

Módulo 1

Introdução à Programação I



Lição 2

Histórico de Java

Versão 1.0 - Jan/2007

Autor

Florence Tiu Balagtas

Equipe

Joyce Avestro
 Florence Balagtas
 Rommel Feria
 Reginald Hutcherson
 Rebecca Ong
 John Paul Petines
 Sang Shin
 Raghavan Srinivas
 Matthew Thompson

Necessidades para os Exercícios**Sistemas Operacionais Suportados****NetBeans IDE 5.5** para os seguintes sistemas operacionais:

- Microsoft Windows XP Professional SP2 ou superior
- Mac OS X 10.4.5 ou superior
- Red Hat Fedora Core 3
- Solaris™ 10 Operating System (SPARC® e x86/x64 Platform Edition)

NetBeans Enterprise Pack, poderá ser executado nas seguintes plataformas:

- Microsoft Windows 2000 Professional SP4
- Solaris™ 8 OS (SPARC e x86/x64 Platform Edition) e Solaris 9 OS (SPARC e x86/x64 Platform Edition)
- Várias outras distribuições Linux

Configuração Mínima de Hardware**Nota:** IDE NetBeans com resolução de tela em 1024x768 pixel

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	500 MHz Intel Pentium III workstation ou equivalente	512 MB	850 MB
Linux	500 MHz Intel Pentium III workstation ou equivalente	512 MB	450 MB
Solaris OS (SPARC)	UltraSPARC II 450 MHz	512 MB	450 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Série 1.8 GHz	512 MB	450 MB
Mac OS X	PowerPC G4	512 MB	450 MB

Configuração Recomendada de Hardware

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	1 GB
Linux	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	850 MB
Solaris OS (SPARC)	UltraSPARC IIIi 1 GHz	1 GB	850 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Series 1.8 GHz	1 GB	850 MB
Mac OS X	PowerPC G5	1 GB	850 MB

Requerimentos de Software

NetBeans Enterprise Pack 5.5 executando sobre Java 2 Platform Standard Edition Development Kit 5.0 ou superior (JDK 5.0, versão 1.5.0_01 ou superior), contemplando a Java Runtime Environment, ferramentas de desenvolvimento para compilar, depurar, e executar aplicações escritas em linguagem Java. Sun Java System Application Server Platform Edition 9.

- Para **Solaris**, **Windows**, e **Linux**, os arquivos da JDK podem ser obtidos para sua plataforma em <http://java.sun.com/j2se/1.5.0/download.html>
- Para **Mac OS X**, Java 2 Platform Standard Edition (J2SE) 5.0 Release 4, pode ser obtida diretamente da Apple's Developer Connection, no endereço: <http://developer.apple.com/java> (é necessário registrar o download da JDK).

Para mais informações: <http://www.netbeans.org/community/releases/55/relnotes.html>

Colaboradores que auxiliaram no processo de tradução e revisão

Alexandre Mori	Hugo Leonardo Malheiros Ferreira	Mauro Regis de Sousa Lima
Alexis da Rocha Silva	Ivan Nascimento Fonseca	Namor de Sá e Silva
Aline Sabbatini da Silva Alves	Jacqueline Susann Barbosa	Néres Chaves Rebouças
Allan Wojcik da Silva	Jader de Carvalho Belarmino	Nolyanne Peixoto Brasil Vieira
André Luiz Moreira	João Aurélio Telles da Rocha	Paulo Afonso Corrêa
Andro Márcio Correa Louredo	João Paulo Cirino Silva de Novais	Paulo José Lemos Costa
Antonie de Assis Lima	João Vianney Barrozo Costa	Paulo Oliveira Sampaio Reis
Antonio Jose R. Alves Ramos	José Augusto Martins Nieviadonski	Pedro Antonio Pereira Miranda
Aurélio Soares Neto	José Leonardo Borges de Melo	Pedro Henrique Pereira de Andrade
Bruno da Silva Bonfim	José Ricardo Carneiro	Renato Alves Félix
Bruno dos Santos Miranda	Kleberth Bezerra G. dos Santos	Renato Barbosa da Silva
Bruno Ferreira Rodrigues	Lafaiete de Sá Guimarães	Reydersen Magela dos Reis
Carlos Alberto Vitorino de Almeida	Leandro Silva de Moraes	Ricardo Ferreira Rodrigues
Carlos Alexandre de Sene	Leonardo Leopoldo do Nascimento	Ricardo Ulrich Bomfim
Carlos André Noronha de Sousa	Leonardo Pereira dos Santos	Robson de Oliveira Cunha
Carlos Eduardo Veras Neves	Leonardo Rangel de Melo Filardi	Rodrigo Pereira Machado
Cleber Ferreira de Sousa	Lucas Mauricio Castro e Martins	Rodrigo Rosa Miranda Corrêa
Cleyton Artur Soares Urani	Luciana Rocha de Oliveira	Rodrigo Vaez
Cristiano Borges Ferreira	Luís Carlos André	Ronie Dotzlaw
Cristiano de Siqueira Pires	Luís Octávio Jorge V. Lima	Rosely Moreira de Jesus
Derlon Vandri Aliendres	Luiz Fernandes de Oliveira Junior	Seire Pareja
Fabiano Eduardo de Oliveira	Luiz Victor de Andrade Lima	Sergio Pomerancblum
Fábio Bombonato	Manoel Cotts de Queiroz	Silvio Sznifer
Fernando Antonio Mota Trinta	Marcello Sandi Pinheiro	Suzana da Costa Oliveira
Flávio Alves Gomes	Marcelo Ortolan Pazzetto	Tásio Vasconcelos da Silveira
Francisco das Chagas	Marco Aurélio Martins Bessa	Thiago Magela Rodrigues Dias
Francisco Marcio da Silva	Marcos Vinicius de Toledo	Tiago Gimenez Ribeiro
Gilson Moreno Costa	Maria Carolina Ferreira da Silva	Vanderlei Carvalho Rodrigues Pinto
Givailson de Souza Neves	Massimiliano Girolodi	Vanessa dos Santos Almeida
Gustavo Henrique Castellano	Mauricio Azevedo Gamarra	Vastí Mendes da Silva Rocha
Hebert Julio Gonçalves de Paula	Mauricio da Silva Marinho	Wagner Eliezer Roncoletta
Heraldo Conceição Domingues	Mauro Cardoso Mortoni	

