



北京希望电子出版社  
www.bhp.com.cn

21 世纪新工具软件开发指南丛书

2

最新 Java 编程指南

[美] 弗·海斯布洛克 著



本光盘内容是：  
本版电子书

# Programming Guide for JAVA

英文版

47

Visual Age for Java version 3.5



北京希望电子出版社  
www.bhp.com.cn

21 世纪新工具软件开发指南丛书 (2)

最新 Java 编程指南

[美] 弗·海斯布洛克 著



本光盘内容是:  
本版电子书

# Programming Guide for JAVA

英文版

Visual Age for Java version 3.5

## 内 容 简 介

这是一本关于 Java 最新版编程指南专著。书中详述了 Java 应用程序、Java 小应用程序、小服务程序和 JSP 编程,以及 3 层应用程序设计概念,并指导用户使用关系数据库和 WebSphere 测试环境,以及帮助用户学会如何使用开发工具所提供的各种功能,诸如综合式调试、WebSphere 测试环境,以及 Web 站点资源的发表。

全书由 12 章和一个附录组成。各章内容分述如下。第 1 章环境概述;第 2 章组织代码;第 3 章迁移到 Java 2;第 4 章 ATM 工程;第 5 章创建小服务程序;第 6 章创建 JSP (Java 服务器网页);第 7 章创建图形用户界面 (GUI) 应用程序;第 8 章更改代码的型式;第 9 章 Web 应用程序的测试与调试;第 10 章如何使用关系数据库;第 11 章国际化;第 12 章如何运用 Web 应用程序。

本书既适用于 Java 和 Web 应用环境下从事软件开发的广大编程人员,又是高等院校师生教学、自学参考书,以及科研院所必备的馆藏读物。

本光盘内容包括本版电子书。

系 列 书: 21 世纪新工具软件开发指南丛书 (2) (英文版)  
书 名: 最新 Java 编程指南  
总 策 划: 北京希望电子出版社  
文本著作者: 弗·海斯布洛克  
责 任 编 辑: 谢建勋  
CD 制 作 者: 希望多媒体创作中心  
CD 测 试 者: 希望多媒体测试部  
出版、发行者: 北京希望电子出版社  
地 址: 北京中关村大街 26 号, 100080  
网址: [www.bhp.com.cn](http://www.bhp.com.cn) E-mail: [lwm@hope.com.cn](mailto:lwm@hope.com.cn)  
电话: 010-62562329, 62541992, 62637101, 62637102  
010-62633308, 62633309 (发行)  
010-62613322-215 (门市) 010-62629581 (编辑部)  
经 销: 各地新华书店、软件连锁店  
排 版: 希望图书输出中心 李毅  
CD 生 产 者: 北京中新联光盘有限责任公司  
文本印刷者: 北京广益印刷厂  
开本 / 规格: 787×1092 毫米 1/16 开本 20.75 印张 450 千字  
版次 / 印次: 2001 年 7 月第 1 版 2001 年 7 月第 1 次印刷  
印 数: 1-2000 册  
本 版 号: ISBN 7-900071-68-7/TP·67  
定 价: 52.00 元 (1CD, 含配套书)

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损,本社发行部负责调换。

# Contents

|                  |                                                      |          |
|------------------|------------------------------------------------------|----------|
| <b>CHAPTER 1</b> | <b>Introduction to the environment</b>               | <b>1</b> |
| 1.1              | VisualAge for Java product family .....              | 1        |
| 1.1.1            | VisualAge for Java Professional Edition .....        | 2        |
| 1.1.2            | VisualAge for Java Entry Professional Edition .....  | 4        |
| 1.1.3            | VisualAge for Java Enterprise Edition .....          | 4        |
| 1.1.4            | VisualAge for Java Entry Enterprise Edition .....    | 5        |
| 1.1.5            | Updates to VisualAge Java .....                      | 5        |
| 1.2              | Building your first applet .....                     | 5        |
| 1.2.1            | Let's get started! .....                             | 6        |
| 1.2.2            | SmartGuide .....                                     | 8        |
| 1.2.3            | The Workbench .....                                  | 10       |
| 1.2.4            | Modifying your applet .....                          | 11       |
| 1.2.5            | Creating an animated applet .....                    | 12       |
| 1.2.6            | Changing the properties of the applet .....          | 13       |
| 1.3              | Building your first application .....                | 14       |
| 1.3.1            | Running a program as an applet and application ..... | 16       |
| 1.3.2            | The VisualAge for Java Source View window .....      | 17       |
| 1.4              | The VisualAge for Java Scrapbook .....               | 18       |

