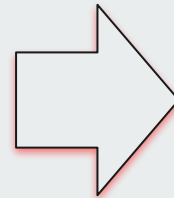


Java - Everywhere

Alunos:
Eduardo Pereira Costa
Gustavo Gomes de Souza
Jonathan Douglas Diego Tavares
Stephanie Silva Vieira Gomes



Fonte: Brands Logos e Fox on Green

Sumário



- Histórico
- Paradigma(s)
- Características mais marcantes
- Linguagens similares ou “confrontantes” (contraditórias)
- Exemplos de programas
- Considerações finais

Introdução



Fonte: GEMInIS UFSCar

Histórico da Linguagem - 1991 a 1995



- Java teve seu planejamento e criação iniciados em 1991, sendo lançado oficialmente apenas em 1995.
- A linguagem fazia parte do Green Project e seus mentores/fundadores principais foram James Gosling, Patrick Naughton e Mike Sheridan.
- O projeto foi iniciado com o objetivo de ser multiplataforma, o que contribuiu para a noção de IoT (Internet of Things) no futuro.

Histórico da Linguagem - 1995 a 2000

- Também em 1995, a Sun anunciou a primeira versão do Oak, agora denominado de Java e a Netscape anunciou outro browser com suporte a Java.
- A Sun apresentou o slogan “Write Once, Run Anywhere” para descrever os recursos multiplataforma do Java.
- Entre 1996 e 2000 o JDK 1.1 (Java Development Kit) foi lançado e o sistema operacional JavaOS foi apresentado, em seguida vieram o JDK 1.2 que foi renomeado para J2SE 1.2 e logo após o J2SE 1.3, trazendo consigo as JVMs.



Fonte: Moved by Java Timeline

Histórico da Linguagem - 2001 a 2005

- Em 2001, o CEO da Oracle, Larry Ellison, juntou-se à liderança da Sun no palco do JavaOne para demonstrar um software capaz de executar Java.
- Entre 2001 e 2004, o “java.com” foi lançado para permitir o download do Java Runtime Package.
- Em uma das conferências anuais JavaOne, a NASA demonstrou um protótipo do Mars Rover.
- Em 2004, o J2SE 5.0 foi lançado implementando tipos genéricos e varargs (variáveis como argumento).



Fonte: Moved by Java Timeline

Histórico da Linguagem - 2006 a 2014

- Em 2006, a Sun lançou o Java open source sob as licenças gerais da GNU e também, o Java SE 6.
- Em 2009, a Oracle comprou a Sun, consequentemente comprando o Java também.
- Entre 2009 e 2011, a Oracle lançou a Java Magazine, um blog oficial da linguagem de publicações técnicas sobre Java.



Fonte: Moved by Java Timeline

Histórico da Linguagem - 2015 a 2017

- Neste período o Twitter migrou sua infraestrutura para a JVM.
- Ainda em 2016 a Uber começou a utilizar o Java como uma das principais linguagens em seu desenvolvimento.
- Em 2017 o Java SE 9 foi lançado com novos recursos.
- No mesmo ano, 2017, Java é ranqueada como a linguagem de programação número um no mundo.

