

Conceitos iniciais para devs iniciantes}



Pablynne Caldeira

Sumário

1. Boas-vindas ao Mundo Java

- Por que aprender Java?

2. Preparando o Terreno

- Instalando Java: Aventura Iniciante
- Configurando seu Ambiente de Desenvolvimento

3. Primeiros Passos com Java

- 'Olá, Mundo!'
- Entendendo a Sintaxe Básica

4. Variáveis e Tipos de Dados

- Variáveis: Guardando Informações
- Tipos de Dados: Números, Texto e Mais

5. Conclusão e agradecimentos

*Esse material foi produzido para fins didáticos.



1 Boas vindas ao mundo Java



POR QUE APRENDER JAVA?:

Aventuras e oportunidades no mundo do código

Se você está começando no mundo da programação, pode estar se perguntando: "Por que escolher Java?". Existem muitas linguagens de programação por aí, então o que torna o Java especial? Vamos descobrir algumas razões pelas quais aprender Java é uma excelente escolha para iniciantes e profissionais experientes.

1. Popularidade e Demanda no Mercado

- Java é uma das linguagens de programação mais populares do mundo. Ela é amplamente utilizada em diversas indústrias, desde desenvolvimento de aplicativos para Android até sistemas corporativos complexos. Essa popularidade se traduz em uma alta demanda por desenvolvedores Java no mercado de trabalho, oferecendo muitas oportunidades de emprego e crescimento profissional.

2. Portabilidade e Multiplataforma

- Uma das maiores vantagens do Java é sua portabilidade. O lema "Write Once, Run Anywhere" (Escreva uma vez, execute em qualquer lugar) destaca a capacidade do Java de rodar em qualquer dispositivo ou sistema operacional que tenha a Máquina Virtual Java (JVM) instalada. Isso significa que você pode desenvolver um programa no seu computador e ele funcionará em qualquer outro sistema que suporte Java, sem necessidade de alterações.

3. Comunidade e Suporte

- Java tem uma comunidade enorme e ativa. Isso é uma grande vantagem para iniciantes, pois há uma abundância de recursos disponíveis, como tutoriais, fóruns, documentações e bibliotecas de código. Se você encontrar um problema ou tiver uma dúvida, é muito provável que alguém já tenha enfrentado a mesma situação e possa ajudar.

4. Ferramentas e Ecossistema

- A linguagem Java vem acompanhada de um robusto conjunto de ferramentas e bibliotecas que facilitam o desenvolvimento. IDEs (Ambientes de Desenvolvimento Integrado) como IntelliJ IDEA, Eclipse e NetBeans fornecem

funcionalidades avançadas que ajudam a escrever, depurar e manter código de forma eficiente. Além disso, o vasto ecossistema de bibliotecas e frameworks, como Spring e Hibernate, permite que você desenvolva aplicações complexas com menos esforço.

6. Base para Outras Tecnologias

- Aprender Java fornece uma base sólida que pode ser aplicada a outras tecnologias e linguagens de programação. Muitos conceitos de programação orientada a objetos, padrões de design e melhores práticas que você aprenderá com Java são relevantes em outras linguagens como C#, Python e até mesmo em linguagens modernas como Kotlin e Scala.

7. Versatilidade

- Java é extremamente versátil. Você pode usá-lo para desenvolver uma ampla gama de aplicações, desde pequenos aplicativos para dispositivos móveis até grandes sistemas distribuídos e aplicações web. Essa versatilidade significa que, independentemente do tipo de desenvolvimento que você deseja seguir, o conhecimento de Java será útil.

8. Crescimento Pessoal e Profissional

- Aprender Java não só aumenta suas habilidades técnicas, mas também melhora suas habilidades de resolução de problemas e pensamento lógico. A prática da programação ajuda a desenvolver uma mentalidade analítica e a capacidade de quebrar problemas complexos em partes menores e mais gerenciáveis, habilidades que são valiosas em qualquer carreira.

Conclusão

- Aprender Java é uma escolha estratégica tanto para iniciantes quanto para programadores experientes. A combinação de sua popularidade, portabilidade, comunidade ativa e robusto conjunto de ferramentas faz de Java uma linguagem poderosa e versátil. Ao investir seu tempo em aprender Java, você estará construindo uma base sólida para uma carreira promissora no campo da tecnologia.

