

## ***INTRODUCTION***

---

**W**ith more than 600 Java books now in print, why would the computer industry need yet one more? *Java® Application Strategies for the AS/400* is written specifically for the AS/400 RPG programmer. For the most part, the myriad of available Java books presume a readership replete with C++ or at least C programming experience.

Also, many Java books include no discussion whatsoever about business programming. Written for AS/400 business programmers, IBM's Java Redbooks serve as an important reference resource; yet, they don't provide a straightforward discussion on strategies for Java application design and programming.

This book gives you real-world strategies for the development of AS/400 Java applications. *Java® Application Strategies for the AS/400* is divided into five sections:

- An Introduction to Java for RPG Programmers.
- Object-Oriented Design.
- Accessing Your AS/400.
- Object-Oriented Strategies for Mapping DB2/400 to Java.
- Graphical User Interface.

The first section, An Introduction to Java for RPG Programmers, acquaints you with Java from the perspective of an RPG programmer. This section of the book

introduces you to not just the structure of the Java language but also the intent of the language. Learning Java as just another programming language is easy; but if you develop Java applications using the same structured techniques that you've employed in RPG applications, those Java applications will fail. This inevitable failure arises from Java's intrinsic design as an object-oriented rather than a structured programming language. The benefits of Java can be achieved only through the use of proper object-oriented programming strategies.

The second section, Object-Oriented Design, provides seven chapters on object-oriented programming with Java, the first viable object-oriented language for business programming. Not just a fad, object-oriented programming is a proven methodology that has been widely used on most computing platforms for more than 10 years. Even the AS/400 has hosted object-oriented programming for a half-dozen years with IBM's VisualAge for C++ and VisualAge for Smalltalk. Finally, however, with the power and simplicity of Java, the AS/400 has an object-oriented language that doesn't require a computer science degree to understand. Section II of the book provides you with complete coverage of the Java language's implementation of object-oriented programming.

The third section, Accessing Your AS/400, details the use of the utilities that IBM Rochester delivers with its Java Toolbox for the AS/400. These utilities provide record-level and SQL access to DB2/400, legacy program interoperability, and data queue support. The curious thing about this section comes to light in the closing paragraphs, which point out that my own Java examples are poor illustrations of object-oriented programming. For ease of understanding, these examples were developed using structured techniques that RPG programmers will find familiar. But, as mentioned earlier, a successful Java application must be properly designed using object-oriented techniques. The fourth section of this book gives you the strategies you'll need to do that.

Section IV, Object-Oriented Strategies for Mapping DB2/400 to Java, provides instruction on how to develop Java classes that encapsulate access to business entities stored in DB2/400. This section also provides a framework of Java classes, the purpose of which is twofold. First, it furnishes a complete example of the development of object-oriented Java business classes. Second, it provides a frame

work of classes that you can use in the development of your own business classes. Both IBM and Inprise have object-to-relational utilities (*wizards* or *smartguides*, as they prefer to call them) that produce Java classes that map your DB2/400 data to Java. The resulting classes that are generated from these utilities function much the same as the frameworks presented in Section IV so the coverage in this section still applies, regardless of the object-to-relational tool used. The frameworks presented in this section and the Java classes generated from Inprise or IBM primarily differ in that Inprise's and IBM's classes use SQL for data access and mine use record-level access. Record-level access, as you know, is faster than SQL.

Users want a graphical user interface. They've wanted one since Windows first became popular. A GUI empowers users, puts them in the driver's seat. For years, they have worked with the user interface of the 5250 where the programmer had control. Java encompasses some very elegant GUI programming strategies; and Section V, Graphical User Interface, provides a tutorial on how to develop Java GUIs using those strategies. The first chapter within this section covers basic Java GUI programming using Java's package of GUI classes known as the Abstract Windowing Toolkit (AWT). The second chapter details the design of several Java GUIs that present DB2/400 data that is accessed from the example business classes developed in Section IV. One constant theme throughout this section is the separation of UI (graphical or otherwise) code from the code that maintains your business entities.

Ah, subfiles. What will replace AS/400 subfiles? The final chapter of this book provides a tutorial on the replacement paradigm for AS/400 subfiles—*grids* or, more specifically, the JTable grid component from Sun Microsystems' Java Foundation Classes (JFC). The chapter not only provides a tutorial on JFC's JTable, but it also introduces you to the JFC GUI classes for the development of advanced business GUIs. This last chapter contains a tutorial on how to use JFC's JTree component and JFC's JTabbedPane. JFC's JTree uses the visual metaphor of a tree such as is used with Windows Explorer. JFC's JTabbedPane component is used for the creation of tabbed dialogs. The GUI of robust business applications requires sophisticated components such as grids, trees, and tabbed dialogs; this section tells you how to develop those GUI applications using Sun's JFC.

## INTRODUCTION

---

Ultimately, by reading *Java® Application Strategies for the AS/400*, you will:

- Learn the structure and intent of the Java programming language.
- Gain an understanding of the power of object-oriented programming.
- Learn standard strategies for object-oriented design.
- Know how to access DB2/400.
- Have strategies for interoperating Java applications with your legacy applications.
- Have a strategy and a framework for the development of business classes.
- Know how to develop Java GUIs complete with a replacement paradigm for AS/400 subfiles.