

**CENTRO UNIVERSITÁRIO METROPOLITANO DE
SÃO PAULO
BACHAREL EM SISTEMAS DE INFORMAÇÃO**

NOME: *Ailton Sousa*

NOME: *Marcio Fabiano*

NOME: *Luiz Ricardo*

LINGUAGEM DE PROGRAMAÇÃO JAVA



**GUARULHOS
2012**

Conteúdo

Java	5
História	5
Padronização	6
Características	6
Licença.....	7
Exemplos de código	7
<i>Método main</i>	7
<i>Classes</i>	7
<i>Classes internas</i>	8
<i>Interfaces</i>	8
<i>Objetos anônimos</i>	9
Ferramentas	10
Frameworks	10
Ambientes de desenvolvimento	10
IDEs completas (recomendado para programadores profissionais)	10
Outras IDEs (menos populares).....	11
Extensões em Java:	11
Certificações	12
Comunidade	13
Custo de um programador JAVA	13
Oracle acquire Sun.....	13
Conclusão	14
Bibliografia	14

Java

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por **James Gosling**, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.

História

Em 1991, na Sun Microsystems, foi iniciado o Green Project, o berço do Java, uma linguagem de programação orientada a objetos. Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a “próxima onda” do mundo digital. Eles acreditavam que, em algum tempo, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados pelas pessoas no seu dia-a-dia.

Para provar a viabilidade desta ideia, 13 pessoas trabalharam arduamente durante 18 meses. No verão de 1992 eles emergiram de um escritório de Sand Hill Road, no Menlo Park, com uma demonstração funcional da ideia inicial. O protótipo se chamava *7 (lê-se “StarSeven”), um controle remoto com uma interface gráfica touchscreen. Para o *7, foi criado um mascote, hoje amplamente conhecido no mundo Java, o Duke. O trabalho do Duke no *7 era ser um guia virtual ajudando e ensinando o usuário a utilizar o equipamento. O *7 tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o *7. Gosling decidiu batizá-la de “Oak”, que quer dizer carvalho, uma árvore que ele podia observar quando olhava pela sua janela.

O próximo passo era encontrar um mercado para o *7. A equipe achava que uma boa ideia seria controlar televisões e vídeo por demanda com o equipamento. Eles construíram uma demonstração chamada de MovieWood, mas infelizmente era muito cedo para que o vídeo por demanda bem como as empresas de TV a cabo pudessem viabilizar o negócio. A ideia que o *7 tentava vender, hoje já é realidade em programas interativos e também na televisão digital. Permitir ao telespectador interagir com a emissora e com a programação em uma grande rede de cabos, era algo muito visionário e estava muito longe do que as empresas de TV a cabo tinham capacidade de entender e comprar. A ideia certa, na época errada.

Entretanto, o estouro da internet aconteceu e rapidamente uma grande rede interativa estava se estabelecendo. Era este tipo de rede interativa que a equipe do *7 estava tentando vender para as empresas de TV a cabo. E, da noite para o dia, não era mais necessário construir

a infra-estrutura para a rede, ela simplesmente estava lá. Gosling foi incumbido de adaptar o Oak para a internet e em janeiro 1995 foi lançada uma nova versão do Oak que foi rebatizada para Java. A tecnologia Java tinha sido projetada para se mover por meio das redes de dispositivos heterogêneos, redes como a internet. Agora aplicações poderiam ser executadas dentro dos navegadores nos Applets Java e tudo seria disponibilizado pela internet instantaneamente. Foi o estático HTML dos navegadores que promoveu a rápida disseminação da dinâmica tecnologia Java. A velocidade dos acontecimentos seguintes foi assustadora, o número de usuários cresceu rapidamente, grandes fornecedores de tecnologia, como a IBM anunciaram suporte para a tecnologia Java.

Desde seu lançamento, em maio de 1995, a plataforma Java foi adotada mais rapidamente do que qualquer outra linguagem de programação na história da computação. Em 2004 Java atingiu a marca de 3 milhões de desenvolvedores em todo mundo. Java continuou crescendo e hoje é uma referência no mercado de desenvolvimento de software. Java tornou-se popular pelo seu uso na internet e hoje possui seu ambiente de execução presente em navegadores, mainframes, sistemas operacionais, celulares, palmtops, cartões inteligentes etc.