Auxiliadores especiais

Revisão Geral do texto para os seguintes Países:

- **Brasil** – Tiago Flach
- **Guiné Bissau** – Alfredo Cá, Bunene Sisse e Buon Olossato Quebi – ONG Asas de Socorro

Coordenação do DFJUG

- **Daniel deOliveira** – JUGLeader responsável pelos acordos de parcerias
- **Luci Campos** - Idealizadora do DFJUG responsável pelo apoio social
- **Fernando Anselmo** - Coordenador responsável pelo processo de tradução e revisão, disponibilização dos materiais e inserção de novos módulos
- **Regina Mariani** - Coordenadora responsável pela parte jurídica
- **Rodrigo Nunes** - Coordenador responsável pela parte multimídia
- **Sérgio Gomes Veloso** - Coordenador responsável pelo ambiente JEDI™ (Moodle)

Agradecimento Especial

John Paul Petines – Criador da Iniciativa JEDI™

Rommel Faria – Criador da Iniciativa JEDI™

1. Objetivos

Nesta lição iremos discutir um pouco da história de Java e o que é a tecnologia Java. Também iremos discutir as fases de um programa Java.

Ao final desta lição, o estudante será capaz de:

- Descrever as características da tecnologia Java como a JVM - Máquina Virtual Java, Garbage Collection e segurança do código;
- Descrever as diferentes fases de um programa Java.

2. Explorando o Java

2.1. Um pouco da história

Java foi criado em 1991 por James Gosling da Sun Microsystems. Inicialmente chamada OAK (Carvalho), em homenagem à uma árvore de janela do Gosling, seu nome foi mudado para Java devido a existência de uma linguagem com o nome OAK.



Figura 1: James Gosling criador do Java

A motivação original do Java era a necessidade de uma linguagem independente de plataforma que podia ser utilizada em vários produtos eletrônicos, tais como torradeiras e refrigeradores. Um dos primeiros projetos desenvolvidos utilizando Java era um controle remoto pessoal chamado *7 (Star Seven).



Figura 2: Star Seven

Ao mesmo tempo, a World Wide Web e a Internet foram ganhando popularidade. Gosling achava que a linguagem Java poderia ser usada para programação da Internet.

2.2. O que é a tecnologia Java?

2.2.1. Uma linguagem de programação

Como linguagem de programação, Java pode ser utilizado para criar todos os tipos de aplicações existentes, de programas de Inteligência Artificial para Robôs até programas para aparelhos celulares.

2.2.2. Um ambiente de desenvolvimento

Como ambiente de desenvolvimento, a tecnologia Java fornece um grande conjunto de ferramentas: um compilador, um interpretador, um gerador de documentação, ferramenta de empacotamento de classes de arquivos e outros.

2.2.3. Um ambiente de aplicação

Aplicações de tecnologia Java são tipicamente programas de propósito geral que executam sobre uma máquina onde o Java Runtime Environment é instalado.

