

Java

Guia do Programador

Peter Jandl Junior

Capítulo 1

Introdução

Este capítulo apresenta a plataforma Java, seu histórico resumido, suas principais características e aplicações. Também são indicados os recursos necessários para trabalhar com o Java.

1.1 A Linguagem Java

Java é uma plataforma de programação apresentada ao mercado em 1995 pela Sun Microsystems que ainda provoca entusiasmo em programadores, analistas e projetistas de software, pois é o resultado de um enorme trabalho de pesquisa científica e tecnológica. A plataforma Java é um ambiente completo de desenvolvimento e execução de programas que reúne um conjunto ímpar de facilidades: uma linguagem completamente orientada a objetos, robusta, de extrema portabilidade, permitindo a operação em rede (com destaque à internet) e a distribuição de aplicações e que incorpora diversas características voltadas à segurança.

Segundo sua especificação, o Java poderia ser assim caracterizado:

O Java é uma linguagem de programação de propósito geral, concorrente, baseada em classes e orientada a objetos. Foi projetada para ser simples o bastante para que a maioria dos programadores se torne fluente na linguagem. A linguagem Java tem relação com C e C++, mas é organizada diferentemente, com vários aspectos de C e C++ omitidos e algumas idéias de outras linguagens incluídas.

1.2 Breve histórico

Tudo começou, em 1991, com um pequeno grupo de projeto da Sun Microsystems denominado Green que pretendia criar uma nova geração de computadores portáteis inteligentes, capazes de se comunicar de muitas formas, ampliando suas potencialidades de uso. Para tanto decidiu-se criar também uma nova plataforma para o desenvolvimento destes equipamentos de modo que seu software pudesse ser portado para

os mais diferentes tipos de equipamentos. A primeira escolha de uma linguagem de programação para tal desenvolvimento foi C++, aproveitando suas características e a experiência dos integrantes do grupo no desenvolvimento de produtos. Mas mesmo o C++ não permitia realizar com facilidade tudo aquilo que o grupo visionava.

James Gosling, um dos líderes do projeto, decidiu pela criação de uma nova linguagem de programação que pudesse conter tudo aquilo considerado importante e que ainda fosse simples, portátil e fácil de programar. Surgiu então a linguagem interpretada Oak (carvalho em inglês), batizada assim em razão da existência de uma destas árvores em frente ao escritório de Gosling. Para dar suporte à linguagem também surgiram o Green OS e uma interface gráfica padronizada.

Após dois anos de trabalho, o grupo finaliza o Star7 (ou *7), um avançado PDA (Personal Digital Assistant), e, em 1993, surge a primeira grande oportunidade de aplicação desta solução da Sun numa concorrência pública da Time-Warner para desenvolvimento de uma tecnologia para TV a cabo interativa, vencida pela Silicon Graphics.

Devido a problemas de copyright, o Oak recebe o novo nome Java, mas continua sem uso definido até 1994, quando, estimulados pelo grande crescimento da internet, Patrick Naughton e Jonathan Payne desenvolveram o programa navegador WebRunner, capaz de efetuar o download e a execução de código Java via internet. Este navegador, sob o nome HotJava, e a linguagem Java foram então apresentados formalmente pela Sun em maio de 1995 no SunWorld'95, onde o interesse pela solução se mostrou explosivo. Logo no início de 1996 a Netscape Corp. lança a versão 2.0 de seu browser Navigator, o qual incorpora as capacidades de efetuar o download e realizar a execução de pequenas aplicações Java denominadas applets.

Numa iniciativa inédita, a Sun decide disponibilizar um conjunto de ferramentas de desenvolvimento Java gratuitamente para a comunidade de software, embora detenha todos os direitos relativos à linguagem e às ferramentas de sua autoria. Surge assim, em meados de 1996, o JDK 1.02 (Java Development Kit). As plataformas inicialmente atendidas foram: Sun Solaris e Microsoft Windows 95/NT. Uma vez que a especificação da linguagem e do ambiente de execução também foram colocados publicamente disponíveis, progressivamente foram aparecendo kits para outras plataformas, tais como IBM OS/2, Linux e Apple Mac.

