



SENAI

FUNDAMENTOS DE JAVA

TÓPICOS ABORDADOS

- Funções:
 - Conceito.
 - Criação e estrutura básica.
- Invocação.
- Observações.





FUNDAMENTOS DE JAVA: **FUNÇÕES**



FUNDAMENTOS DE JAVA

CONCEITO DE FUNÇÕES



- Uma **função** ou **método** em Java é um bloco de código que executa uma tarefa específica e pode ser **reutilizado** em diferentes partes do programa.
- A principal ideia por trás de uma função é evitar repetição de código, **organizar a lógica** do programa e **facilitar a manutenção**.

FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA

A estrutura básica de uma função em Java inclui:

1. Modificadores de acesso (opcional):

- Define quem pode acessar o método (ex.: **public**, **private**).
- Em Java o modificador de acesso padrão é o **public** e pode ser omitido.

```
public static void funcaoSemParametroSemRetorno() {  
    System.out.println("teste");  
}
```

FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA

A estrutura básica de uma função em Java inclui:

2. Palavra-chave static:

- Associa as funções (método) ou variáveis diretamente à classe, e não a um **objeto instanciado** dessa classe.
- Significa poder acessar a esses membros diretamente pela classe.

```
public static void funcaoSemParametroSemRetorno() {  
    System.out.println("teste");  
}
```

FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA

A estrutura básica de uma função em Java inclui:

3. Nome da função:

- Deve ser descritivo e seguir as convenções de nomenclatura (ex.: **calcularMedia**).

```
public static void funcaoSemParametroSemRetorno() {  
    System.out.println("teste");  
}
```


FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA

A estrutura básica de uma função em Java inclui:

4. Parâmetros (opcional):

- Valores que a função pode receber para trabalhar (ex.: **funcao(int a, double b)**).
- Exemplo de função sem parâmetros: **funcao()**.

```
static void funcaoComParametrosSemRetorno(int parametro1, String parametro2) {  
    System.out.println("parametro1 = " + parametro1);  
    System.out.println("parametro2 = " + parametro2);  
}
```

A diagram illustrating the structure of a function signature. A red rectangle highlights the parameter list "(int parametro1, String parametro2)". Two red arrows point from the text "parametro1" and "parametro2" in the function body to their respective declarations in the parameter list.

FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA

A estrutura básica de uma função em Java inclui:

5. Tipo de retorno:

- Especifica o tipo de dado que a função retorna (ex.: **int**, **void**, **double**).
- Nesse caso é obrigatório a palavra-chave ***return***.
- Se a função não retornar nada, usamos **void**.

```
private static int funcaoComParametrosComRetorno(int parametro1, int parametro2) {  
    return parametro1 + parametro2;  
}
```

A red arrow points from the bottom left towards the **int** return type in the code snippet.

FUNDAMENTOS DE JAVA



ESTRUTURA DE UMA FUNÇÃO EM JAVA


A estrutura básica de uma função em Java inclui:

6. Corpo da função:

- O código que realiza a tarefa definida pela função.

```
private static int funcaoComParametrosComRetorno(int parametro1, int parametro2) {  
    return parametro1 + parametro2;  
}
```

A diagram illustrating the body of a Java function. A dark gray rectangular box contains the code snippet. Two red arrows point to the curly braces: one points to the opening brace '{' at the end of the first line, and the other points to the closing brace '}' at the end of the third line.



FUNDAMENTOS DE JAVA: **INVOCACÃO DE FUNÇÕES (MÉTODOS)**



FUNDAMENTOS DE JAVA



INVOCAÇÃO DE FUNÇÃO SEM RETORNO E SEM PASSAGEM DE PARÂMETROS

```
public class Funcoes { new *  
    public static void main(String[] args) { new *  
        funcaoSemParametroSemRetorno();  
    }  
  
    public static void funcaoSemParametroSemRetorno() { 1 usage new *  
        System.out.println("teste");  
    }  
}
```

