



Linguagem Java

Prof. MSc. Tiago Araújo

tiagodavi70@gmail.com

Histórico

Sun Microsystems, 90/91: um grupo de desenvolvimento recebeu a tarefa de projetar uma linguagem de programação pequena que pudesse ser usada em dispositivos eletrodomésticos (controles remoto etc.) – Projeto *Green*

Esses dispositivos não possuem muita memória, então a linguagem teve que ser bem pequena e gerar códigos restritos.

A linguagem intitulada de *Oak* (carvalho) foi projetada para ser utilizado em diferentes tipos de hardware, graças ao uso de uma Máquina Virtual.

Histórico

1991: decidiu-se por nomear a linguagem como Java (apelido dado pelos americanos ao café).

1992: entregue o primeiro produto chamado *7 - *StarSeven* (controle remoto extremamente inteligente).

Não houve interesse comercial. Buscou-se outro modo para comercializar a tecnologia.

Enquanto isso a Internet estava crescendo e era necessário um browser para poder acessá-la.

Decidiu-se então pela construção deste browser, denominado HotJava, que foi inteiramente programado em Java

Histórico

A grande inovação em relação ao Java se deu em 1995.

A Netscape decidiu dar suporte à linguagem a partir de seu navegador Web para Internet (Netscape Navigator 2.0).

A partir disto, todo usuário deste navegador passou a ser um potencial usuário da linguagem Java.

Desde então, um fluxo constante de novidades vem acontecendo, culminando com a linguagem Java tornando-se a atual “coqueluche” dos programadores.

Histórico

Em abril de 2009, a Oracle ofereceu **US\$ 7,4 bilhões** pela aquisição da Sun Microsystems e a proposta foi aceita.

Existem mais de 38 bilhões de dispositivos com JVM ativas

Hoje funciona como a linguagem de programação por trás de dispositivos Android

Programming, Scripting, and Markup Languages

Unsurprisingly, for the eighth year in a row, JavaScript has maintained its stronghold as the most commonly used programming language. Going further down the list, we also see moderate gains for TypeScript, edging out C in terms of popularity. Additionally, Ruby, once in the top 10 of this list as recently as 2017, has declined, being surpassed by newer, trendier technologies such as Go and Kotlin.

All Respondents

Professional Developers

57,378 responses; select all that apply



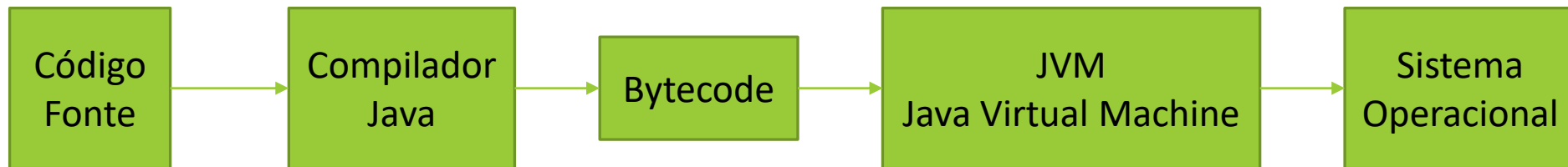
Java

Java é uma linguagem de programação orientada a objetos e uma plataforma computacional