|       |                                           |    |
|-------|-------------------------------------------|----|
| 1.4.1 | Using the Scrapbook .....                 | 18 |
| 1.4.2 | Scrapbook context .....                   | 20 |
| 1.4.3 | Correcting errors in the Scrapbook .....  | 20 |
| 1.4.4 | If your Scrapbook page remains busy ..... | 22 |
| 1.5   | Customizing VisualAge for Java .....      | 22 |
| 1.5.1 | Workbench Options .....                   | 23 |
| 1.6   | Building your first servlet .....         | 25 |

## **CHAPTER 2** Organizing your code 32

|       |                                                       |    |
|-------|-------------------------------------------------------|----|
| 2.1   | Projects in VisualAge for Java .....                  | 33 |
| 2.1.1 | Adding features .....                                 | 34 |
| 2.2   | Packages in VisualAge for Java .....                  | 34 |
| 2.3   | The Workbench .....                                   | 34 |
| 2.3.1 | The Workbench Projects page .....                     | 36 |
| 2.3.2 | The Workbench Packages view .....                     | 37 |
| 2.4   | types from other packages .....                       | 39 |
| 2.4.1 | Workbench Resources page .....                        | 40 |
| 2.4.2 | The Workbench Classes page .....                      | 41 |
| 2.4.3 | The Workbench Interfaces page .....                   | 42 |
| 2.4.4 | The Workbench All Problems page .....                 | 43 |
| 2.5   | Full source code edit .....                           | 45 |
| 2.6   | Code Assists .....                                    | 46 |
| 2.7   | Importing and exporting with VisualAge for Java ..... | 46 |
| 2.7.1 | Importing into VisualAge for Java .....               | 47 |
| 2.7.2 | Exporting from VisualAge for Java .....               | 49 |

## **CHAPTER 3** Migrating to Java2 54

|     |                                       |    |
|-----|---------------------------------------|----|
| 3.1 | The Fix/Migrate SmartGuide .....      | 55 |
| 3.2 | The repair process .....              | 57 |
| 3.3 | Migrating your servlet and JSPs ..... | 59 |

## **CHAPTER 4** Beginning the ATM Project 60

|       |                              |    |
|-------|------------------------------|----|
| 4.1   | Problem domain .....         | 61 |
| 4.2   | Building the ATM model ..... | 63 |
| 4.2.1 | Use cases .....              | 64 |

|       |                                                      |    |
|-------|------------------------------------------------------|----|
| 4.2.2 | ATM state diagrams .....                             | 65 |
| 4.2.3 | Analysis class diagram .....                         | 68 |
| 4.2.4 | Design class diagrams .....                          | 70 |
| 4.2.5 | Interaction diagram .....                            | 74 |
| 4.3   | Overall architecture .....                           | 76 |
| 4.3.1 | The big picture .....                                | 76 |
| 4.3.2 | GUI client .....                                     | 77 |
| 4.3.3 | Browser client .....                                 | 77 |
| 4.3.4 | Database access .....                                | 79 |
| 4.4   | Example implementation .....                         | 79 |
| 4.4.1 | Detailed steps implementing the first class .....    | 80 |
| 4.4.2 | Reusing existing method to create a new method ..... | 85 |
| 4.4.3 | PrimaryKey class hierarchy .....                     | 86 |
| 4.4.4 | Creating an inner class .....                        | 87 |
| 4.4.5 | Persistency based on HashMaps .....                  | 89 |
| 4.4.6 | Finder methods .....                                 | 89 |
| 4.4.7 | Implementations of the state diagram .....           | 90 |
| 4.4.8 | Test application .....                               | 90 |