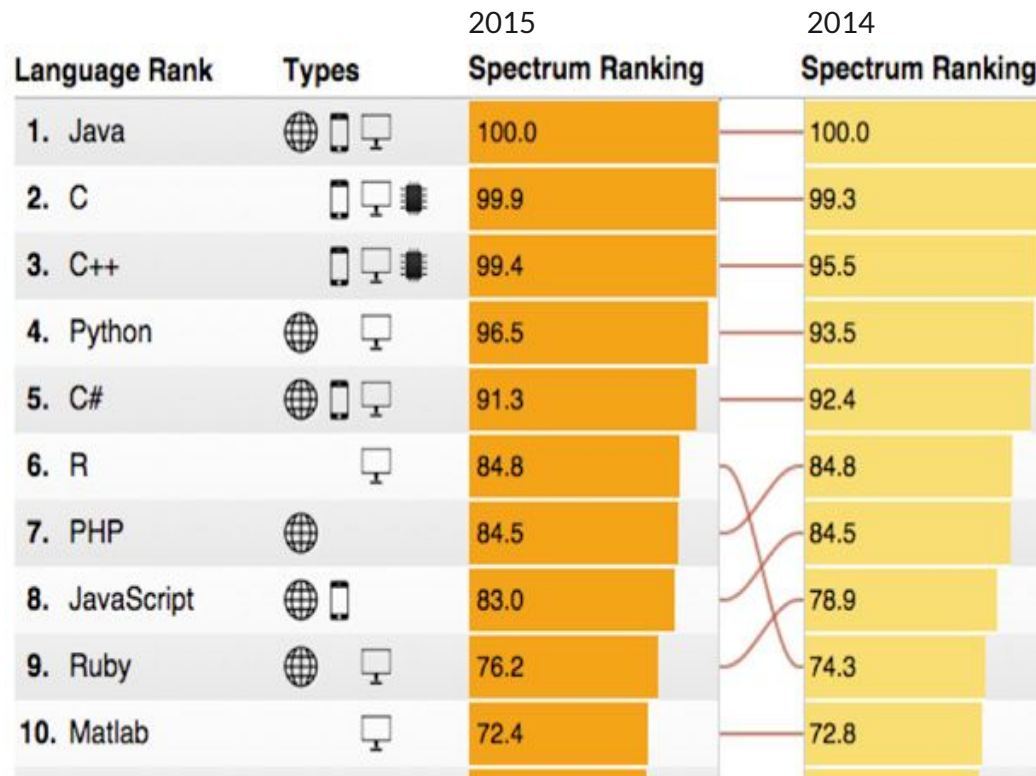


Fonte: Twitter



Fonte: Uber

Ranking de popularidade das linguagens em 2015



Fonte: IEEE SPECTRUM

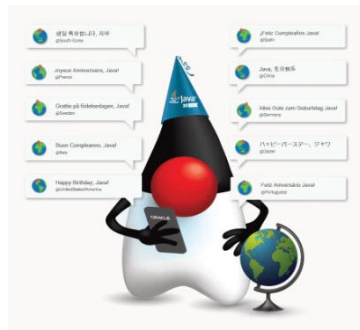
Ranking de usabilidade e popularidade das linguagens em 2017

	Indeed	IEEE Spectrum
1	Java	Java
2	JavaScript	C
3	.NET	Python
4	HTML	C++
5	Python	JavaScript
6	SQL	C#
7	C or C++	PHP
8	Node.js	HTML
9	Ruby	Ruby
10	PHP	Swift

Fonte: IEEE SPECTRUM

Histórico da Linguagem - 2018 a 2020

- O Java SE 10 e o 11 foram lançados em 2018, incluindo os recursos, versionamento de lançamentos com base na data e HTTP Cliente.
- Em 2019, o Java SE 12 e o 13 foram lançados trazendo como recurso as Switch Expressions e a capacidade de desalocar memória não utilizada.
- Em 2020, o Java SE 14 e o 15 foram lançados.



Fonte: Moved by Java Timeline


Paradigmas



Java é considerada uma linguagem multiparadigma, possuindo características dos seguintes paradigmas:

- Paradigma Imperativo
- Paradigma Orientado a Objetos
- Paradigma Funcional

Paradigma Imperativo



```
1  public class HashMapExample{
2
3      public static void HashMapExample(String args[]) {
4
5          /*Declaração de um HashMap*/
6          HashMap<Integer, String> aluno = new HashMap<Integer, String>();
7
8          /*Inserção de um elemento no HashMap*/
9          aluno.put(1, "Stephanie Silva");
10         aluno.put(2, "Gustavo Gomes");
11         aluno.put(3, "Giulia Chiucchi");
12         aluno.put(4, "Eduardo Pereira");
13         aluno.put(5, "Jonathan Douglas");
14     }
15 }
```

O paradigma imperativo tem como características principais a execução sequencial de instruções e a mudança do estado de um programa através de comandos.

