaquetado de soluciones
 - 3.5.2. Documentación automática del código
 - 3.5.3. Otros asistentes
 - 3.5.4. Herramientas diversas
- 3.6. Depuración
- 3.7. Resumen

4. Conceptos de Java

- 4.1. Introducción
- 4.2. Estructuración del código
 - 4.2.1. Módulos de clases Java
 - 4.2.2. Definición de paquetes
 - 4.2.3. Referencias a paquetes existentes
 - 4.2.4. Paquetes estándar
- 4.3. Clases
 - 4.3.1. Definición de una clase
 - 4.3.2. Clases internas

- 4.3.3. Modificadores de visibilidad
 - 4.3.4. Definición de subclases
 - 4.3.5. Clases abstractas y finales
 - 4.4. Tipos de datos
 - 4.4.1. Declaración de variables
 - 4.4.2. Visibilidad y vida de las variables
 - 4.4.3. Operadores
 - 4.4.4. Valores literales
 - 4.4.5. Conversiones explícitas
 - 4.4.6. Arreglos
 - 4.4.7. Cadenas de caracteres
 - 4.4.8. La clase Object
 - 4.5. Estructuras de control
 - 4.5.1. Implementación de métodos
 - 4.5.1.1. Modificadores de métodos
 - 4.5.1.2. Sobrecarga de métodos
 - 4.5.1.3. Redefinición de métodos
 - 4.5.1.4. Constructores
 - 4.5.1.5. Referencias this y super
 - 4.5.2. Estructuras condicionales
 - 4.5.3. Estructuras de repetición
 - 4.5.4. Control de excepciones
 - 4.6. Interfaces
 - 4.6.1. Definición de una interfaz
 - 4.6.2. Implementación de interfaces en clases
 - 4.6.3. Uso polimórfico de los objetos
 - 4.7. Resumen
- 5. Interfaces de usuario
 - 5.1. Introducción
 - 5.2. AWT, JFC y Swing
 - 5.3. Diseño de una interfaz sencilla
 - 5.3.1. Inicio de la nueva aplicación
 - 5.3.2. Personalización de la interfaz
 - 5.3.2.1. Adaptación de los botones
 - 5.3.2.2. Distribución de los componentes en el contenedor
 - 5.3.2.3. Inserción de componentes para los datos
 - 5.3.3. Una clase para almacenamiento de datos
 - 5.3.4. Gestión de los eventos de botones
 - 5.3.5. Ejecución del programa
 - 5.4. Análisis del código generado
 - 5.4.1. Una visión general
 - 5.4.2. Inicio de la ejecución
 - 5.4.3. Construcción de la clase Agenda
 - 5.4.4. Definición de la clase AppFrame
 - 5.4.5. Preparación de la ventana
 - 5.4.6. Resto del código
 - 5.5. Gestores de distribución
 - 5.5.1. Selección de un gestor de distribución
 - 5.5.2. Propiedades del gestor de distribución
 - 5.5.3. Relación entre gestor y componentes
 - 5.5.4. Contenedores dentro de contenedores
 - 5.6. Componentes Swing
 - 5.7. Resumen
- 6. Applets
 - 6.1. Introducción
 - 6.2. Esquema de funcionamiento de un applet
 - 6.2.1. Transferencia del servidor al cliente
 - 6.2.2. La etiqueta <applet>
 - 6.2.3. La JVM del cliente
 - 6.2.4. Java Plug-in
 - 6.2.4.1. Sustitución de la etiqueta <applet>
 - 6.2.4.2. Java Plug-in HTML Converter
 - 6.2.4.3. Instalación del Java Plug-in
 - 6.2.4.4. Panel de control del Java Plug-in
 - 6.3. Lógica de un applet
 - 6.3.1. Clases Applet y JApplet
 - 6.3.2. Ciclo de vida del applet
 - 6.3.3. Acceso a datos del entorno
 - 6.4. Desarrollo de un applet
 - 6.4.1. Inicio de un nuevo applet
 - 6.4.2. Análisis del código
 - 6.4.3. Diseño de la interfaz
 - 6.4.4. Implementación de funcionalidad
 - 6.4.5. Ejecución en appletviewer