Então, pronto para mergulhar no mundo de Java? Vamos começar essa emocionante jornada juntos!

2 Preparando o terreno



INSTALANDO JAVA

Aventura Iniciante

Por que Precisamos do JDK?

Java Development Kit (JDK) é o pacote básico que você precisa para começar a programar em Java. Ele contém tudo, desde o compilador até a Máquina Virtual Java (JVM), que roda seus programas. É como ter um supercomputador portátil que pode entender e executar seu código.

❖ Passo a passo para instalação

Windows

1. Baixe o JDK: Visite oracle.com/java/technologies/javase-downloads.html e baixe a versão mais recente para Windows.
2. Instale o JDK: Abra o arquivo baixado e siga as instruções na tela. É tão simples quanto instalar um jogo no seu computador.
3. Configure o Ambiente:
 - Vá para Painel de Controle > Sistema e Segurança > Sistema > Configurações Avançadas do Sistema.
 - Clique em Variáveis de Ambiente e encontre a variável Path.
 - Adicione o caminho C:\Program Files\Java\jdkX.X.X\bin (substitua X.X.X pela versão do JDK).

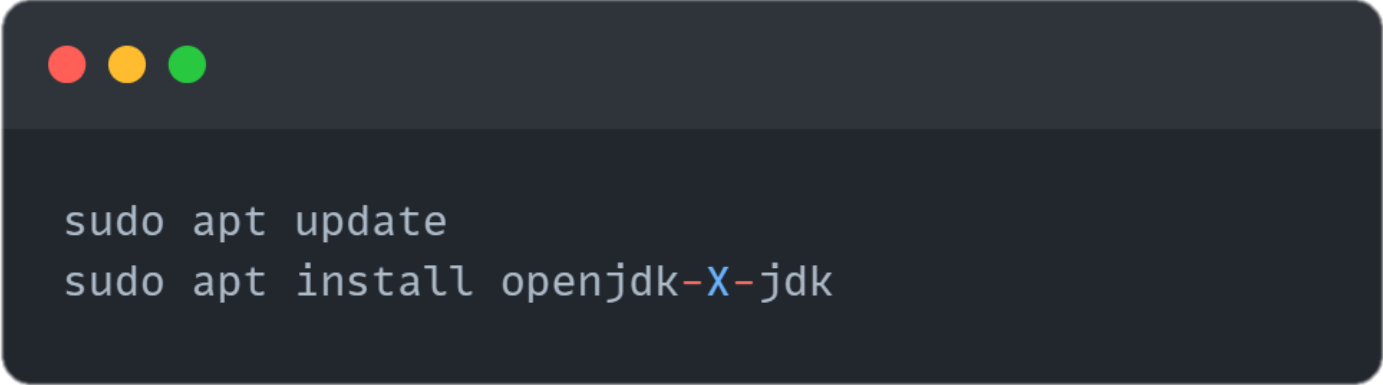
macOS

1. Baixe o JDK: Acesse o mesmo site e baixe a versão para macOS.
2. Instale o JDK: Abra o arquivo .dmg e siga as instruções.
3. Configure o PATH:
 - Abra o Terminal.
 - Digite `nano ~/.zshrc`.
 - Adicione a linha `export PATH=/Library/Java/JavaVirtualMachines/jdkX.X.X.jdk/Contents/Home/bin:$PATH`. Salve e feche o editor (Ctrl + X, depois Y, depois Enter).
 - Digite `source ~/.zshrc`.

Linux

1. Instale via Terminal:

- Para distribuições baseadas em Debian (como Ubuntu):



```
sudo apt update
sudo apt install openjdk-X-jdk
```

- Para distribuições baseadas em Red Hat:



```
sudo yum update
sudo yum install java-X-openjdk-devel
```

2. Configure o PATH:

- Abra o terminal.
- Digite `nano ~/.bashrc`. Adicione `export PATH=/usr/lib/jvm/java-X-openjdk/bin:$PATH`.
- Salve e feche (Ctrl + X, depois Y, depois Enter).
- Digite `source ~/.bashrc`.

