

Java (linguagem de programação)

Origem: Wikipédia, a enciclopédia livre.

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Em 2008 o Java foi adquirido pela empresa Oracle Corporation. Diferente das linguagens de programação modernas, que são compiladas para código nativo, a linguagem Java é compilada para um *bytecode* que é interpretado por uma máquina virtual (Java Virtual Machine, mais conhecida pela sua abreviação JVM). A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não é a sua única linguagem. J2ME Para programas e jogos de computador, celular, calculadoras, ou até mesmo o rádio do carro.

Índice

História

- Padronização
- Aquisição pela Oracle

Características

- Licença

Exemplos de código

- Método main
- Classes
- Interfaces
- Objetos anônimos
- Programação funcional

Ferramentas

- Frameworks
- Ambientes de desenvolvimento
- Extensões

Certificações

Comunidade

Ver também

Referências

Bibliografia

Ligações externas

História

Java



Paradigma	Orientação a objetos (baseada em classes), estruturada, imperativa, genérica, funcional, reflectiva, concorrente
Surgido em	1995
Última versão	11 (25 de setembro de 2018)
Criado por	James Gosling e Sun Microsystems
Estilo de tipagem:	estática, forte
Compiladores	GCJ, Javac
Influenciada por	Ada 83, C++, C#,[¹] Eiffel, Generic Java, Mesa, Modula-3, Object Pascal, Objective-C, UCSD Pascal, Smalltalk
Influenciou	Ada 2005, BeanShell, C#,[²] Clojure, D, ECMAScript, Fantom, Groovy, J#, JavaScript, PHP, Python, Scala
Licença:	GNU General Public License/Java Community Process
Extensão do arquivo:	.java, .class, .jar
Página oficial	www.java.com (https://www.java.com)

Em 1991, na Sun Microsystems, foi iniciado o *Green Project*, o berço do

w.java.com)

Java, uma linguagem de programação orientada a objetos. Os mentores

do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling. Eles acreditavam que, eventualmente, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos frequentemente 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 “Star Seven”), 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 através da sua janela.

O próximo passo era encontrar um mercado para o starseven. 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 infraestrutura 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* — diz-se que inspirado no café que o time de desenvolvimento consumia, oriundo da ilha de Java, e que também está presente na logomarca *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.^{[3][4][5]} Java ainda é um padrão de fato, que é controlada através da JCP Java Community Process.^[6] 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.

Aquisição pela Oracle

Em 2008 a Oracle Corporation adquire a empresa responsável pela linguagem Java, a Sun Microsystems, por US\$ 7,4 bilhões, com o objetivo de levar o Java e outros produtos da Sun ao dispor dos consumidores.

Características

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

- Orientação a objetos - Baseado no modelo de Simular;
- 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.^[7] A liberação completa do código fonte sob a GPL ocorreu em maio de 2007.^[8]

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 acima 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 a 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 implementar várias interfaces.

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 Cavalo().fazerBarulho(); // Objeto anônimo.

        // Abaixo um objeto e classe anônimos!
    }
}

```

```

    new Animal() {
        public void fazerBarulho() {
            System.out.println("QUAC!");
        }
    }.fazerBarulho();
}
}

```

Programação funcional

A partir da versão 8, o Java adiciona aspectos de linguagem funcional, permitindo utilizar técnicas funcionais, como mapeamento, redução, bem como tratar funções como variáveis. Para tanto, a linguagem utiliza interfaces para esse tipo de manipulação, quase que eliminando a necessidade do uso das chamadas classes anônimas. ^[9]

Exemplo:

```

import java.util.ArrayList;
import java.util.List;

public class Main {

    public static void main(String[] args) {
        // Lista de paradigmas
        List<String> paradigmas = new ArrayList<>();

        // Adiciona paradigmas
        paradigmas.add("Genérico (1.5)");
        paradigmas.add("Funcional (8)");