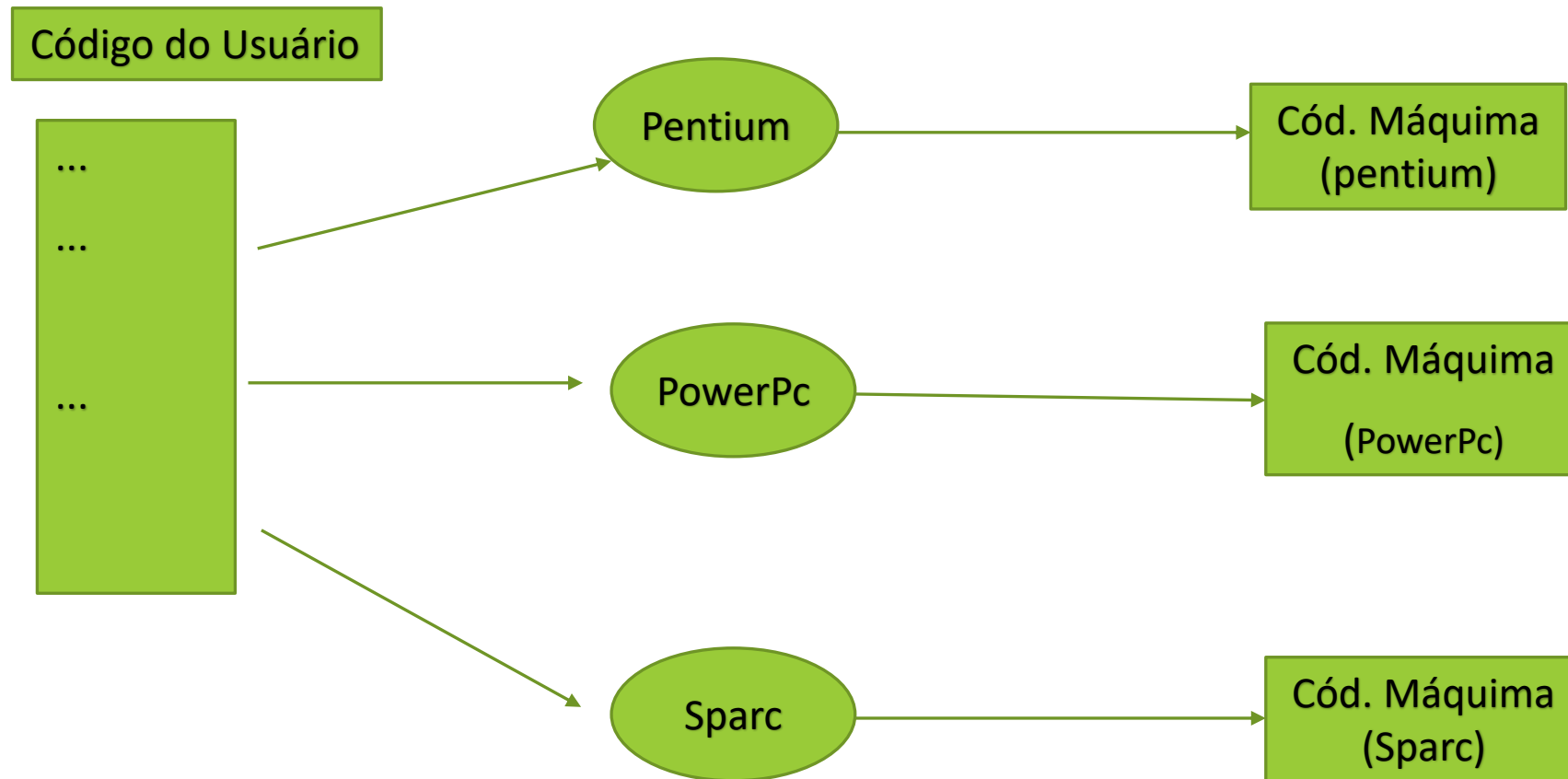
Java como plataforma permite que se desenvolva código somente uma vez, e ele pode ser reusado em vários outros contextos com a mínima modificação possível

Java é popular na Web, mas ganhou muito espaço em vários outros ambientes

Java



Java



Componentes Principais

JVM = Java Virtual Machine, apenas a virtual machine, esse download não existe, ela sempre vem acompanhada.

JRE = Java Runtime Environment, ambiente de execução Java, formado pela JVM e bibliotecas, tudo que você precisa para executar uma aplicação Java.

JDK = Java Development Kit: Ele é formado pela JRE somado a ferramentas, como o compilador.