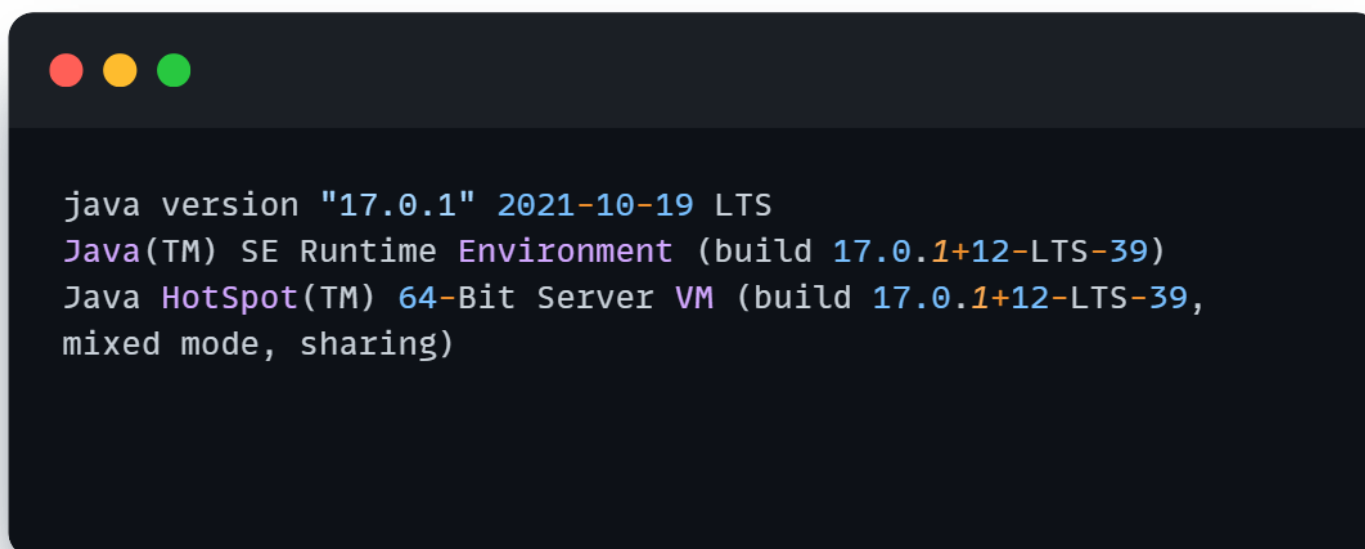
Verificando a Instalação

Para ter certeza de que tudo está funcionando, abra o Terminal (ou Prompt de Comando no Windows) e digite:



```
java -version
```


Você deve ver algo parecido com:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal is as follows:

```
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39,
mixed mode, sharing)
```

CONFIGURANDO SEU PLAYGROUND

Agora que temos o Java instalado, é hora de configurar um ambiente de desenvolvimento onde você vai escrever e testar seu código. Existem várias opções, mas vamos focar nas mais populares e fáceis de usar.

Escolhendo uma IDE

Uma IDE (Integrated Development Environment) é como seu estúdio de criação onde você escreve, edita e testa seu código. Aqui estão duas ótimas opções:

➤ IntelliJ IDEA

- **Baixe IntelliJ IDEA:** Acesse jetbrains.com/idea/download e baixe a versão Community (gratuita).
- **Instale e Configure:** Siga as instruções de instalação. Ao abrir pela primeira vez, você pode personalizar a aparência e as configurações de acordo com suas preferências.

➤ Eclipse

- **Baixe Eclipse:** Visite eclipse.org/downloads e baixe o instalador.
- **Instale e Configure:** Execute o instalador e escolha "Eclipse IDE for Java Developers". Siga as instruções e personalize como desejar.

Explorando sua IDE

Vamos dar uma olhada rápida em como usar sua nova IDE:

1. **Criar um Novo Projeto:** Abra a IDE e escolha a opção para criar um novo projeto Java.
2. **Escreva seu Primeiro Código:** Crie uma nova classe Java e digite:

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

3. **Execute o Programa:** Clique no botão de execução (geralmente um ícone de play) e veja a mágica acontecer!

Pronto para Codificar!

Com o Java e sua IDE configurados, você está pronto para começar a codificar. Imagine que seu ambiente de desenvolvimento é como um parque de diversões cheio de atrações. Cada linha de código que você escrever será uma nova aventura!