2.2.4. Um ambiente de distribuição

Há dois ambientes de distribuição principais: Primeiro, o **JRE**, fornecido através do Java 2 Software Development Kit (SDK), contém um conjunto completo de arquivos de classes para todos pacotes de tecnologia Java. Outro ambiente de distribuição é o **navegador web**, ou seja, o **browser**. Os navegadores web atuais fornecem interpretação à tecnologia e ambiente Java em tempo de execução.

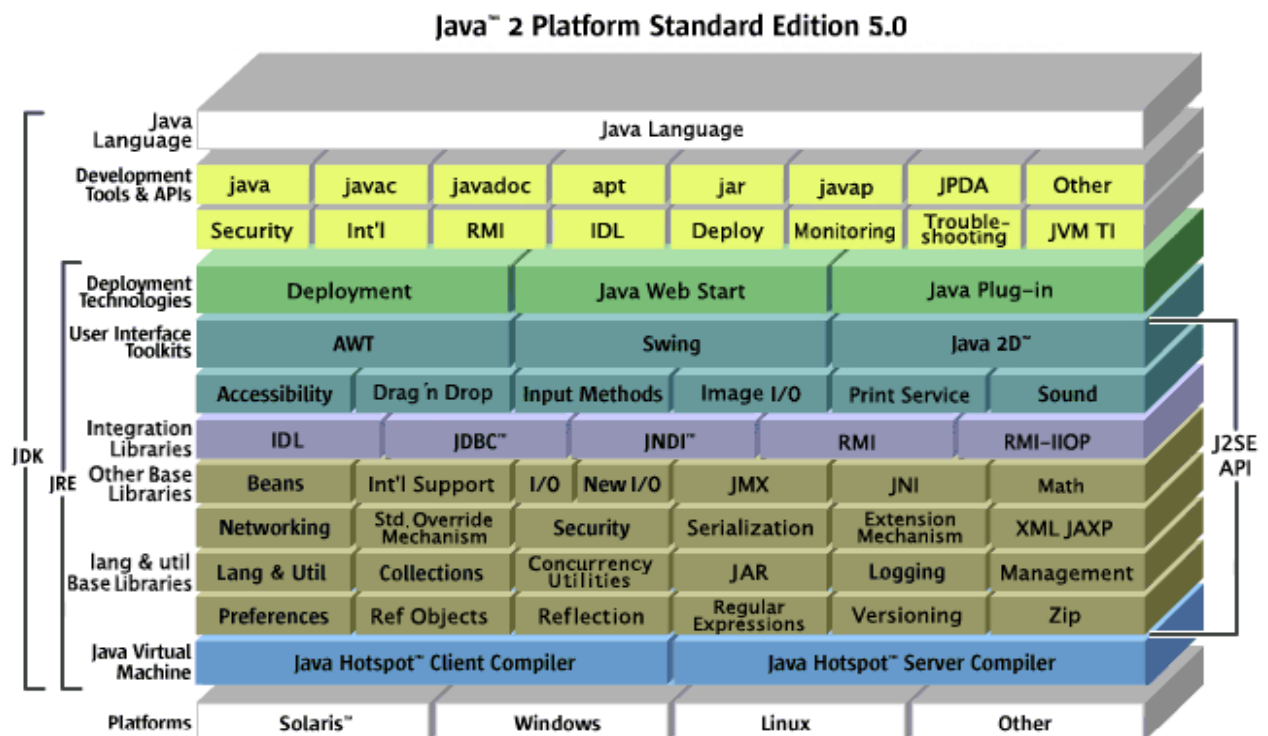


Figura 3: JDK e JRE

2.3. Algumas características do Java

2.3.1. Máquina Virtual Java

A **Máquina Virtual Java** é uma máquina imaginária que é implementada através de um software emulador em uma máquina real. A JVM provê especificações de plataforma de hardware na qual compila-se todo código de tecnologia Java. Essas especificações permitem que o software Java seja uma plataforma independente pois a compilação é feita por uma máquina genérica conhecida como JVM.

O **bytecode** é uma linguagem de máquina especial que pode ser entendida pela **Máquina Virtual Java (JVM)**. O bytecode é independente de qualquer hardware de computador particular. Assim, qualquer computador com o interpretador Java pode executar um programa Java compilado, não importando em que tipo de computador o programa foi compilado.

2.3.2. Garbage Collection

Muitas linguagens de programação permitem ao programador alocar memória durante o tempo de execução. Entretanto, após utilizar a memória alocada, deve existir uma maneira para desalocar o bloco de memória de forma que os demais programas a utilizem novamente. Em C, C++ e outras linguagens o programador é o responsável por isso. Isso, às vezes, pode ser difícil já que instâncias podem ser esquecidas de serem desalocadas da memória pelos programadores e resultar no que chamamos de escapes da memória.

