

Métodos Void em Java: Os Bastidores da Programação

Bem-vindos ao mundo dos métodos void! Vamos descobrir como esses métodos sem retorno podem transformar seu código Java em algo mais organizado e poderoso.



por Angélica Karize Viecegli

VOVEL



VOID



METHOD

E - GREINTS



O que são Métodos Void?

1

Definição

Métodos void são funções que executam tarefas sem devolver valores. Eles fazem o trabalho e pronto!

2

Sintaxe Básica

Usamos a palavra reservada "void" antes do nome do método para indicar que não há retorno.

3

Propósito

Eles organizam operações e evitam repetição de código. São verdadeiros trabalhadores silenciosos!

JAVA VOID

Return Type

Peça a: >

Parameters

Curly-Brahclet

93 RITUR

78

14

12 -osenesters

Anatomia de um Método Void

1

Declaração

Começa com o modificador de acesso, seguido de "void" e o nome do método.

2

Parâmetros

Dados que o método recebe entre parênteses (opcional). São como ingredientes para uma receita!

3

Corpo

Código entre chaves {} que será executado quando o método for chamado.

Exemplos Simples de Métodos Void

Sem Parâmetros

```
void dizerOla() {  
    System.out.println("Olá,  
    mundo!");  
}
```

Com Parâmetros

```
void cumprimentar(String  
nome) {  
    System.out.println("Olá, " +  
    nome + "!");  
}
```

Múltiplos Parâmetros

```
void somar(int a, int b) {  
    System.out.println(a + b);  
}
```

Chamando Métodos Void

Definir o Método

Crie seu método void em algum lugar do código.

Chamar o Método

Digite o nome do método seguido de parênteses e ponto e vírgula.

Adicionar Argumentos

Se o método pedir parâmetros, coloque-os dentro dos parênteses.



Aplicações Práticas



Imprimir Relatórios

Métodos para formatar e exibir dados de maneira organizada.



Realizar Cálculos

Executar operações e mostrar resultados sem necessidade de retorná-los.



Criar Menus

Mostrar opções ao usuário em aplicações de terminal.



Configurar Sistema

Preparar o ambiente da aplicação antes da execução principal.





Métodos Void vs. Métodos com Retorno

Métodos Void	Métodos com Retorno
Não devolvem valores	Devolvem valores específicos
Usam palavra-chave "void"	Especificam tipo de retorno (int, String...)
Não usam "return" com valor	Precisam de "return" com valor
Chamados diretamente	Geralmente armazenam resultado em variável

Exemplo Completo: Jogo de Adivinhação

Método Principal
Coordena o jogo e chama outros métodos

Exibir Resultado
Mostra mensagem de vitória ou derrota



Mostrar Regras
Exibe instruções para o jogador

Processar Palpite
Verifica se o palpite está correto

```
void mostrarRegras() {  
    System.out.println("Adivinhe o número entre 1 e 10!");  
}
```

```
void processarPalpite(int palpite, int numero) {  
    if (palpite == numero) {  
        exibirResultado(true);  
    } else {  
        exibirResultado(false);  
    }  
}
```

```
void exibirResultado(boolean acertou) {  
    if (acertou) {  
        System.out.println("Parabéns! Você acertou!");  
    } else {  
        System.out.println("Que pena! Tente novamente.");  
    }  
}
```


Exemplo Completo: Implementação

Veja abaixo como implementar o jogo de adivinhação com uma classe main chamando os métodos void:

```
public class JogoAdivinhacao {

    public static void main(String[] args) {
        // Número secreto
        int numeroSecreto = 7;

        // Chamando método para mostrar regras
        mostrarRegras();

        // Simulando um palpite do usuário
        int palpiteUsuario = 7;
        System.out.println("Seu palpite: " + palpiteUsuario);

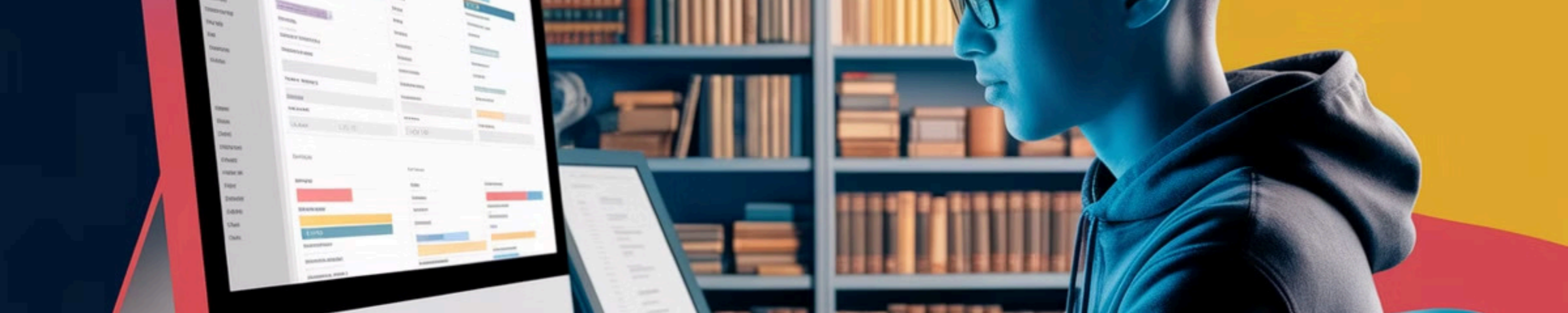
        // Chamando método para processar o palpite
        processarPalpite(palpiteUsuario, numeroSecreto);
    }

    static void mostrarRegras() {
        System.out.println("Adivinhe o número entre 1 e 10!");
    }

    static void processarPalpite(int palpite, int numero) {
        if (palpite == numero) {
            exibirResultado(true);
        } else {
            exibirResultado(false);
        }
    }

    static void exibirResultado(boolean acertou) {
        if (acertou) {
            System.out.println("Parabéns! Você acertou!");
        } else {
            System.out.println("Que pena! Tente novamente.");
        }
    }
}
```

Observe como a classe main chama os métodos void sem precisar armazenar nenhum valor de retorno, já que eles não retornam nada.