Padronização

Em 1997 a Sun Microsystems tentou submeter a linguagem a padronização pelos órgãos ISO/IEC e ECMA, mas acabou desistindo. Java ainda é um padrão de fato, que é controlada através da JCP Java Community Process. Em 13 de novembro de 2006, a Sun lançou a maior parte do Java como Software Livre sob os termos da GNU General Public License (GPL). Em 8 de maio de 2007 a Sun finalizou o processo, tornando praticamente todo o código Java como software de código aberto, menos uma pequena porção da qual a Sun não possui copyright.

Características

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- Orientação a objetos - Baseado no modelo de Simula67
- Portabilidade - Independência de plataforma - "escreva uma vez, execute em qualquer lugar" ("write once, run anywhere")
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP
- Segurança - Pode executar programas via rede com restrições de execução
- Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:
- Sintaxe similar a C/C++

- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM)
- É distribuída com um vasto conjunto de bibliotecas (ou APIs)
- Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa)
- Desalocação de memória automática por processo de coletor de lixo
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

Licença

A Sun disponibiliza a maioria das distribuições Java gratuitamente e obtém receita com programas mais especializados como o Java Enterprise System. Em 13 de novembro de 2006, a Sun liberou partes do Java como software livre, sob a licença GNU General Public License. A liberação completa do código fonte sob a GPL ocorreu em maio de 2007.

Exemplos de código

Método main

O método main é onde o programa inicia. Pode estar presente em qualquer classe. Os parâmetros de linha de comando são enviados para o array de Strings chamado args.

```
public class OlaMundo {
    /**
     * Método que executa o programa
     * public = É visto em qualquer lugar da aplicação. É o modificador de acesso
     * static = é iniciado automaticamente pela JVM, sem precisar de uma instância
     * void = Método sem retorno (retorno vazio)
     * main = Nome do método, que é obrigatório ser este. Recebe como parâmetro um array de
     String.
     * String[] args = Array de argumentos que podem ser repassados na chamada do programa.
     */
    public static void main(String[] args) {
        System.out.println("Olá, Mundo!"); //Imprime na tela a frase
    }
}
```

Classes

Exemplo:

```

public abstract class Animal {
    public abstract void fazerBarulho();
}

public class Cachorro extends Animal {
    public void fazerBarulho() {
        System.out.println("AuAu!");
    }
}

public class Gato extends Animal {
    public void fazerBarulho() {
        System.out.println("Miau!");
    }
}

```

O exemplo na página anterior cria a classe `Animal` e duas classes derivadas de `Animal`. É importante observar que nas classes derivadas temos a redefinição do método `fazerBarulho()`. Esta redefinição é classificada como uma sobreposição (override) de métodos. O conceito de sobreposição somente pode ser identificado e utilizado quando temos classes dispostas em um relacionamento de herança.

Java não suporta herança múltipla, devido à possibilidade de uma classe pai ter um método com o mesmo nome de outra classe pai, e gerar possíveis falhas ao chamar o método, e todas as classes em Java derivam da classe `Object`. A única possibilidade de se ver herança múltipla em Java é no uso de interfaces, pois uma classe pode herdar várias interfaces.

Classes internas

Java pode ter classes internas. Exemplos:

```

public class Cavalo extends Animal {
    public void fazerBarulho() {
        System.out.println("RIINCH!");
    }

    // Classe interna e privada. Existe só no contexto da classe "Cavalo".
    private class Parasita extends Animal {
        public void fazerBarulho() {
            System.out.println("SQRRT");
        }
    }
}

