

Contents

Acknowledgments	xv
Introduction	xvii
Part I: Thinking Like a Java Developer	1
Chapter 1: Key Java Language Features and Libraries	3
Introducing Derby	3
Using Derby	4
Language Features Added in Java 5	7
Generics	7
Enhanced for Loop	16
Additions to the Java Class Library	17
Variable Arguments	18
Boxing and Unboxing Conversions	19
Static Imports	21
Enumerations	24
Metadata	26
Important Java Utility Libraries	34
Java Logging	35
Java Preferences	70
Summary	77
Chapter 2: Tools and Techniques for Developing Java Solutions	79
Principles of Quality Software Development	80
Habits of Effective Software Development	81
Communicate	81
Model	81
Be Agile	81
Be Disciplined	82
Trace Your Actions to Need	82
Don't Be Afraid to Write Code	83
Think of Code as a Design, not a Product	83

Contents

Read a Lot	84
Build Your Process from the Ground Up	84
Manage Your Configuration	84
Unit Test Your Code	85
Continuously Integrate	85
Maintaining Short Iterations	86
Measure What You Accomplished — Indirectly	87
Track Your Issues	87
Development Methodology	88
Waterfall Methodology	88
Unified Process	90
Extreme Programming	91
Observations on Methodology	92
Practical Development Scenarios	93
Ant	93
Maven 2	101
TestNG	106
XDoclet	110
JMeter	117
Summary	120
<hr/>	
Chapter 3: Exploiting Patterns in Java	123
Why Patterns Are Important	124
Keys to Understanding the Java Programming Language	124
Keys to Understanding Tools Used in Java Development	125
Keys to Developing Effective Java Solutions	126
Building Patterns with Design Principles	127
Designing a Single Class	127
Creating an Association between Classes	128
Creating an Interface	129
Creating an Inheritance Loop	129
Important Java Patterns	131
Adapter	131
Model-View-Controller	134
Command	142
Strategy	146
Composite	150
Summary	154

Part II: A Broad Understanding of Java APIs, Tools, and Techniques	155
Chapter 4: Developing Effective User Interfaces with JFC	157
Layout Managers	158
BorderLayout	158
BoxLayout	164
FlowLayout	173
GridLayout	177
GridBagLayout	189
SpringLayout	194
CardLayout	202
GroupLayout	208
Mustang Release Desktop Enhancements	214
Managing Navigation Flows in Swing Applications	225
Summary	235
Chapter 5: Persisting Your Application Using Files	237
Application Data	237
Saving Application Data	239
Sample Configuration Data Model for an Application	239
Java Serialization: Persisting Object Graphs	241
Key Classes	242
Serializing Your Objects	243
Extending and Customizing Serialization	257
When to Use Java Serialization	261
JavaBeans Long-Term Serialization: XMLEncoder/Decoder	262
Design Differences	262
Serializing Your JavaBeans	265
When to Use XMLEncoder/Decoder	269
Flexible XML Serialization: Java API for XML Binding (JAXB)	270
Sample XML Document for the Configuration Object	271
Defining Your XML Format with an XML Schema	273
JAXB API Key Classes	280
Marshalling and Unmarshalling XML Data	281
Creating New XML Content with JAXB-Generated Classes	283
Using JAXB-Generated Classes in Your Application	283

Contents

Annotating Existing Java Classes for Use with JAXB	290
When to Use JAXB	307
Where JAXB Fits in the JDK	308
Summary	308
Chapter 6: Persisting Your Application Using Databases	311
JDBC API Overview	312
Setting Up Your Environment	313
JDBC API Usage in the Real World	313
Understanding the Two-Tier Model	313
Understanding the Three-Tier Model	314
Effectively Using JDBC 4.0	315
Overview	315
Managing Connections	316
Understanding Statements	318
Utilizing Result Sets	328
Advanced Concepts	333
Hibernate	335
Hibernate Components	336
Hibernate Example	341
Summary	354
Chapter 7: Developing Web Applications Using the Model 1 Architecture	355
What Is Model 1? Why Use It?	356
JSP 2.0 Overview	357
Integrated Expression Language (EL)	365
JSTL 1.1 Overview	366
Developing Your Web Application Visualizations with JSTL 1.1	370
Developing Your Web Application Visualizations with JSP 2.0	376
AJAX	381
Summary	391
Chapter 8: Developing Web Applications Using the Model 2 Architecture	393
The Problem	393
What Is Model 2?	393
Why Use Model 2?	395
Developing an Application with WebWork	396
What Is Inversion of Control and Why Is it Useful?	397
Architecture	399