Um ano após seu anúncio, ocorre o primeiro grande evento da linguagem Java, o JavaOne Conference, que, desde então, vem sendo usado para apresentar as novas características, casos de sucesso, produtos e tecnologias associadas. Foi assim que se iniciou a história de sucesso do Java.

Atualmente a plataforma está organizada em três segmentos principais:

- **JavaME (Java Micro Edition)** – Destinado a pequenos dispositivos computacionais móveis, tais como celulares, PDAs e set-top boxes. É composto de máquinas virtuais otimizadas para ambientes mais restritos, especificações de funcionalidades e uma API mais compacta.
- **JavaSE (Java Standard Edition)** – Integra os elementos padrão da plataforma e permite o desenvolvimento de aplicações de pequeno e médio porte. Inclui todas as APIs consideradas de base, além da máquina virtual padrão.
- **JavaEE (Java Enterprise Edition)** – Voltada para o desenvolvimento de aplicações corporativas complexas. Adiciona APIs específicas aos elementos-padrão da plataforma.

Desde o lançamento do Java, ocorreram poucas mudanças na linguagem (Tabela 1.1), embora sua API tenha sido largamente ampliada. A partir da versão 5, cujo projeto foi denominado Tiger, a plataforma recebeu nova numeração¹, sendo então chamada de Java 5.0, e sensíveis adições na linguagem e API. A última versão, Java 6.0, conhecida como Mustang, também incorpora diversas melhorias na API. Nestas versões o kit básico de desenvolvimento de software é denominado simplesmente JDK (Java Standard Edition Development Kit).

A próxima versão do Java, 7.0, denominada Dolphin, já está em desenvolvimento, indicando as perspectivas de longo prazo desta tecnologia e também seu sucesso.

Tabela 1.1 – Histórico das versões da plataforma Java

Ano	Versão	Versão Interna	Novas características da linguagem Java
1996	1.0	1.0	Versão inicial
1997	1.1	1.1	Classes internas
1998	1.2	1.2	Declaração strictfp. Compilação JIT
2000	1.3	1.3	Tecnologia HotSpot para JVM
2002	1.4	1.4	Diretiva assert
2004	5.0	1.5	Autoboxing. Enumerações. Genéricos. Metadata (Anotações)
2006	6.0	1.6	Suporte para scripting

1.3 Características principais

A linguagem Java exibe importantes características que, em conjunto, diferenciam-na de outras linguagens de programação. Dentre as características citadas de forma quase unânime por vários autores, são destacados alguns pontos-chave: orientada a objetos, independente de plataforma, sem ponteiros, performance, segurança e multithreaded.

¹ Segundo a Sun, o novo esquema de numeração vem refletir o nível de maturidade, estabilidade, escalabilidade e segurança da plataforma Java.

- **Orientada a Objetos** – Java é uma linguagem puramente orientada a objetos, pois, com exceção de seus tipos primitivos de dados, tudo em Java são classes ou instâncias de classes. Java atende a todos os requisitos necessários para que uma linguagem seja considerada orientada a objetos (que resumidamente são: oferecer mecanismos de abstração, encapsulamento e hereditariedade).
- **Independente de Plataforma** – Java é uma linguagem independente de plataforma, ou seja, os programas Java não são compilados para uma plataforma de hardware específica, mas sim para uma forma intermediária de código destinada à JVM (Java Virtual Machine). O formato intermediário é chamado de bytecodes, uma espécie de linguagem de máquina da JVM que utiliza instruções e tipos primitivos de tamanho fixo, ordenação big-endian e uma biblioteca de classes padronizada. A máquina virtual Java é na verdade um interpretador de bytecodes para a plataforma na qual é executada. Em razão da possibilidade de se implementar uma JVM para qualquer plataforma, é viável que um mesmo programa Java seja executado em qualquer destas arquiteturas.
- **Sem Ponteiros** – Java não possui ponteiros, i.e., não permite a manipulação direta de endereços de memória nem exige que os objetos criados sejam explicitamente destruídos, livrando os programadores de uma tarefa complexa. Toda a manipulação de variáveis e objetos se dá por meio de referências. Além disso, a JVM possui um mecanismo de gerenciamento de memória, conhecido como automatic garbage collector, o qual recupera a memória alocada para objetos não mais referenciados pelo programa.
- **Performance** – Java foi projetada para ser compacta, independente de plataforma, e para utilização em rede, o que levou à decisão de ser interpretada com o emprego do esquema de bytecodes. Nas primeiras versões, o Java oferecia performance apenas razoável devido ao mecanismo de interpretação. Esta limitação foi superada pela incorporação de um compilador JIT (Just In Time) na JVM capaz de converter os bytecodes em código nativo durante a carga do programa, possibilitando uma melhora significativa na performance dos programas Java, equiparando-os ao desempenho obtido com programas nativos. Esta tecnologia deu origem à família de JVMs HotSpot da Sun.
- **Segurança** – Considerando a possibilidade de que aplicações sejam obtidas por meio de uma rede, a linguagem Java possui mecanismos de segurança que podem, no caso de applets, evitar, por exemplo, qualquer operação no sistema de arquivos da máquina-alvo, minimizando problemas de segurança. Tal mecanismo é flexível o suficiente para determinar se uma applet é considerada segura, especificando nesta situação diferentes níveis de acesso ao sistema-alvo.