## **CHAPTER 5** Creating servlets 92

|     |                                             |    |
|-----|---------------------------------------------|----|
| 5.1 | Overview of Java servlets .....             | 93 |
| 5.2 | The Java Servlet API .....                  | 97 |
| 5.3 | Building the ATM application servlets ..... | 99 |

## **CHAPTER 6** Creating JSPs 112

|       |                                    |     |
|-------|------------------------------------|-----|
| 6.1   | Java Server Pages .....            | 113 |
| 6.2   | How Java Server Pages work .....   | 114 |
| 6.3   | JSP interactions .....             | 115 |
| 6.4   | Invoking a JSP by URL .....        | 115 |
| 6.4.1 | Calling a servlet from a JSP ..... | 116 |
| 6.5   | JSP 0.91 and 1.0 .....             | 117 |
| 6.6   | Designing the JSP model .....      | 118 |
| 6.6.1 | Model-View-Controller (MVC) .....  | 118 |
| 6.6.2 | Servlet based modeling .....       | 118 |
| 6.7   | Building the ATM application ..... | 120 |
| 6.7.1 | JSP tags .....                     | 123 |

|       |                                                                        |     |
|-------|------------------------------------------------------------------------|-----|
| 7.1   | Abstract Windowing Toolkit and Java Foundation Classes refresher ..... | 129 |
| 7.2   | Visual Composition Editor .....                                        | 130 |
| 7.2.1 | The Beans Palette .....                                                | 131 |
| 7.2.2 | Modification of the Beans Palette .....                                | 133 |
| 7.2.3 | Visual Composition Editor toolbar .....                                | 134 |
| 7.2.4 | The free-form surface .....                                            | 134 |
| 7.3   | Working with beans in the Visual Composition Editor .....              | 135 |
| 7.3.1 | Adding beans .....                                                     | 135 |
| 7.3.2 | Customizing Beans .....                                                | 136 |
| 7.3.3 | Naming beans .....                                                     | 138 |
| 7.3.4 | Beans List .....                                                       | 139 |
| 7.3.5 | Factory and variable .....                                             | 139 |
| 7.4   | Visual Programming in action .....                                     | 139 |
| 7.4.1 | The ATM classes created .....                                          | 140 |
| 7.4.2 | Building the CardBean class .....                                      | 142 |
| 7.4.3 | Building the CardBeanHome class .....                                  | 146 |
| 7.4.4 | Building the BankAccountBean class .....                               | 147 |
| 7.4.5 | Building the ATM application .....                                     | 149 |
| 7.4.6 | Connections .....                                                      | 153 |
| 7.4.7 | Connection properties .....                                            | 155 |
| 7.4.8 | Creating connections .....                                             | 156 |

|       |                                                       |     |
|-------|-------------------------------------------------------|-----|
| 8.1   | introduction to versioning .....                      | 161 |
| 8.1.1 | Program elements .....                                | 162 |
| 8.2   | Workspace versus repository .....                     | 163 |
| 8.2.1 | The workspace is only a cache .....                   | 164 |
| 8.2.2 | Backup or restore the workspace .....                 | 165 |
| 8.2.3 | Clean workspace copy .....                            | 166 |
| 8.2.4 | Multiple workspaces on one repository .....           | 166 |
| 8.2.5 | Backup or restore the repository .....                | 167 |
| 8.2.6 | Workspace versus repository continued .....           | 167 |
| 8.3   | Version control .....                                 | 168 |
| 8.3.1 | Editions and versions .....                           | 169 |
| 8.3.2 | Consequences of versioning .....                      | 170 |
| 8.3.3 | How to version elements with VisualAge for Java ..... | 171 |
| 8.3.4 | Apply this to the ATM application .....               | 173 |
| 8.3.5 | Methods, a special case .....                         | 173 |
| 8.3.6 | Importance of versioning your code regularly .....    | 174 |
| 8.3.7 | Fields and inner classes .....                        | 175 |