        // Abre uma stream
        paradigmas.stream()
            // Faz todos os textos na lista ficarem em maiúsculo
            .map(String::toUpperCase)
            // Faz loop em todos paradigmas em maiúsculo.
            .forEach(paradigma -> System.out.println(paradigma)); // ou
        .forEach(System.out::println);
    }
}

```

Ferramentas

Frameworks

É possível utilizar frameworks para facilitar o desenvolvimento de aplicações, dos quais os mais utilizados podem-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)

- Eclipse — (recomendado para programadores fluentes)

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:

- | | |
|---|--|
| <ul style="list-style-type: none"> ▪ Java ME (Micro-Edition for PDAs and cellular phones) ▪ Java SE (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) | <ul style="list-style-type: none"> ▪ Java EE (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 Oracle para Java:

- Oracle Certified Enterprise Architect (OCEA)
- Oracle Certified Mobile Application Developer (OCMAD)
- Oracle Certified Developer For Java Web Services (OCDJWS)
- Oracle Certified Business Component Developer (OCBCD)
- Oracle Certified Web Component Developer (OCWCD)

- Oracle Certified Java Developer (OCJD)
- Oracle Certified Java Programmer (OCJP)
- Oracle Certified Java Associate (OCJA)

Cada certificação testa algum tipo de habilidade dentro da plataforma e linguagem Java. Todos os testes são realizados pela empresa Pearson VUE^[11] 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,^[12] como por exemplo, o PortalJava, GUJ e o JavaFree.

Ver também

- C#
- Microsoft .NET
- Plataforma Java

Referências

1. «Java 5 catches up with C#» (<http://www.barrycornelius.com/papers/java5/onefile/>) (em inglês). Universidade de Oxford. Consultado em 10 de janeiro de 2010
2. «Why Microsoft's C# isn't» (<http://news.cnet.com/2008-1082-817522.html>) (em inglês). CNET. Consultado em 10 de janeiro de 2010
3. Java Study Group (<http://www.open-std.org/JTC1/SC22/JSG/>)
4. Why Java™ Was - Not - Standardized Twice (<http://csdl2.computer.org/comp/proceedings/hicss/2001/0981/05/09815015.pdf>)
5. «What is ECMA--and why Microsoft cares» (<https://web.archive.org/web/20060928175248/http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2832719,00.html>). Consultado em 16 de setembro de 2006. Arquivado do original (<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2832719,00.html>) em 28 de setembro de 2006
6. Java Community Process website (<http://www.jcp.org/en/home/index>)
7. «Sun begins releasing Java under the GPL» (<http://www.fsf.org/news/fsf-welcomes-gpl-java.html>) (em inglês). Consultado em 6 de julho de 2010
8. «JavaOne opening keynote notes and comments» (http://rollerweblogger.org/roller/entry/javaone_opening_keynote_notes) (em inglês). Consultado em 6 de julho de 2010
9. «O mínimo que você deve saber de Java 8» (<http://blog.caelum.com.br/o-minimo-que-voce-deve-saber-de-java-8/>). Blog da Caelum. Consultado em 3 de março de 2019
10. Java Certification (<http://www.sun.com/training/certification/java/index.xml>)
11. «Pearson VUE» (<http://www.pearsonvue.com>)
12. Profissao Java - Página 13- Java no Brasil (<http://www.profissaojava.com.br/arquivos/ProfissaoJava-Abertura.pdf>)

Bibliografia

- Harvey M. Deitel (2005). *Java: Como Programar* 6 ed. São Paulo: Pearson education do Brasil. 1097 páginas
- Apostila Java e Orientação a Objetos (<http://www.caelum.com.br/apostila-java-orientacao-objetos/>), licença Creative Commons, 7ª edição, pela Caelum Ensino e Inovação.
- Patrick Naughton, Dominando o Java, Guia Autorizado da Sun Microsystems, Editora Makron Books, 1997, Osborne, ISBN 80025-75540
- Brian Jepson, Java Database Programming Master Next Generation Web Database Techniques, 1997, Editora: Wiley Computer Publishing, ISBN 0-471-16518-2

- Décio Heizelman Luckow, Alexandre Altair de Melo, Programação Java para a Web: Aprenda a desenvolver uma aplicação financeira pessoal com as ferramentas mais modernas da plataforma Java, ano 2010, Editora Novatec, ISBN 978-85-7522-238-6

Ligações externas

- Sítio oficial (<http://www.oracle.com/technetwork/java/index.html>) (em inglês)
 - Sítio oficial (<http://www.oracle.com/br/technologies/java/index.html>) (em português)
 - Tutoriais oficiais (<http://download.oracle.com/javase/tutorial/>) (em inglês)
-

Obtida de "[https://pt.wikipedia.org/w/index.php?title=Java_\(linguagem_de_programação\)&oldid=56411561](https://pt.wikipedia.org/w/index.php?title=Java_(linguagem_de_programação)&oldid=56411561)"

Esta página foi editada pela última vez às 20h15min de 6 de outubro de 2019.

Este texto é disponibilizado nos termos da licença Atribuição-Compartilhualgual 3.0 Não Adaptada (CC BY-SA 3.0) da Creative Commons; pode estar sujeito a condições adicionais. Para mais detalhes, consulte as condições de utilização.