Paradigma Orientado a Objetos

O principal objetivo do paradigma orientado a objetos é permitir o desenvolvimento de programas que possam representar objetos e entidades que sejam flexíveis, de fácil manutenção e que permitam reuso de "estruturas".

São os pilares desse paradigma :

- Abstração
- Encapsulamento
- Herança
- Polimorfismo



Fonte: DevMedia



```
1  abstract class Animal{
2      // atributos protegidos para que apenas os filhos, alem da propria classe pai possam acessar
3      protected String nome;
4      protected String classe;
5      //atributo privado no qual so a classe Animal pode acessar
6      private String familia;
7
8      // contrutor da classe Animal
9      Animal(String nome, String classe, String familia){
10         this.nome = nome;
11         this.classe = classe;
12         this.familia = familia;
13     }
14
15     // getters e setters para recuperar ou definir valores de atributos de objetos do tipo Animal
16     String getNome(){ return this.nome; }
17     void setNome(String nome){ this.nome = nome; }
18
19     String getClasse() { return this.classe; }
20     void setClasse(String classe){ this.classe = classe; }
21
22     String getFamilia() { return this.familia; }
23     void setFamilia(String familia){ this.familia = familia; }
24
25     abstract void emitirSom();
26
27 }
```



```
1  //Cachorro herda os atributos e metodos publicos/protegidos de Animal
2  class Cachorro extends Animal{
3      // novos atributos sao declarados para especificar um cachorro
4      private String corPelo;
5      private String raca;
6
7      Cachorro(String nome, String classe, String familia, String corPelo, String raca){
8          super(nome, classe, familia);
9          this.corPelo = corPelo;
10         this.raca = raca;
11     }
12
13     String getCorPelo(){ return corPelo; }
14     void setCorPelo(String corPelo){ this.corPelo = corPelo; }
15
16     String getRaca() { return this.raca; }
17     void setRaca(String raca){ this.raca = raca;}
18
19     // metodo utilizado apenas por objetos do tipo Cachorro
20     @Override
21     void emitirSom() {
22         System.out.println("Au au...");
23     }
24 }
```


Paradigma Funcional

O paradigma funcional é um sub-paradigma do declarativo, nele as funções são similares às presentes na matemática, sendo a saída de cada função dependente apenas de seus argumentos, e sendo assim, o estado atual do programa não altera os argumentos e a forma com que os mesmos serão avaliados.





```
1  import java.util.stream.*;
2
3  public class Exemplos {
4      public static void main(String[] args) {
5          IntStream.range(0, 10).filter(i -> i % 2 == 0).
6              forEach(System.out::println);
7      }
8  }
```

Características mais importantes



- Orientação a Objetos
- Portabilidade - JVM
- Flexibilidade da linguagem
- Ubiquidade

Orientação a Objetos

Java surgiu como uma linguagem que está intimamente conectada ao paradigma orientado a objetos, sendo também incluída ao paradigma imperativo.

Uma vez que Java lida com OO, ela torna-se uma linguagem capaz de explorar uma vasta extensão de mecanismos inerentes a esse paradigma. São exemplos desses mecanismos os 4 pilares do POO, sendo eles a Abstração, o Encapsulamento, a Herança e o Polimorfismo.



Portabilidade - JVM



Java é uma linguagem que foge aos padrões quando o assunto é o processo de compilação. É bastante conhecida pelo uso de uma máquina virtual (JVM), que por sua vez interpreta e executa programas através da geração de *bytecodes*.

Tal característica peculiar garante à linguagem a capacidade de executar qualquer programa independentemente da plataforma que estiver presente. Uma vez que a execução é feita pela JVM, torna-se possível alcançar uma portabilidade até então não vista em outras linguagens anteriores ao Java.

portabilidade

Propriedade do que é portátil, daquilo que se consegue portar, carregar ou levar de um lugar para outro.