Componentes do JDK

javac	- Compilador Java
java	- Interpretador Java (Java Virtual Machine)
jdb	- Java Debugger
javadoc	- Gerador Automáticos de Documentos
jar	- Gerenciador de arquivos .jar (zipador de classes)
jre	- Java Run-Time Environment (JVM para distribuir as apps.)
appletviewer	- Visualizador de applets

Utilização

Main.java - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
public class Main {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Nome

Main.class

Main.java

C:\Windows\System32\cmd.exe

Microsoft Windows [versão 10.0.19042.1052]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\tiago\IdeaProjects\sandbox\src>javac Main.java

C:\Users\tiago\IdeaProjects\sandbox\src>java Main
Hello World

C:\Users\tiago\IdeaProjects\sandbox\src>

Básico da Linguagem

Estrutura básica

```
public class Main {
```

 → Declaração de classe

```
    public static void main(String[] args) {
```

 → Função principal, executa instruções

```
        System.out.println("Hello World");
```

 → Instruções

```
    }
```

```
}
```

{ } marca um bloco de instruções

Vamos deixar a declaração de classe e todo o preâmbulo para outro momento e focar nas instruções

Sumário

Variáveis

Operadores

Controle de fluxo

Variáveis

Nomes das variáveis diferenciam maiúsculas e minúsculas

Os nomes podem ter qualquer caracter Unicode válido – letras, dígitos e *underscore*

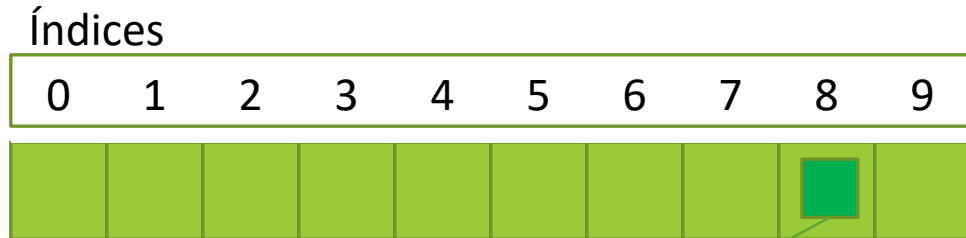
Normalmente em letra minúscula

Variáveis

Tipo	Tamanho	Valor padrão	Faixa de valores
<i>boolean</i>	n/a	false	true ou false
<i>byte</i>	8 bits	0	-128 a 127
<i>char</i>	16 bits	(não designado)	\u0000' \u0000' a \uffff' ou 0 a 65535
<i>short</i>	16 bits	0	-32768 a 32767
<i>int</i>	32 bits	0	-2147483648 a 2147483647
<i>long</i>	64 bits	0	-9223372036854775808 a 9223372036854775807
<i>float</i>	32 bits	0.0	1.17549435e-38 a 3.4028235e+38
<i>double</i>	64 bits	0.0	4.9e-324 a 1.7976931348623157e+308

```
boolean result = true;  
char capitalC = 'C';  
byte b = 100;  
short s = 10000;  
int i = 100000;  
float f = 3.65f;  
double d = 3.76;
```


Variáveis



Tamanho: 10

Elemento no índice 8

```
byte[] anArrayOfBytes;  
short[] anArrayOfShorts;  
long[] anArrayOfLongs;  
float[] anArrayOfFloats;  
double[] anArrayOfDoubles;  
boolean[] anArrayOfBooleans;  
char[] anArrayOfChars;  
String[] anArrayOfStrings;
```

```
int[] umArray;  
umArray = new int[10];
```

```
umArray[0] = 100; // inicializar primeiro elemento  
umArray[1] = 200; // inicializar segundo elemento  
umArray[2] = 300; // e por diante
```

```
int[] umArray = {  
    100, 200, 300,  
    400, 500, 600,  
    700, 800, 900, 1000  
};
```

```
int [][] umaMatriz;  
umaMatriz = { {1, 2}, {3, 4} };
```

Operadores

Operador	Descrição
+	Adição (também usado para concatenação de String)
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (Resto da divisão)

$1 + 2 = 3$
 $3 - 1 = 2$
 $2 * 2 = 4$
 $4 / 2 = 2$
 $2 + 8 = 10$
 $10 \% 7 = 3$

++ e -- são operadores unários de soma e subtração no mesmo lugar

$x++ = x + 1$

$y-- = x - 1$

Operadores

Operador	Descrição
==	Igualdade
!=	Desigualdade
>	Maior que
>=	Maior e igual que
<	Menor que
<=	Menor ou igual que

