

JAVA

Dominando Java



Domine os fundamentos, estruturas e práticas essenciais de Java para criar aplicações poderosas.

João Vitor

Introdução ao Java

Entendendo os pilares e o propósito da linguagem Java

Java é uma linguagem de programação versátil e poderosa, usada no mundo todo para criar desde aplicativos Android até sistemas corporativos. Com sua filosofia de "escreva uma vez, execute em qualquer lugar", Java permite que o mesmo código funcione em diferentes plataformas. Robusta, segura e orientada a objetos, é ideal tanto para iniciantes quanto para profissionais. Neste ebook, você vai explorar os pilares da linguagem e aprender a construir aplicações com confiança.




01

Primeiros Passos com Java

Aprenda a escrever, compilar e executar seu primeiro código
Java passo a passo. Vamos ver:

Primeiros Passos com Java

Java é uma linguagem de programação poderosa, bastante usada em empresas, bancos, startups e sistemas embarcados. Seu grande diferencial está em ser orientada a objetos, multiplataforma e segura. Para rodar um código em Java, você precisa criar uma classe. O ponto de partida é sempre o método main, que é o primeiro a ser executado.



```
public class OlaMundo {  
    public static void main(String[] args) {  
        System.out.println("Olá, mundo!");  
    }  
}
```

Esse exemplo imprime uma mensagem no terminal. Parece simples, mas aqui você já vê conceitos importantes:

- *public class* define uma classe.
 - *main* é o método inicial.
- *System.out.println()* é usado para exibir algo no console.

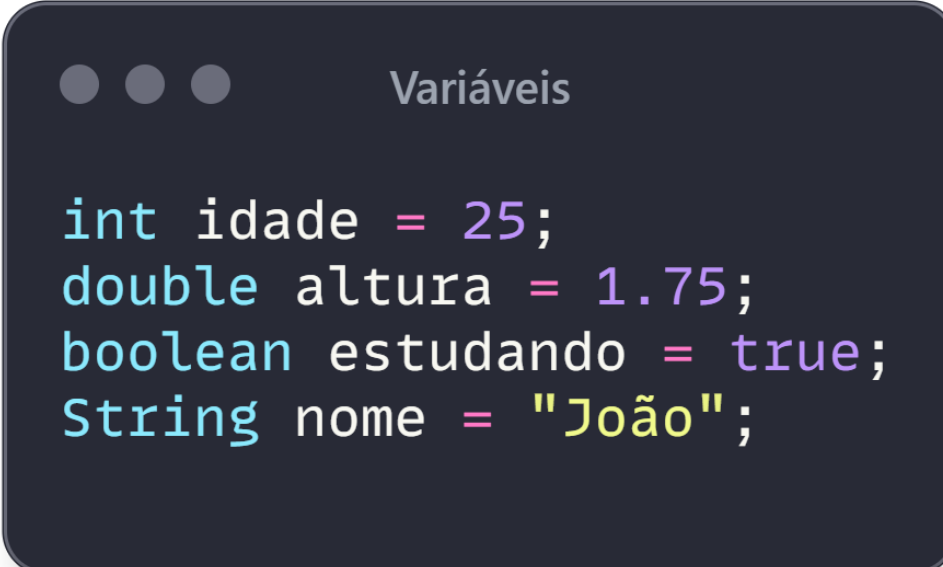
02

Variáveis e Tipos

Toda linguagem precisa guardar informações. Em Java, usamos variáveis para isso. Você sempre precisa dizer o tipo da informação antes de usá-la.

Variáveis e Tipos

Aqui você aprenderá como o Java lida com informações: como declarar variáveis, escolher o tipo correto e fazer operações básicas. Com esse conhecimento, você poderá guardar e manipular dados no seu programa.



```
int idade = 25;
double altura = 1.75;
boolean estudando = true;
String nome = "João";
```

Explicação:

- *int* armazena números inteiros.
- *double* armazena números com casas decimais.
- *boolean* representa verdadeiro ou falso.
- *String* guarda textos.

03

Tomando Decisões com `if` e `else`

Em programação, você frequentemente precisa tomar decisões. Para isso, usamos condicionais.

Tomando Decisões com if e else

A capacidade de tomar decisões é essencial em qualquer programa. Neste capítulo, você verá como escrever blocos de código que se adaptam a diferentes situações, usando estruturas condicionais.

```
TomandoDecisao

int idade = 16;

if (idade ≥ 18) {
    System.out.println("Maior de idade");
} else {
    System.out.println("Menor de idade");
}
```

O código verifica a idade e imprime uma mensagem de acordo com a condição.

Se a idade for maior ou igual a 18, irá imprimir “Maior de idade”, caso o contrário, “Menor de idade”.

Como nesse exemplo definimos a idade como 16, logo no terminal irá aparecer “Menor de idade”.

04

Repetições: for e while

Laços de repetição ajudam a automatizar tarefas, como imprimir 10 vezes uma mesma ação.

Repetições: for e while

Repetir ações automaticamente é uma das bases da programação. Você aprenderá os principais laços de repetição e quando usar cada um, para escrever códigos mais eficientes.

```
for
for (int i = 1; i ≤ 5; i++) {
    System.out.println("Contando: " + i);
}
```

No laço for, você define onde começa, onde termina e de quanto em quanto a variável será incrementada. No exemplo, ela começa em 1, vai até 5 e soma de 1 em 1, exibindo no terminal: Contando: 1, Contando: 2, ..., Contando: 5.

```
while
int i = 1;
while (i ≤ 5) {
    System.out.println("Contando: " + i);
    i++;
}
```

No laço while, vai funcionar da mesma forma praticamente, mas ele só vai entrar no loop enquanto alguma condição for verdadeira.

05

Separação e Reaproveitando Código

Métodos ajudam a organizar ações específicas, permitindo reaproveitamento e clareza no código.

Separação e Reaproveitando Código

Métodos tornam seu código mais limpo, organizado e reutilizável. Neste capítulo, você vai criar suas próprias funções e aprender a passar e retornar informações entre elas.

```
Reaproveitando

public class Calculadora {
    public static int somar(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int resultado = somar(10, 5);
        System.out.println("Resultado: " + resultado);
    }
}
```

O método somar é responsável apenas por realizar a soma de dois números. No main, chamamos esse método sempre que precisar fazer a operação de soma, sem precisar repetir o código. Isso torna o programa mais organizado e facilita futuras modificações. Se quiser adicionar mais operações no futuro, basta criar novos métodos, sem bagunçar o código principal.

06

Arrays: Armazenando Muitos Valores

Arrays são usados quando queremos guardar uma lista de itens.

Arrays: Armazenando Muitos Valores

Nem sempre um valor é suficiente. Você verá como usar arrays para trabalhar com conjuntos de dados, iterar sobre eles e realizar operações em lote.

```
Array

String[] frutas = {"Maçã", "Banana", "Uva"};

for (String fruta : frutas) {
    System.out.println("Fruta: " + fruta);
}
```

Neste exemplo, usamos um array de String para armazenar uma lista de frutas. O laço for-each percorre cada item do array e imprime o nome de cada fruta na tela. Arrays são úteis quando temos que trabalhar com um grupo de dados semelhantes, como uma lista de produtos, nomes ou valores numéricos. O for-each facilita a iteração sem precisar lidar com índices diretamente.

AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu Github

.

Esse conteúdo foi gerado com fins didáticos de construção,
não foi realizado uma validação cuidadosa humana no
conteúdo e pode conter erros gerados por uma IA.



<https://github.com/felipeAguiarCode/prompts-recipe-to-create-a-ebook>