Transferência de um número de uma operadora telefônica para outra.

[1] Dicio.com.br

Flexibilidade da linguagem

São exemplos de possíveis cenários em que Java está sendo utilizado atualmente e fora utilizado no passado:

- Banco Inter, no cenário empresarial, amplamente utilizado no Backend.
- Entretenimento voltado a jogos eletrônicos, tendo nomes como The Sims, Runescape, GTA: San Andreas(Mobile), Flappy Bird, Tibia e muito mais.
- Educação e aprendizagem de programação, sendo utilizado dentro de escolas e universidades para ensinar conceitos de lógica de programação e paradigma OO. Vale citar nesse contexto, que podem ser trabalhados os usos através do modo console ou interface gráfica.
- Na WEB, em que Java no passado foi amplamente utilizado para criar os famosos "Applets", que eram aplicações nativas em Java executadas pelos navegadores. Atualmente, é utilizado ainda mais na WEB através do JSP(Java Server Pages).



Fonte: Revista Pilates

Ubiquidade

A portabilidade da linguagem permitiu a ela alcançar tamanho feito, uma vez que mais de 2 bilhões de dispositivos no mundo inteiro estão rodando alguma aplicação feita em Java.

Citado este fato, é preciso então mencionar a importância da linguagem no mais novo conceito que está sendo introduzido nos tempos atuais, a "Internet das Coisas" ou IoT(Internet of Things).

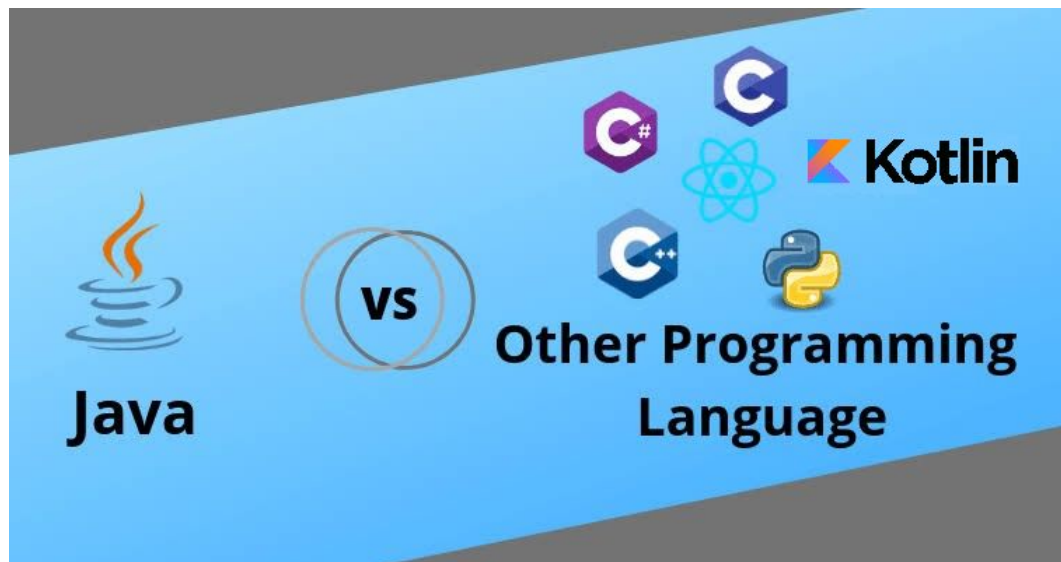
Há entorno de mais de 7 bilhões de dispositivos IoT conectados atualmente. Os especialistas esperam que esse número aumente para 10 bilhões até 2020 e 22 bilhões até 2025.