|       |                                              |     |
|-------|----------------------------------------------|-----|
| 8.3.8 | Versioning resource files.....               | 175 |
| 8.4   | Using editions.....                          | 176 |
| 8.4.1 | Method edition tab .....                     | 176 |
| 8.4.2 | Comparison result window .....               | 179 |
| 8.4.3 | Merging compared elements .....              | 180 |
| 8.4.4 | Types edition tab.....                       | 182 |
| 8.4.5 | Packages edition tab.....                    | 183 |
| 8.4.6 | Projects edition tab.....                    | 184 |
| 8.4.7 | Replacing current edition .....              | 185 |
| 8.4.8 | External versioning systems.....             | 188 |
| 8.5   | Import and export effects.....               | 188 |
| 8.5.1 | Import and export with Java files.....       | 189 |
| 8.5.2 | Import and export with repository files..... | 189 |
| 8.6   | Repository Explorer.....                     | 193 |
| 8.6.1 | Purging and restoring elements .....         | 194 |
| 8.6.2 | Compacting a repository .....                | 196 |
| 8.6.3 | Go To tools .....                            | 197 |
| 8.6.4 | Solutions .....                              | 198 |

## CHAPTER 9 Testing and debugging the Web application 200

|       |                                                        |     |
|-------|--------------------------------------------------------|-----|
| 9.1   | VAJ Debugger.....                                      | 201 |
| 9.1.1 | The debugger .....                                     | 201 |
| 9.1.2 | The Debug Page.....                                    | 202 |
| 9.1.3 | The Breakpoints Page .....                             | 205 |
| 9.1.4 | External Debug .....                                   | 211 |
| 9.1.5 | Generating a Class Trace .....                         | 212 |
| 9.1.6 | Performance and the Class Trace option.....            | 213 |
| 9.1.7 | Inspectors .....                                       | 213 |
| 9.1.8 | The Inspector window.....                              | 214 |
| 9.2   | WebSphere Test Environment(WTE).....                   | 216 |
| 9.2.1 | Start the WebSphere Test Environment .....             | 217 |
| 9.3   | Testing JSPs under WebSphere Test Environment.....     | 220 |
| 9.3.1 | VisualAge for Java configuration for JSPs .....        | 220 |
| 9.3.2 | Running a simple JSP .....                             | 221 |
| 9.4   | Debugging servlets and JSPs .....                      | 222 |
| 9.4.1 | Debugging a servlet .....                              | 222 |
| 9.4.2 | JSP Execution Monitor .....                            | 224 |
| 9.4.3 | Debugging JSP generated source code.....               | 226 |
| 9.4.4 | Debugging JSP without importing .....                  | 226 |
| 9.5   | Persistent Name Server.....                            | 227 |
| 9.6   | WebSphere Test Environment—advanced configuration..... | 229 |
| 9.6.1 | Types of resources .....                               | 229 |
| 9.6.2 | Resource locations .....                               | 229 |

|       |                                                            |     |
|-------|------------------------------------------------------------|-----|
| 9.6.3 | The key configuration files .....                          | 230 |
| 9.7   | WebSphere Test Environment—multiple Web applications ..... | 236 |
| 9.7.1 | Configuring multiple Web applications .....                | 236 |
| 9.7.2 | Using the ServletEngineConfigDumper servlet .....          | 238 |