```

Interfaces

Uma interface modela um comportamento esperado. Pode-se entendê-la como uma classe que contenha apenas métodos abstratos. Embora uma classe não possa conter mais de uma super classe, a classe pode implementar mais de uma interface. **Exemplo:**

```
public interface Pesado {
    double obterPeso();
}

public interface Colorido {
    Color obterCor();
}

public class Porco extends Animal implements Pesado, Colorido {
    public void fazerBarulho() {
        System.out.println("Óinc!");
    }

    // Implementação da interface Pesado
    public double obterPeso() {
        return 50.00;
    }

    // Implementação da interface Colorido
    public Color obterCor() {
        return Color.BLACK;
    }

    // Uma propriedade só do porco
    public boolean enlameado() {
        return true;
    }
}
```

Objetos anônimos

Podemos ter também objetos anônimos, onde não é necessário instanciar o objeto em uma variável para utilizá-lo. Exemplo:

```
public class MostraBarulho {
    public static void main(String args[]) {
        new Cavallo().fazerBarulho(); // Objeto anônimo.

        // Abaixo um objeto e classe anônimos!
        new Animal() {
            public void fazerBarulho() {
                System.out.println("QUAC!");
            }
        }.fazerBarulho();
    }
}
```

Ferramentas

Frameworks

É possível utilizar frameworks para facilitar o desenvolvimento de aplicações. Dentre os mais utilizados pode-se destacar:

Hibernate ferramenta para ORM;

Junit ferramenta para auxiliar na criação de testes unitários;

Log4j ferramenta para facilitar a criação de logs na aplicação;

Spring ferramenta que auxilia principalmente implementação de injeção de dependências e inversão de controle;

Struts controlador MVC (Model 2) web;

Ambientes de desenvolvimento

É possível desenvolver aplicações em Java através de vários ambientes de desenvolvimento integrado (IDEs). Dentre as opções mais utilizadas pode-se destacar:

BlueJ — um ambiente desenvolvido por uma faculdade australiana (considerado muito bom para iniciantes);

JCreator — (gratuito/shareware) — um ambiente desenvolvido pela Xinox (recomendado para programadores iniciantes);

jEdit — (recomendado para programadores iniciantes);

IDEs completas (recomendado para programadores profissionais)

Eclipse — um projeto aberto iniciado pela IBM

IntelliJ IDEA (comercial) — uma IDE desenvolvida pela JetBrains

JBuilder — um ambiente desenvolvido pela empresa Borland

JDeveloper (gratuito OTN) — uma IDE desenvolvida pela empresa Oracle

NetBeans (software livre) — uma IDE desenvolvida pela Sun Microsystems



Emulador do Android no Eclipse, exibindo um simples "Hello World", escrito em JAVA e XML

Outras IDEs (menos populares)

Gel (IDE) — open source

Greenfoot — bem parecido com o BlueJ

JGRASP — bom para intermediários, feito pela equipe do Projeto GRASP

Java Studio Creator/Enterprise (gratuito SDN) um ambiente criado pela empresa Sun Microsystems

Workshop for WebLogic (comercial/desenvolvedor) um ambiente criado pela empresa Oracle

WebSphere Studio Application Developer um ambiente criado pela empresa IBM

Extensões

Extensões em Java:

J2ME (Micro-Edition for PDAs and cellular phones)

J2SE (Standard Edition)

J3D (A high level API for 3D graphics programming)

JAAS (Java Authentication and Authorization Service)

JAIN (Java API for Integrated Networks)

Java Card

JMX (Java Management Extensions)

JavaFX

JSF (JavaServer Faces)

JSP (JavaServer Pages)

JavaSpaces

JCE (Java Cryptography Extension)

JDBC (Java Database Connectivity)
JDMK (Java Dynamic Management Kit)
JDO (Java Data Objects)
JEE (Enterprise Edition)
Jini (a network architecture for the construction of distributed systems)
Jiro
JMF (Java Media Framework)
JMI (Java Metadata Interface)
JMS (Java Message Service)
JNDI (Java Naming and Directory Interface)
JNI (Java Native Interface)
JOGL (A low level API for 3D graphics programming, using OpenGL)
JSML (Java Speech API Markup Language)
JXTA (open source-based peer-to-peer infrastructure)
MARF (Modular Audio Recognition Framework)
OSGi (Dynamic Service Management and Remote Maintenance)
SuperWaba (JavaVMs for handhelds)