No próximo capítulo, vamos criar nosso primeiro programa em Java e começar a entender a sintaxe básica. Prepare-se, porque a diversão está apenas começando!



3 Primeiros passos com Java



‘Olá, mundo!’

Bem-vindo ao seu primeiro mergulho no mundo da programação em Java! Vamos começar com um clássico absoluto e depois explorar os conceitos básicos. Preparado? Vamos lá!

O Clássico "Olá, Mundo!" Todo mundo começa com um simples "Olá, Mundo!". É um rito de passagem no mundo da programação. Vamos escrever nosso primeiro programa em Java para mostrar essa mensagem na tela.

Escrevendo o Código:

1. Abra sua IDE: Se você escolheu IntelliJ IDEA, Eclipse ou qualquer outra, abra-a agora.
2. Crie um Novo Projeto:
 - **Em IntelliJ IDEA:** Clique em File > New > Project.
 - **Em Eclipse:** Clique em File > New > Java Project.
3. Crie uma Nova Classe:
 - **Em IntelliJ IDEA:** Clique com o botão direito na pasta src e selecione New > Java Class. Nomeie-a como HelloWorld.
 - **Em Eclipse:** Clique com o botão direito no projeto e selecione New > Class. Nomeie-a como HelloWorld.
4. Digite o código

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

Executando o Programa

Agora, vamos ver o "Olá, Mundo!" em ação.

Execute o Programa:

- **Em IntelliJ IDEA:** Clique no ícone de play próximo ao código.
- **Em Eclipse:** Clique com o botão direito na classe HelloWorld e selecione Run As > Java Application.

Parabéns! Se tudo correu bem, você verá "Olá, Mundo!" impresso no console. Você escreveu e executou seu primeiro programa Java!

ENTENDENDO A SINTAXE BÁSICA

Quebrando o Código

Vamos entender o que cada parte do seu código "Olá, Mundo!" faz.

- **public class HelloWorld {**: Define uma classe pública chamada HelloWorld.
- **public static void main(String[] args) {**: Este é o ponto de entrada do seu programa. Quando você executa o programa, é aqui que ele começa.
- **System.out.println("Olá, Mundo!");**: Este comando imprime a mensagem "Olá, Mundo!" na tela.

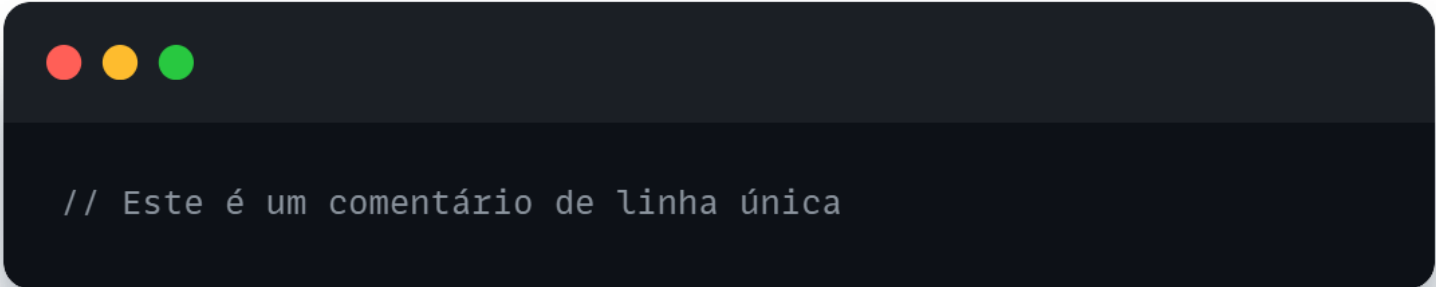
Estrutura de um Programa Java

Todo programa Java é composto por classes e métodos. A classe é como um plano, e os métodos são como instruções específicas. O método main é onde o programa começa a executar.

Comentários no Código

Comentários são anotações que você pode adicionar ao seu código para explicar o que ele faz. Eles são ignorados pelo compilador. Existem dois tipos principais de comentários:

- **Comentários de linha única:** Usam `//`. Exemplo:



```
// Este é um comentário de linha única
```

- **Comentários de múltiplas linhas:** Usam `/* ... */`. Exemplo:

```
/* Este é um  
   comentário de múltiplas linhas */
```

Agora que você escreveu seu primeiro programa e entendeu a sintaxe básica, está no caminho certo para se tornar um mestre em Java! No próximo capítulo, vamos nos aprofundar em operadores e estruturas de controle, para que você possa fazer seu código tomar decisões e repetir ações.