## CHAPTER 10 Using relational databases 240

|         |                                             |     |
|---------|---------------------------------------------|-----|
| 10.1    | JDBC 2.0 .....                              | 242 |
| 10.1.1  | DataSource versus DriverManager .....       | 244 |
| 10.2    | Queries and result sets .....               | 244 |
| 10.3    | Stored procedures .....                     | 244 |
| 10.4    | Updating the database.....                  | 245 |
| 10.5    | Using SQLJ inside Visual Age for Java.....  | 245 |
| 10.6    | Data access beans .....                     | 247 |
| 10.7    | Making the ATM persistent.....              | 249 |
| 10.7.1  | Prerequisites .....                         | 249 |
| 10.7.2  | Creating tables.....                        | 250 |
| 10.8    | Making the card class persistent .....      | 250 |
| 10.9    | Creating the Select beans.....              | 251 |
| 10.9.1  | Card Select bean.....                       | 251 |
| 10.9.2  | Card Select All bean.....                   | 259 |
| 10.9.3  | CardAcctSelect.....                         | 260 |
| 10.10   | Modify beans .....                          | 260 |
| 10.10.1 | Card Insert.....                            | 261 |
| 10.10.2 | Card Delete .....                           | 262 |
| 10.10.3 | Card Update .....                           | 263 |
| 10.10.4 | Card Visual Composition Editor View.....    | 264 |
| 10.10.5 | Modifying related methods .....             | 265 |
| 10.11   | Data Access Beans with an application ..... | 267 |

## CHAPTER 11 Internationalization 270

|        |                                                  |     |
|--------|--------------------------------------------------|-----|
| 11.1   | Java Internationalization Framework.....         | 271 |
| 11.1.1 | Locales .....                                    | 272 |
| 11.1.2 | Resource bundle.....                             | 274 |
| 11.2   | Internationalization in VisualAge for Java ..... | 276 |
| 11.3   | Building a language panel.....                   | 278 |
| 11.3.1 | LanguagePanel view .....                         | 279 |
| 11.3.2 | Creating the resource bundles .....              | 280 |
| 11.4   | Dynamically changing the locale.....             | 281 |

|        |                                                   |     |
|--------|---------------------------------------------------|-----|
| 11.4.1 | Loading resource bundles .....                    | 281 |
| 11.4.2 | Retrieving resources from resource bundles .....  | 282 |
| 11.4.3 | Finishing the LanguagePanel .....                 | 282 |
| 11.4.4 | Formatting dates and times .....                  | 286 |
| 11.5   | Other internationalization considerations .....   | 288 |
| 11.5.1 | Using predefined formats .....                    | 288 |
| 11.6   | Internationalization in the Web environment ..... | 293 |
| 11.7   | Character codes on the Web .....                  | 294 |

## **CHAPTER 12** Deploying the Web application 296

|        |                                              |     |
|--------|----------------------------------------------|-----|
| 12.1   | Before you start .....                       | 297 |
| 12.2   | Using WebSphere Application Server .....     | 299 |
| 12.2.1 | Deploying a Web application .....            | 299 |
| 12.2.2 | Planning for multiple Web applications ..... | 303 |
| 12.2.3 | Deploying a JSP .....                        | 304 |
| 12.3   | Deploying an application .....               | 304 |
| 12.3.1 | Include Referenced Types .....               | 305 |
| 12.4   | Deploying an applet .....                    | 306 |
| 12.4.1 | Web browsers .....                           | 306 |
| 12.4.2 | CLASSPATH or CODEBASE .....                  | 307 |
| 12.4.3 | Applet Tags .....                            | 307 |
| 12.4.4 | Deploying the ATMApplication applet .....    | 309 |
| 12.4.5 | Deploying supporting code .....              | 309 |

## **APPENDIX** JSP tag syntax 312

|                               |     |
|-------------------------------|-----|
| JSP tag syntax summary .....  | 313 |
| WebSphere specific tags ..... | 316 |

VisualAge for Java product is IBM's integrated, visual development environment for building Java applications, servlets, applets, and Java Beans. VisualAge for Java supports the complete cycle of Java program development. Using the true rapid application development (RAD) capability provided by VisualAge for Java, you can shorten the development life cycle of your applications and improve their time to market.

VisualAge for Java is a comprehensive, best-of-breed Java tool for creating e-business applications that target the IBM WebSphere software platform for e-business — the industry's most flexible and reliable e-business foundation for the rapid development and delivery of a brave new world of e-business applications. The ability to rapidly build, test, and deploy e-business applications sets VisualAge for Java apart from its competition.