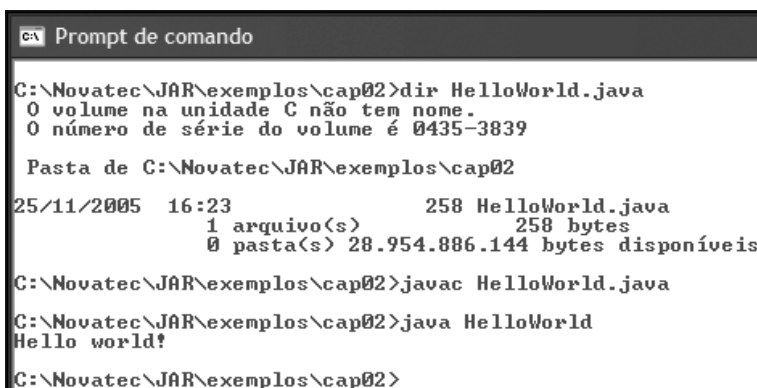
- **Multithreaded** – Java oferece recursos para o desenvolvimento de aplicações capazes de executar múltiplas rotinas concorrentemente, possibilitando inclusive sua sincronização. Cada uma destas rotinas é como um fluxo de execução independente, usualmente denominada thread. As threads são um importante recurso de programação para construção de aplicações mais sofisticadas.

Além de possuir as características comentadas, Java é também uma linguagem bastante robusta que incentiva o controle detalhado de erros. Oferece ainda: tipos inteiros e ponto flutuante bem como aritmética compatíveis com as especificações IEEE; suporte para caracteres Unicode; mecanismos de reflexão (determinação em tempo de execução dos tipos dos objetos em uso e de outras informações); sendo extensível dinamicamente, além de ser naturalmente voltada para o desenvolvimento de aplicações em rede ou distribuídas.

Mesmo com tudo isso, o Java ainda é uma linguagem sintática e estruturalmente simples, o que a torna uma linguagem de programação única. Exatamente por isso tornou-se a linguagem de programação orientada a objetos mais utilizada no mundo.

1.4 Ambiente Java

Java é uma linguagem independente de plataforma porque seus programas são compilados em um formato independente denominado bytecodes, os quais são armazenados em arquivos de classe `.class`. Os bytecodes são instruções de tamanho fixo que constituem a linguagem especificada para a JVM. Em cada combinação específica de hardware e sistema operacional deve existir uma JVM apropriada, capaz de interpretar os bytecodes ou de transformá-los em código nativo que possa ser executado pelo processador do sistema. Em qualquer caso a JVM utiliza os serviços oferecidos pelo sistema operacional em uso. Assim o ambiente Java é composto pela JVM, pela API Java e pelas classes da aplicação, como mostra a Figura 1.1.



```
C:\> Prompt de comando
C:\Novatec\JAR\exemplos\cap02>dir HelloWorld.java
O volume na unidade C não tem nome.
O número de série do volume é 0435-3839

Pasta de C:\Novatec\JAR\exemplos\cap02
25/11/2005  16:23                258 HelloWorld.java
                1 arquivo(s)                258 bytes
                0 pasta(s) 28.954.886.144 bytes disponíveis

C:\Novatec\JAR\exemplos\cap02>javac HelloWorld.java
C:\Novatec\JAR\exemplos\cap02>java HelloWorld
Hello world!
C:\Novatec\JAR\exemplos\cap02>
```

Figura 1.1 – Ambiente Java.