Extending the Framework to Support Hibernate	402
Preventing the Hanging Session	403
Defining Your Domain Model	405
Implementing Your Use Cases with Actions	412
Developing Your Views	415
Configuring Your Application	422
Adapting to Changes	424
Summary	426
Chapter 9: Interacting with C/C++ Using Java Native Interface	427
A First Look at Java Native Interface	427
Creating the Java Code	428
Creating the Native Code and Library	429
Executing the Code	431
Java Native Interface	432
Data Types	432
Strings in JNI	432
Arrays in JNI	436
Working with Java Objects in C/C++	442
Handling Java Exceptions in Native Code	449
Working with Object References in Native Code	451
Advanced Programming Using JNI	455
Developing an Email Client	460
System Design	460
User Interface	461
Summary	471
Chapter 10: EJB 3 and the Java Persistence API	473
New Features	474
Java Persistence API (JPA)	475
Entities	475
Query Language	475
EntityManager	476
What Are Session Beans? The Demise of Entity Beans?	480
Interceptors	481
EJB 3 and Java Persistence API Web Component Examples	485
Summary	517

Chapter 11: Communicating between Java Components and Components of Other Platforms **519**

Component Communication Scenarios	520
News Reader: Automated Web Browsing	520
A Bank Application: An EJB/Java EE Client	520
A Portal: Integrating Heterogeneous Data Sources and Services	521
Overview of Interprocess Communication and Basic Network Architecture	521
Sockets	522
The Java Socket API	523
Implementing a Protocol	530
Remote Method Invocation	542
Core RPC/RMI Principles	542
Common Object Request Broker Architecture	547
CORBA Basics	547
RMI-IIOP: Making RMI Compatible with CORBA	551
How to Turn an RMI Object into an RMI-IIOP Object	551
When to Use CORBA	553
Distributed File System Notifications: An Example CORBA System	554
Web Services	563
Random-Weather.org	564
Platform-Independent RPC	566
Summary	597

Chapter 12: Service Oriented Integration **599**

Service Oriented Architecture	599
Enabling Technology	600
Java Management Extensions	600
Why Is JMX Important?	601
The JMX Architecture	601
Creating and Managing a Standard MBean	602
JMX Management	606
Java Messaging Service	610
Why Is JMS Important?	610
Endpoints: Queues and Topics	610
Sending and Receiving Messages	611
System Integration Patterns	619
Processing Chain	619
Request-Reply	619
Split-Aggregate	621
Summary	623

Chapter 13: Java Security	625
Java Cryptography Architecture and Java Cryptography Extension (JCA/JCE)	625
JCA Design and Architecture	626
Java Cryptography Extension	656
Program Security Using JAAS	666
User Identification	667
Executing Code with Security Checks	668
Authorization	672
Summary	673
Chapter 14: Packaging and Deploying Your Java Applications	675
Examining Java Classpaths	675
Investigating the Endorsed Directory	680
Exploring Java Archives	681
Manipulating JAR Files	681
Examining the Basic Manifest File	684
Examining Applets and JARs	685
Signing JAR Files	686
Examining the JAR Index Option	690
Creating an Executable JAR	691
Analyzing Applets	691
Basic Anatomy of an Applet	692
Packaging an Applet for Execution	693
Examining Applet Security	694
Exploring Web Applications	694
Examining the WAR Directory Structure	695
Understanding the WAR Deployment Descriptor	696
Packaging Enterprise JavaBeans	698
Inspecting Enterprise Archives	699
The EAR Descriptor File	700
Deployment Scenario	700
Jumping into Java Web Start	702
Examining the TicTacToe Example	702
Summarizing Java Web Start	709
Using Ant with Web Archives	709
Installing Ant	709
Building Projects with Ant	710
Summary	713
Index	715