- 6.4.6. Ejecución en un navegador
 - 6.5. Uso de hilos paralelos de ejecución
 - 6.5.1. Implementación de la interfaz Runnable
 - 6.5.2. Control del hilo de ejecución
 - 6.6. Resumen
 - 7. Acceso a datos
 - 7.1. Introducción
 - 7.2. Estructuras de datos
 - 7.2.1. Arreglos
 - 7.2.2. Pilas y vectores
 - 7.2.2.1. Trabajo con enumeraciones
 - 7.2.2.2. Vectores especializados
 - 7.2.3. Diccionarios y arreglos asociativos
 - 7.2.4. Colecciones
 - 7.3. Servicios de entrada/salida
 - 7.3.1. Flujos de datos genéricos
 - 7.3.2. Entrada y salida por consola
 - 7.3.3. Flujos con almacenamiento temporal
 - 7.3.4. Flujos a y desde archivos
 - 7.3.5. Manipulación de archivos
 - 7.4. Acceso a bases de datos
 - 7.4.1. Administración de JDataStore
 - 7.4.2. JDataStore Explorer
 - 7.4.3. Database Pilot
 - 7.4.4. Componentes de acceso a datos
 - 7.4.5. Componentes dbSwing
 - 7.5. Resumen
 - 8. Trabajo con XML
 - 8.1. Introducción
 - 8.2. ¿Qué es XML?
 - 8.2.1. Trabajo con etiquetas
 - 8.2.2. Visualización de los documentos
 - 8.2.3. Estructura de un documento XML
 - 8.2.4. Estructura de los datos
 - 8.2.5. Conjuntos de caracteres en XML
 - 8.2.6. Documentos XML bien formados
 - 8.2.7. Documentos XML válidos
 - 8.3. Analizadores de XML
 - 8.4. Validación de documentos
 - 8.4.1. ¿Qué es una DTD?
 - 8.4.2. Análisis de la estructura del documento
 - 8.4.3. Elaboración de la DTD
 - 8.4.4. Cómo aplicar una DTD a un documento XML
 - 8.4.5. Validación del documento
 - 8.5. Atributos o propiedades
 - 8.6. Introducción a XSL
 - 8.6.1. Transformación de documentos
 - 8.6.2. XSL básico
 - 8.6.3. Cómo aplicar plantillas
 - 8.6.4. Selección de datos
 - 8.6.5. Las posibilidades de XSL
 - 8.7. Introducción a XSLT y XPath
 - 8.8. Clases XML
 - 8.8.1. Generación automática de clases
 - 8.8.2. Recuperación de un documento XML
 - 8.8.3. Acceso a los datos del documento
 - 8.8.4. Creación y modificación de datos
 - 8.9. Resumen
 - 9. JavaBeans
 - 9.1. Introducción
 - 9.2. ¿Qué es un bean?
 - 9.3. Desarrollo de un componente simple
 - 9.3.1. Análisis del código
 - 9.3.2. Diseño del componente
 - 9.3.3. Administración de las propiedades del componente
 - 9.3.4. Implementación de la lógica
 - 9.3.5. Comprobación del componente
 - 9.4. Instalación y uso del componente
 - 9.5. Otras posibilidades
 - 9.6. Resumen
 - 10. Servlets
 - 10.1. Introducción
 - 10.2. ¿Qué es un servlet?

- 10.3. Funcionamiento de un servlet
 - 10.3.1. Superclases e interfaces
 - 10.3.2. La clase HttpServlet
 - 10.3.3. Clases HttpServletRequest y HttpServletResponse
- 10.4. El asistente para creación de servlets
 - 10.4.1. Parámetros de creación del servlet
 - 10.4.2. Análisis del código generado
 - 10.4.3. Compilación y ejecución del servlet
- 10.5. Resumen

- 11. Java Server Pages
 - 11.1. Introducción
 - 11.2. ¿Qué es una JSP?
 - 11.2.1. Estructura de una JSP
 - 11.2.2. Proceso de una JSP
 - 11.2.3. JSP en JBuilder
 - 11.3. Creación de una JSP simple
 - 11.3.1. El componente JSPSimpleBean
 - 11.3.2. Análisis del documento jsp
 - 11.3.3. Ejecución de la JSP
 - 11.3.4. Vistas Web y locales
 - 11.4. Directivas, expresiones y scriptlets
 - 11.4.1. Inclusión de directivas
 - 11.4.2. Evaluación de expresiones
 - 11.4.3. Codificación de scriptlets
 - 11.4.4. Declaraciones
 - 11.5. Objetos predefinidos
 - 11.6. Resumen

- 12. Aplicaciones distribuidas
 - 12.1. Introducción
 - 12.2. Implementaciones RPC
 - 12.3. ¿Cómo funciona RMI?
 - 12.3.1. La interfaz genérica del objeto servidor
 - 12.3.2. Implementación de una clase remota
 - 12.3.3. Creación del objeto remoto
 - 12.3.4. Configuración de compilación y ejecución
 - 12.3.5. Objetos remotos y seguridad
 - 12.3.6. Diseño de un cliente
 - 12.4. Pase de parámetros entre objetos
 - 12.4.1. Parámetros que son objetos no remotos
 - 12.4.2. Parámetros que son objetos remotos
 - 12.5. RMI y applets Java
 - 12.6. Ventajas de RMI
 - 12.7. Limitaciones de RMI
 - 12.8. Resumen

Índice alfabético