Exercício Prático: Sistema de Biblioteca

1 Crie estes métodos void

- `mostrarMenu()`
- `cadastrarLivro(String titulo, String autor)`
- `pesquisarLivro(String titulo)`
- `exibirTodosLivros()`

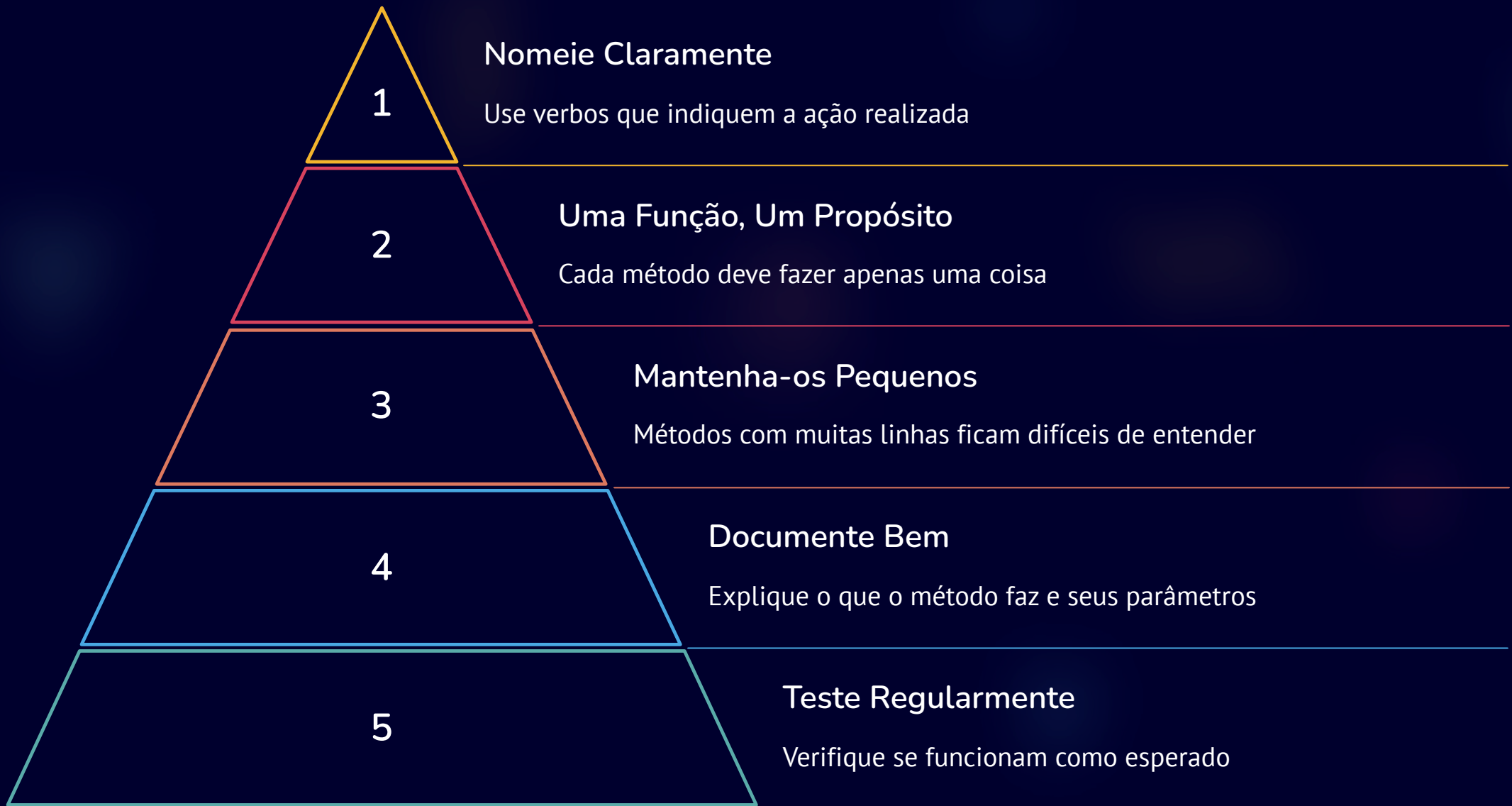
2 Implemente

Use arrays para armazenar os livros. Cada método deve realizar sua função específica sem retorno.

3 Teste seu sistema

Adicione alguns livros, pesquise-os e exiba a lista completa para verificar o funcionamento.

Dicas para Métodos Void de Sucesso



Agora você está pronto para organizar seu código com métodos void! Lembre-se: prática leva à perfeição. Bons estudos e boa programação!

Referências Bibliográficas

DEITEL, H. M.; DEITEL, P. J. **Java: como programar**. 10. ed. São Paulo: Pearson Education do Brasil, 2016.

SANTOS, R. **Introdução à programação orientada a objetos usando Java**. 2. ed. Rio de Janeiro: Elsevier, 2013.

SIERRA, K.; BATES, B. **Use a cabeça! Java**. 2. ed. Rio de Janeiro: Alta Books, 2010.

HORSTMANN, C. S. **Core Java: fundamentos**. São Paulo: Pearson Prentice Hall, 2010.

ORACLE. **Java Documentation: Methods**. Disponível em: <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>. Acesso em: 15 maio 2023.