Continue assim, e logo você estará criando programas incríveis!



4 Variáveis e tipos de dados



VARIÁVEIS

Vamos explorar o universo das variáveis e dos tipos de dados em Java. Pense neles como as ferramentas básicas que você usará para construir seus programas. Assim como um artista precisa de tintas e pincéis, você precisa entender variáveis e tipos de dados para criar suas obras-primas de código.

Variáveis: Guardando Tesouros

O que são Variáveis?

Variáveis são como pequenas caixas onde você pode guardar informações importantes que seu programa usa. Imagine que você tem uma caixa rotulada "idade" e coloca um número nela, como 25. Sempre que você precisar saber a idade, basta olhar nessa caixa.

Declarando Variáveis

Para declarar uma variável em Java, você precisa especificar o tipo de dado e o nome da variável. É assim que você diz ao Java que tipo de tesouro (dado) vai guardar na caixa (variável).

Exemplo:

```
1  int idade = 25;  
2
```

Aqui, `int` é o tipo de dado (número inteiro), `idade` é o nome da variável, e `25` é o valor que estamos guardando.

Nomes de Variáveis

Ao nomear suas variáveis, siga estas regras básicas:

- Use letras, números, underscores (_) e cifrões (\$).
- Não comece com um número.
- Use nomes significativos. Por exemplo, idade é melhor que x.

Exemplos:

Bom: altura, notaFinal, numeroDeEstrelas

Ruim: a, nf, 123idade

TIPOS DE DADOS

Tipos de Dados Comuns

Java tem vários tipos de dados que você pode usar, cada um adequado para diferentes tipos de informação.

- **Inteiros (int)**

Usado para números inteiros (sem decimais).

```
1 int idade = 25;  
2
```


- **Ponto Flutuante (double)**

Usado para números com decimais.

```
1 double preco = 19.99;  
2
```

- **Booleanos (boolean)**


Usado para valores verdadeiros ou falsos



```
1 boolean estaChovendo = false;  
2
```

- **Caracteres (char)**


Usado para um único caractere.



```
1 char inicial = 'A';  
2
```

- **Strings (String)**

Usado para sequências de caracteres, como palavras ou frases.



```
1 String saudacao = "Olá, Mundo!";  
2
```



Conclusão e agradecimentos



É HORA DE DAR TCHAU!

Ei, você conseguiu! 🎉 Chegamos ao fim do nosso ebook! Dá um tapinha nas costas e celebra essa vitória porque, sério, você mandou muito bem.

O que Aprendemos?

- **Boas-vindas ao Mundo Java:** Abrimos as portas para um novo universo.
- **Instalando Java:** Superamos o primeiro desafio de configuração.
- **Primeiros Passos com Java:** Criamos nosso primeiro "Olá, Mundo!" – o início de tudo.
- **Variáveis e Tipos de Dados:** Descobrimos como guardar e manipular todo tipo de informação, de números a textos.
- **Valeu Demais!**

Primeiro de tudo, um super obrigada por embarcar nessa jornada comigo! Foi incrível compartilhar essas dicas e truques com você. Programação pode parecer um quebra-cabeça infinito às vezes, mas você provou que pode montar cada peça e resolver qualquer desafio.
- **Continue Brilhando!**

A aventura está só começando. Cada linha de código que você escreve é um passo a mais na sua jornada de desenvolvedora. Não desanime com os erros (eles são nossos melhores professores!) e celebre cada pequena vitória. Java é só o começo – há um universo inteiro de linguagens e tecnologias esperando por você.

Mantenha o Contato!

- **Linkedin:** Pablynne Caldeira
- **GitHub:** pablynnecaldeira

Um Último Toque

Nunca pare de perguntar "por quê". Entender o motivo por trás do código é o que separa os bons programadores dos grandes mestres do código. Continue curioso, continue explorando e, acima de tudo, continue se divertindo.

Obrigada por ser uma companhia tão incrível nessa viagem. Continue codando, criando e surpreendendo o mundo!

Até a próxima, jovem ninja do Java!