O desenvolvimento de programas em Java requer um editor que permita salvar o programa fonte como arquivos de extensão `.java`. Um compilador Java deve ser usado para transformar o programa fonte em bytecodes, salvos em arquivos de classe de extensão `.class`. Para executar uma aplicação Java é necessário uma JVM que interpretará ou conduzirá a execução dos arquivos de classe (Figura 1.2), usando direta ou indiretamente o código nativo do sistema.

Um ambiente Java mínimo é aquele que permite apenas a execução de aplicações Java, o que é possível com o uso do Sun JavaRE (Java Runtime Environment), cujo download gratuito pode ser realizado no endereço: <http://java.sun.com/javase/downloads/index.jsp>.

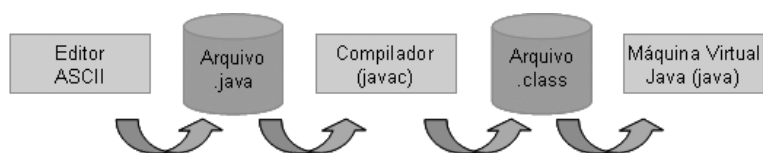


Figura 1.2 – Código-fonte, bytecodes e código nativo.

1.5 Recursos necessários

Para se construir aplicações Java é necessário um ambiente de desenvolvimento. O Sun JDK (Java Development Kit) é considerado padrão e inclui um conjunto útil de ferramentas de desenvolvimento, além do próprio JavaRE². Conhecido apenas como JDK, seu download gratuito pode ser feito em: <http://java.sun.com/javase/downloads/index.jsp>.

Além do JDK, é conveniente dispor de sua documentação, a qual inclui informação sobre suas ferramentas e sobre a API incluída. O download da documentação pode ser feito a partir do mesmo site de obtenção do JDK. Também será necessário um editor de texto simples, tal como o Notepad (bloco de notas) dos sistemas Windows ou Vi dos sistemas Unix e Linux³.

1.5.1 Instalando e configurando o JDK

A instalação do JDK é simples e pode ser feita seguindo as instruções do programa instalador, mas para que o JDK funcione corretamente é necessário fazer alguns ajustes no sistema. O primeiro ajuste é criar uma variável de ambiente denominada `CLASSPATH` e cujo conteúdo inicial seja `.` (ponto, caractere que indica o diretório atual). O segundo

² Recomenda-se o uso do JDK fornecido pela Sun, preferencialmente em versão igual ou superior a 5.0 de modo que todos os exemplos contidos neste material possam ser utilizados integralmente.

³ É bastante conveniente o uso de editores de programas capazes de efetuar o destaque de sintaxe do Java, tais como o JEdit, TextPad ou JCreator. Outra opção ainda mais interessante é o uso de um IDE (Integrated Development Environment) apropriado para Java, como o Eclipse ou NetBeans.

ajuste é editar a variável de ambiente `PATH`, que geralmente já existe, e adicionar o diretório onde estão as ferramentas do JDK instaladas (na plataforma Windows o usual é `C:\Arquivos de Programas\Java\jdk1.6.0\bin`).

Num sistema que utiliza o Windows XP, basta abrir o Explorer, localizar o item **Meu Computador**, com o botão direito exibir o menu de contexto e selecionar a opção **Propriedades**. Na janela que aparece deve ser selecionada a guia **Avançado**, e, nesta, o botão **Variáveis de ambiente** para a criação e edição das variáveis de usuário necessárias. A configuração já estará em vigor após o fechamento das janelas abertas.

Consulte a documentação do sistema operacional em uso para verificar como proceder com os ajustes necessários.

1.6 Exercícios de revisão

1. Como se organiza atualmente a plataforma Java?
2. Por que os programas Java são considerados independentes de plataforma?
3. Descreva o ciclo básico de desenvolvimento de uma aplicação Java.
4. O que são o JavaRE e o JDK?