In this chapter you will find a short description of the VisualAge for Java product family and an overview of VisualAge for Java Version 3.5. You will learn the basic terms that you need to understand to create your first program. Before you finish reading this chapter, you will have your first Java program up and running on the Web!

## 1.1 | VisualAge for Java product family

VisualAge for Java Version 3.5 is available in four editions:

- VisualAge for Java Professional Edition
- VisualAge for Java Entry Professional Edition
- VisualAge for Java Enterprise Edition
- VisualAge for Java Entry Enterprise Edition

This book covers the VisualAge for Java Version 3.5 Professional Edition and the VisualAge for Java Version 3.5 Entry Professional Edition, without describing the features of VisualAge for Java Enterprise Edition and VisualAge for Java Entry Enterprise Edition.

IBM also has several other offerings related to VisualAge for Java, including:

- VisualAge Developer Domain (VADD)

VisualAge Developer Domain (VADD) is Java developer's central access point for products, JavaBeans, tools, tech tips, demos and samples, product support, and product updates. In addition, VADD gives you access to an ever-expanding technical library of Java information, including newsletters, IBM Redbooks, technical articles, white papers, IBM Systems Journal articles, product documentation, FAQs, presentations, and educational opportunities. VADD is also a great place for you to exchange information with a worldwide community of Java developers through the forums and newsgroups. VADD offers you different access levels, depending on your needs.

You can learn more about VisualAge Developers Domain at:

[www.software.ibm.com/vadd](http://www.software.ibm.com/vadd)

- WebSphere Application Server

IBM WebSphere Application Server is an e-business application deployment environment built on open standards-based technology. It is the cornerstone of WebSphere application offerings and services. The Standard Edition lets you use Java servlets, Java Server Pages and XML to quickly transform static Web sites into vital sources of dynamic Web content. The Advanced Edition is a high-performance EJB server for implementing EJB components that incorporate business logic. The Enterprise Edition integrates EJB and CORBA components to build high-transaction, high-volume e-business applications.

### 1.1.1 VisualAge for Java Professional Edition

VisualAge for Java Professional Edition is an integrated visual environment that supports the complete cycle of Java program development. VisualAge for Java gives you everything you need to perform the development tasks. described below. The IDE includes:

- Incremental compiler

Changes to your code are compiled "on-the-fly" as you work with individual methods and class declarations. Errors in your code are immediately flagged so that they can be fixed while you are concentrating on that part of the code.

- Repository-based environment

All of the code in the development environment is stored in a *repository*. This

repository enables incremental compilation and provides for very powerful search capabilities. The code that you are working with is stored in a *workspace*. Version management is built into the repository, and versions or *editions* of code are automatically stored when you change any *program element* (method, class, package, or project) in your workspace.

VisualAge for Java Professional Edition is a single-user, repository-based environment. If you work as part of a development team, you may want to consider using VisualAge for Java Enterprise Edition.

- Project-based development

VisualAge for Java provides *projects*. The basic Java environments provide only the concept of a package to organize your work. In VisualAge for Java you organize your packages in projects.

- Source code editor

A full-featured syntax editor, which helps you write error-free source code.

- Advanced coding tools such as automatic formatting, automatic code completion, fix-on-save, and suggested corrections feature
- An integrated debugger
- A Visual Composition Editor, which enables you to develop your application visually
- A JavaBean creation tool to create 100% pure Java beans that you can use with the Visual Composition Editor

New powerful features that come with Version 3.5 of VisualAge for Java Professional Edition include:

- Full Java 2 SDK, Standard Edition, V1.2.2 support
  - JDK 1.2.2
  - Swing 1.1
- Fix/Migrate SmartGuide assists with Swing 1.0.3 -> Swing 1.1 Migration
- Full source code editing

The new option Open Source View provides you with another way to view entire source in editor. Code assist is available in source view. The file format/order is preserved. The editor-oriented programmers should like this!