Certificações

Existem 8 tipos de certificações[10] da Sun Microsystems para Java:

Sun Certified Enterprise Architect (SCEA)
Sun Certified Mobile Application Developer (SCMAD)
Sun Certified Developer For Java Web Services (SCDJWS)
Sun Certified Business Component Developer (SCBCD)
Sun Certified Web Component Developer (SCWCD)
Sun Certified Java Developer (SCJD)
Sun Certified Java Programmer (SCJP)
Sun Certified Java Associate (SCJA)

Cada certificação testa algum tipo de habilidade dentro da plataforma e linguagem Java. Todos os testes são realizados pela empresa Person VUE e são reconhecidos internacionalmente.

Comunidade

A comunidade de desenvolvedores Java reúne-se em grupo denominados JUGs (Java User Groups). No Brasil o movimento de grupos de usuários expandiu-se bastante e tem formado alguns dos maiores grupos de usuários Java do mundo[carece de fontes], como por exemplo o PortalJava, GUG e o JavaFree.

Custo de um programador JAVA

Em CLT o preço médio de um programador JAVA é de R\$ 15,00 p/h e em projetos a partes geralmente R\$ 300,00 por cada página dinâmica desenvolvida.

Oracle adquire Sun

A Oracle Corporation (NASDAQ: ORCL) e Sun Microsystems (NASDAQ: JAVA) entrou em um acordo definitivo no dia 20 de abril 2009, sob o qual a Oracle adquiriu a Sun ações ordinárias por US \$ 9,50 por ação em dinheiro. A transação avaliada em aproximadamente US \$ 7,4 bilhões dólares líquidos de caixa da Sun.

"A aquisição da Sun transforma a indústria de TI, combinando o melhor da sua classe de software empresarial e sistemas de missão crítica de computação", disse o CEO da Oracle, Larry Ellison. "A Oracle será a única companhia que pode projetar um sistema integrado - aplicações para o disco - onde todas as peças se encaixam e funcionam juntas para que os clientes não precisem fazer isso sozinho. Nossos clientes se beneficiam como os seus custos de integração de sistemas irem para baixo, enquanto o desempenho do sistema, confiabilidade e segurança ir para cima".

Há substanciais vantagens a longo prazo dos clientes estratégicos para o Oracle possuir dois principais ativos de software Sun: Java e Solaris. Java é uma da indústria de computadores mais conhecidas marcas e tecnologias mais utilizados, e é o software mais importante do Oracle já adquiriu. Oracle Fusion Middleware, o mais rápido crescimento de negócios da Oracle, é construído em cima da linguagem Java da Sun e software. Oracle pode agora garantir a contínua inovação e investimento em tecnologia Java para o benefício dos clientes e da comunidade Java.

O sistema operacional Sun Solaris é a principal plataforma para o banco de dados Oracle, o maior negócio da Oracle, e tem sido por um longo tempo. Com a aquisição da Sun, a Oracle pode otimizar o banco de dados Oracle para algumas das únicas, características high-end

de Solaris. A Oracle está tão empenhada como nunca para Linux e outras plataformas abertas e vai continuar a apoiar e reforçar nossas parcerias fortes da indústria.

Conclusão

Java revolucionou o mundo das linguagens programação, pois A partir da plataforma Java tocar quase todos os sistemas de negócios na terra, alimentando bilhões de consumidores em aparelhos móveis e eletrônicos de consumo, para a convergência das redes de armazenamento e computação.

Bibliografia

[http://pt.wikipedia.org/wiki/Java_\(linguagem_de_programa%C3%A7%C3%A3o\)#Hist.C3.B3ria](http://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)#Hist.C3.B3ria) acessado em 26 de setembro de 2012.

<http://www.oracle.com/us/corporate/press/018363> traduzido pelo Google - acessado em 26 de setembro de 2012

<http://www.scriptbrasil.com.br/forum/lofiversion/index.php/t37064.html> acessado em 26 de setembro de 2012.