3 == 3

4 != 5

6 > 2

8 >= 7

10 < 12

7 <= 8

Operadores

Operador	Descrição
&&	E lógico-condicional
	OU lógico-condicional
!	NÃO lógico-condicional

true && true
false || ftrue
!false

Controle de fluxo

Existem vários métodos de Entrada e Saída em Java, mas usar os métodos de linha de comando funciona em muitos ambientes sem precisar de cuidados extras.

```
System.out.println(<Saída aqui>)
```

```
Scanner in = new Scanner(System.in); // Essa sintaxe de atribuição com new será apresentada depois
```

```
String s = in.nextLine(); // Leitura de String  
int a = in.nextInt(); // Leitura de inteiro  
float b = in.nextFloat(); // Leitura de real  
in.close(); // Encerra a entrada
```

Controle de fluxo

```
import java.util.Scanner; // Java tem um pacote voltado para leitura de dados

class EntradaSaida {
    public static void main(String args[])
    {
        // Usando Scanner para capturar uma entrada
        Scanner in = new Scanner(System.in);

        String s = in.nextLine(); // Leitura de String
        System.out.println("Você entrou a String: " + s);

        int a = in.nextInt(); // Leitura de inteiro
        System.out.println("Você entrou o inteiro: " + a);

        float b = in.nextFloat(); // Leitura de real
        System.out.println("Você entrou o real: " + b);

        // fechando scanner
        in.close();
    }
}
```

Controle de fluxo

As estruturas de decisão *if-then*, *if-then-else* e *switch* são as estruturas de decisão em Java

```
if (<condição lógica>) {  
    <instruções>  
}
```

```
if (<condição lógica>) {  
    <instruções>  
} else {  
    <instruções>  
}
```

```
switch (<variável>) {  
    case <valor1>:  
        <instruções>  
        break;  
    case <valor2>:  
        <instruções>  
        break;  
    case <valor3>:  
        <instruções>  
        break;  
    .....  
    default:  
        <instruções>  
        break;  
}
```

```
public class Decisao {  
  
    public static void main(String[] args) {  
  
        double valor = 21.75;  
        String fidelidade = "Estudante";  
        double desconto;  
        double valorDesconto;  
  
        if (valor > 20) {  
            System.out.println("Voce vai ganhar pontos de fidelidade!");  
        }  
  
        switch (fidelidade) {  
            case "Sem desconto":  
                desconto = 0;  
                break;  
            case "Estudante":  
                desconto = 0.25;  
                break;  
            default:  
                desconto = 0.1;  
                break;  
        }  
        valorDesconto = valor * desconto;  
        if (desconto != 0) {  
            System.out.println("Voce vai ter o desconto de: " + valorDesconto);  
        } else {  
            System.out.println("Valor final sem desconto: " + valor);  
        }  
    }  
}
```


Controle de fluxo

As estruturas de repetição em Java são o *for*, *while* e *do-while*

```
for (<inicialização>; <condição de parada>; <incremento>) {  
    <instruções>  
}
```

```
for (<tipo> <elemento> : <iterador>) {  
    <instruções>  
}
```

```
while (<expressão lógica>) {  
    <instruções>  
}
```

```
do {  
    <instruções>  
} while (<expressão lógica>;
```

```

public class Laco {

    public static void main(String[] args) {

        int[] numeros = {1,2,3,4,5,6,7,8,9,10};

        System.out.println("Laco for");
        for (int i = 0; i < numeros.length ; i++){
            System.out.println("Item: " + numeros[i]);
        }

        System.out.println("Laco for melhorado");
        for (int item : numeros) {
            System.out.println("Item: " + item);
        }

        System.out.println("Laco while");
        int j = 0;
        while (j < 10){
            System.out.println("Numero: " + j);
            j++;
        }

        System.out.println("Laco do-while");
        j = 0;
        do {
            System.out.println("Numero: " + j);
            j++;
        } while (j < 10);
    }
}

```

Prática

Escreva um programa em Java que receba 3 valores e mostre a média entre eles.

Escreva um programa em Java que receba 3 valores e mostre "crescente" se estiverem em ordem crescent, "decrescente" se estiverem em ordem decrescente e "sem ordem" caso nenhum dos dois.

Escreva um programa em Java que mostre na tela os quadrados de 1 até 15.