- Improved inner class support
  - You can Browse/Edit inner classes/methods like normal classes/methods in IDE.
- Manage non-Java artifacts from Resources view

Non-Java artifacts are not stored in repository. They are managed based on date/time stamp when resources are released

## 4 Chapter1 | Introduction to the environment

- Enhanced code formatting
- Servlet SmartGuide generates servlets, JSP files, and prototype HTML
- All Problems page filtering  
Filter warnings/errors on All Problems page
- New WebSphere Test Environment Control Center  
You can Start/Stop Servlet Engine, Start/Stop Persistent Name Server (PNS), Set JSP Execution Monitor Settings and Define Data Sources

### 1.1.2 *VisualAge for Java Entry Professional Edition*

The VisualAge for Java Entry Professional Edition provides the same functions as VisualAge for Java Professional, with a limit of 750 Java types (classes and interfaces).

To download VisualAge for Java, Entry Professional Edition, Version 3.5 you must be a registered user of VisualAge Developer Domain and logged in. Registration is free.

VisualAge Developer Domain(VADD) Web site:

[www.software.ibm.com/vadd](http://www.software.ibm.com/vadd)

### 1.1.3 *VisualAge for Java Enterprise Edition*

VisualAge for Java Enterprise Edition is an enterprise-aware, Java application development environment for teams of Java developers. Use it to extend existing server data, transactions, and applications to e-business.

In addition to the functions in the Professional Edition, VisualAge for Java V3.5 Enterprise Edition supports:

- Updated Enterprise Access Builder (EAB) functionality that consolidates connectors at a JDK 1.2.2 level and positions customers for compliance to the emerging Java 2 Platform, Enterprise Edition JCX API.
- Enhanced Enterprise Access Builders (CICS TS, IMS, Encina, MQSeries, TXSeries, Host-On Demand, and SAP R/3) that provide more connectivity than any other Java IDE.
- Test client generation that speeds testing of server side Java code (EJB components). Also generates clients to Enterprise Access Builders connectors.
- Enterprise JavaBeans components. Generation of and complete support for the EJB specification, through wizards, persistent mapping tools, dependency management, and improved test and advanced deployment tools.
- XMI Toolkit for integration with the Rational Rose product, and other XMI-based UML modelling tools.

- Interface definition language (IDL) development environment and improved support for multiple object request brokers (ORBs), which can now be loaded into and unloaded from the development environment.

#### 1.1.4 VisualAge for Java Entry Enterprise Edition

The VisualAge for Java Entry Enterprise Edition provides the same functions as VisualAge for Java Enterprise Edition, with a limit of 750 Java types (classes and interfaces).

To download VisualAge for Java, Entry Enterprise Edition, Version 3.5, you must be registered user of VisualAge Developer Domain and logged in. Registration is free.

VisualAge Developer Domain(VADD) Web site:

[www.software.ibm.com/vadd](http://www.software.ibm.com/vadd)

#### 1.1.5 Updates to VisualAge Java

Updates to the different editions of VisualAge Java are provided at the VisualAge Developer Domain (VADD) Web site:

[www.software.ibm.com/vadd](http://www.software.ibm.com/vadd)

## 1.2 | Building your first applet

Now that you have an idea of the capabilities of VisualAge for Java, you can build your first applet.

Before starting, you should familiarize yourself with the terms in Table 1. If you are not familiar with any of the terms in Table 1 or you are new to the Java language itself, first read some of the related resources.

Table 1 Object-oriented terms

| Term      | Definition                                                                                                                                           |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Class     | A template for creating objects. A class defines the behavior and properties that are common to all objects of that class.                           |
| Interface | A specification of behavior that a class must provide if it implements the interface.                                                                |
| Object    | An instance of a class. An object shares the behavior of all objects of the same class, but each object can have a different state.                  |
| Applet    | A special kind of class introduced in Java. Its instances usually run in a Web browser such a Netscape Navigator. Contrast with <i>Application</i> . |
| JApplet   | If you want to use Swing components in the applet, use JApplet as the super class rather than Applet.                                                |