Em Java, o programador não possui a obrigação de retirar uma variável criada das áreas de memória, isto é feito por uma parte da JVM específica que chamamos de **Garbage Collection**. O **Garbage Collection** é o grande responsável pela liberação automática do espaço em memória. Isso acontece automaticamente durante o tempo de vida do programa Java.

2.3.3. Segurança do Código

Segurança do Código é alcançada em Java através da implementação da **Java Runtime Environment (JRE)**. A JRE roda códigos compilados para a JVM e executa o carregamento de classes (através do **Class Loader**), verificação de código (através do verificador de **bytecode**) e finalmente o código executável.

O **Class Loader** é responsável por carregar todas as classes necessárias ao programa Java. Isso adiciona segurança através da separação do **namespace** entre as classes do sistema de arquivos local e aquelas que são importadas pela rede. Isso limita qualquer ação de programas que podem causar danos, pois as classes locais são carregadas primeiro. Depois de carregar todas as classes, a quantidade de memória que o executável irá ocupar é determinada. Isto acrescenta, novamente, uma proteção ao acesso não autorizado de áreas restritas ao código pois a quantidade de memória ocupada é determinada em tempo de execução.

Após carregar as classes e definir a quantidade de memória, o **verificador de bytecode** verifica o formato dos fragmentos de código e pesquisa nestes fragmentos por códigos ilegais que possam violar o direito de acesso aos objetos.

Depois que tudo isso tiver sido feito, o código é finalmente executado.

2.4. Fases do Programa Java

A figura seguinte descreve o processo de compilação e execução de um programa Java.

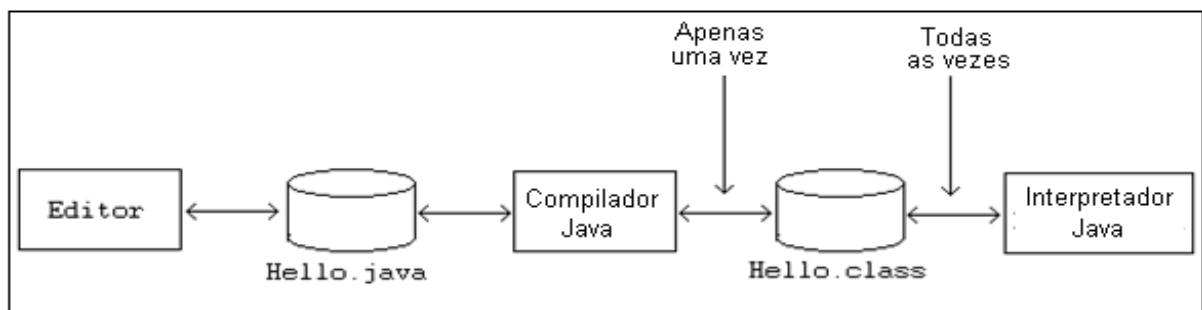


Figura 1: Fases de um Programa Java.

O primeiro passo para a criação de um programa Java é escrever os programas em um editor de texto. Exemplos de editores de texto que podem ser utilizados: bloco de notas, vi, emacs, etc. Esses arquivos são armazenados no disco rígido com a extensão **.java**.

Após o programa Java ter sido criado e salvo, compile o programa utilizando o Compilador Java. A saída desse processo é um arquivo de **bytecode** com extensão `.class`.

O arquivo `.class` é então lido pelo Interpretador Java que converte os bytecodes em linguagem de máquina do computador que se está usando.

Tarefa	Ferramenta utilizada	Saída
Escrever o programa	Qualquer editor de texto	Arquivo com extensão <code>.java</code>
Compilar o programa	Compilador Java	Arquivo com extensão <code>.class</code> (Java bytecode)
Executar o programa	Interpretador Java	Saída do programa

Tabela 1: Resumo das fases de um programa Java.

Parceiros que tornaram JEDI™ possível



Instituto CTS

Patrocinador do DFJUG.

Sun Microsystems

Fornecimento de servidor de dados para o armazenamento dos vídeo-aulas.

Java Research and Development Center da Universidade das Filipinas

Criador da Iniciativa JEDI™.

DFJUG

Detentor dos direitos do JEDI™ nos países de língua portuguesa.

Banco do Brasil

Disponibilização de seus *telecentros* para abrigar e difundir a Iniciativa JEDI™.

Politec

Suporte e apoio financeiro e logístico a todo o processo.

Borland

Apoio internacional para que possamos alcançar os outros países de língua portuguesa.

Instituto Gaudium/CNBB

Fornecimento da sua infra-estrutura de hardware de seus servidores para que os milhares de alunos possam acessar o material do curso simultaneamente.