Fonte: Fox on Green

Linguagens similares ou confrontantes

- C++
- C#
- Python
- Kotlin



Fonte: Medium (adaptada)



- Tanto Java quanto C++ surgiram tendo como um de seus objetivos facilitar algo que era “difícil”, programar em C.
- Ambas as linguagens possuem Orientação a Objetos como seu paradigma principal.
- Sintaxe básica semelhante, uma vez que ambas são do tipo “C-like”.
- Em termos de segurança, algumas restrições impostas pelo Java, como acesso direto à memória, fazem com que ele seja mais seguro.
- O processo de desenvolvimento em Java tende a ser mais rápido.



- Semelhantes em vários aspectos, uma vez que C# surgiu como sendo uma linguagem orientada a objetos que buscava corrigir o que Java, pela visão da Microsoft, não fazia de forma eficiente.
- Ambas fazem uso de máquina virtual, porém, o Java (JVM) independe de SO ou plataforma, enquanto a do C# (CLR) se restringe ao SO Windows.



- Ambas são linguagens de alto nível, orientadas a objetos e pertencentes ao paradigma imperativo.
- Ao contrário de Java, Python é uma linguagem de tipagem fraca, uma vez que é totalmente interpretada.
- A curva de aprendizado de Python é mais suave se comparada à de Java, causando grande disputa no meio didático.
- Nichos de destaque distintos.



- Ambas são linguagens orientadas a objetos e pertencem ao paradigma imperativo, tendo Kotlin suporte a programação funcional.
- Java e Kotlin geram aplicações nativas para Android.
- Ao contrário de Java, Kotlin não suporta conversões implícitas de tipo, mas possui tipagem estática.
- Aplicações em Kotlin fazem menos uso de memória, porém gastam um maior espaço de armazenamento e possuem um tempo de compilação maior.

Exemplos

```
1  class HelloWorld {  
2      public static void main(String[] args) {  
3          System.out.println("Hello, World!");  
4      }  
5  }
```

Hello, World!



Fonte: Reddit

Conclusão



Fonte: Reddit

Referências



- Interactive: The Top Programming Languages 2015. Disponível em: <<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2015>>.
- ORACLE. Our world. Moved by Java. Acesso em 24 abr. 2021. Disponível em: <https://www.oracle.com/java/moved-by-java/timeline/?source=:ow:o:p:po::RC_WWMK200728P00026:DevLiveJava_PromoOcomJava&intcmp=:ow:o:p:po::RC_WWMK200728P00026:DevLiveJava_PromoOcomJava#2015>.
- ORACLE. O Que é Internet of Things (IoT)? — Oracle Brasil. Acesso em 24 abr. 2021. Disponível em: <<https://www.oracle.com/br/internetofthings/whatisiot/>>.
- The Most Popular Programming Languages of 2017. Disponível em: <<https://newrelic.com/blog/nerd-life/popular-programming-languages-2017#:~:text=Java%20tops%20both%20lists%2C%20and,especially%20important%20with%20larger%20companies.%E2%80%9D>>.

Referências



- GABBRIELLI, M. , MARTINE, S. Programming Languages: Principles and Paradigms. Springer-Verlag , Londres. 2010. Acesso em 23 abr. 2021. Disponível em: <http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages_Principles_and_Paradigms_thereids1106.pdf>.
- ORACLE. Java.util.fuction. Acesso em 23 abr. 2021. Disponível em: <<https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html>>.
- SILVA, F. Programação Orientada a Objetos. Universidade Federal de Uberlândia. Uberlândia, Brasil. 2004. Acesso em 23 abr. 2021. Disponível em: <<http://www.facom.ufu.br/~flavio/poo/files/2004-01/Poo-tad.pdf>>.
- SINGH, K. , IANCULESCU,A. , TORJE, L. Design Patterns and Best Practices in Java. 2018. Acesso em 23 abr. 2021. Disponível em: <https://subscription.packtpub.com/book/application_development/9781786463593/1/ch01lvl1sec13/object-oriented-paradigm>.
- WEISFELD, M. The Object-Oriented Thought Process. 2009. Acesso em 23 abr. 2021. Disponível em: <https://www.researchgate.net/publication/262350549_The_object-oriented_thought_process_fourth_edition_by_Matt_Weisfeld>.