*Continued*

| Term               | Definition                                                                                                                                                                                                                   |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Swing Set          | Swing Set = JFC (Java Foundation Classes) JFCs are building blocks that are helpful in developing interfaces to Java applications. They allow Java applications to interact more completely with existing operating systems. |
| Application        | In Java programming, a self-contained, stand-alone Java program that includes a static main method. It does not require an applet viewer. Contrast with Applet.                                                              |
| Attribute or field | A data variable held by a class.                                                                                                                                                                                             |
| Access modifier    | In Java the access modifiers are public, private, protected, and default or package.                                                                                                                                         |
| Method or Message  | Objects communicate with each other by sending messages. When an object receives a message, a corresponding method, defined in the class definition, is invoked to perform the required task.                                |
| Package            | A collection of Java classes that typically serve a common purpose. This is Java's way of organizing classes into logical entities that are easier to maintain and understand than a huge set of classes at the same level.  |
| Server             | The computer that hosts the Web page that contains an applet. The .class files that make up the applet, and the .HTML files that reference the applet reside on the server.                                                  |
| Servlet            | Server-side programs that execute on and add function to Web servers. Java servlets allow for the creation of high-performance, cross-platform Web applications.                                                             |

### 1.2.1 Let's get started!

Before you go any further, you must have VisualAge for Java installed on your computer. Your first Java class is a simple applet that displays the text of your choice in the applet's window. You launch VisualAge for Java by double-clicking the **IBM VisualAge for Java** icon in the IBM VisualAge for Java folder or selecting **Start→ Programs→ IBM VisualAge for Java for Windows→ IBM VisualAge for Java**.

If this is the first time you have started VisualAge for Java, a dialog box will inform you that some features are being installed. Next, the Welcome to VisualAge dialog box (see Figure 1) is displayed. The Welcome to VisualAge dialog box is shown when you start the VisualAge for Java IDE, unless you deselect the checkbox at the bottom of the window. Click **Close** to close the dialog box.

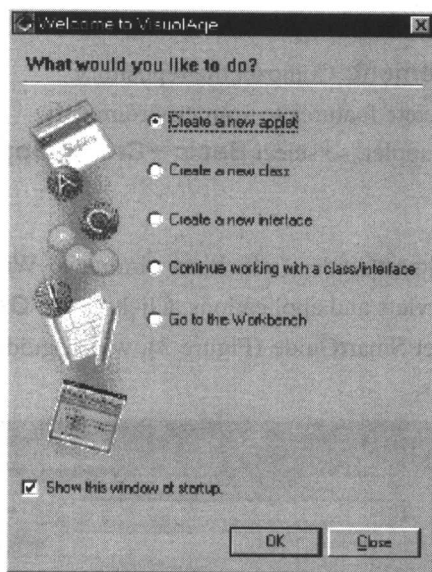


Figure 1. The VisualAge for Java Welcome dialog box

A different dialog, the Quick Start, can also be used to start. The Quick Start is available from the **Workbench**→ **File** menu (see Figure 2).

The Workbench window opens the first time you start VisualAge for Java. The Workbench is where you usually create and manipulate your classes. From the Workbench you can launch the Quick Start window. Open the Quick Start now, by selecting **File**→**Quick Start** from the menu bar.

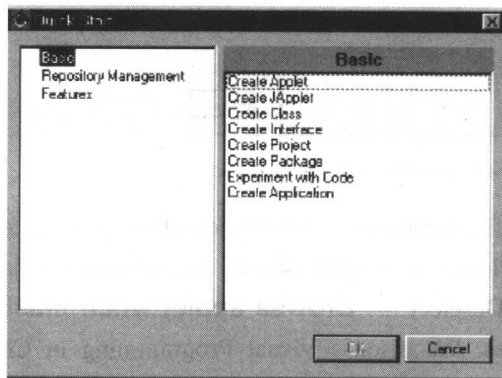


Figure 2. The Quick Start dialog box

Using the Quick Start dialog, you can select from three options:

- **Basic:** Create a new Applet, JApplet, Class, Interface, Project or Application, or