FUNDAMENTOS DE JAVA



INVOCAÇÃO DE FUNÇÃO SEM RETORNO E COM PASSAGEM DE PARÂMETROS

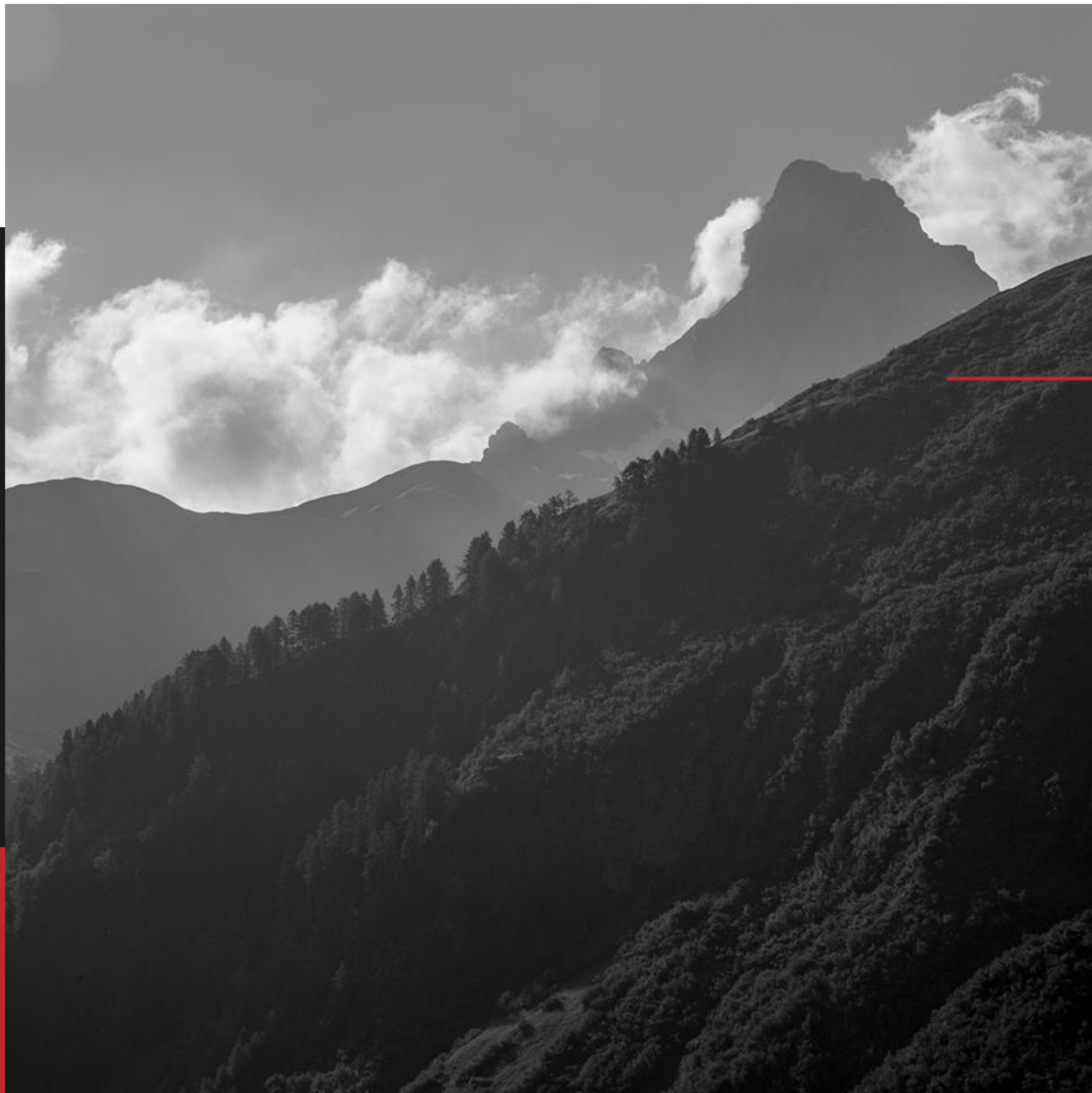
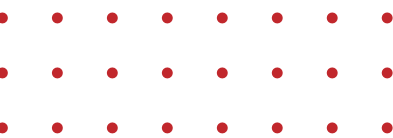
```
public class Funcoes { new *  
  
    public static void main(String[] args) { new *  
        funcaoComParametrosSemRetorno(1, "texto");  
    }  
  
    static void funcaoComParametrosSemRetorno(int parametro1, String parametro2) {  
        System.out.println("parametro1 = " + parametro1);  
        System.out.println("parametro2 = " + parametro2);  
    }  
}
```

FUNDAMENTOS DE JAVA



INVOCAÇÃO DE FUNÇÃO COM RETORNO E COM PASSAGEM DE PARÂMETROS

```
public class Funcoes { new *  
  
    public static void main(String[] args) { new *  
        int soma = funcaoComParametrosComRetorno(2, 2);  
        System.out.println("retorno da função: " + soma);  
    }  
  
    static int funcaoComParametrosComRetorno(int parametro1, int parametro2) {  
        return parametro1 + parametro2;  
    }  
  
}
```

FUNDAMENTOS DE JAVA

EXERCÍCIOS:

CONCEITO DE **FUNÇÕES:**

- CRIAÇÃO
- INVOCACÃO





MÃOS A **OBRA**

Use tudo o que você aprendeu para resolver os exercícios. Concentre-se no objetivo principal de cada um.

Se a sua solução funcionar como o exercício pede, que tal deixar tudo ainda mais legal? Use a sua criatividade para deixar o código mais intuitivo e fácil de entender. Organize as informações e melhore os textos que aparecem para o usuário.

Lembre-se: Faça com calma, peça ajuda quando precisar entender melhor alguma coisa. O importante é aprender e treinar a sua lógica de programação. Não vale copiar a resposta pronta! Esses exercícios são a sua chance de praticar e se tornar um programador ainda melhor. "

SENAI



FUNÇÕES:

EXERCÍCIOS



Exercício 1: Função que Imprime uma Mensagem

Crie uma função chamada *imprimirSaudacao* que imprime "Bem-vindo ao sistema!". Chame essa função no método main.

Exercício 2: Função que Soma Dois Números

Crie uma função chamada *somar* que receba dois números inteiros como parâmetros e retorne a soma deles. Exiba o resultado no método main.

FUNÇÕES:

EXERCÍCIOS



Exercício 3: Função para Calcular o Dobro de um Número

Crie uma função chamada ***dobrar*** que receba um número inteiro e retorne o dobro dele. Exiba o resultado no método main.

Exercício 4: Função para Converter Graus Celsius para Fahrenheit

Crie uma função chamada ***converterCelsiusParaFahrenheit*** que receba um valor em graus Celsius e retorne a conversão para Fahrenheit. A fórmula é: $F = C * 1.8 + 32$.

Exercício 5: Função para Calcular a Média de Três Notas

Crie uma função chamada *calcularMedia* que receba três notas de um aluno (do tipo double) e retorne a média delas. Exiba o resultado no método main.

Exercício 6: Função para Verificar se um Número é Par

Crie uma função chamada *verificarPar* que receba um número inteiro e retorne true se o número for par, ou false se for ímpar.

FUNÇÕES:

EXERCÍCIOS



Exercício 7: Função para Determinar o Maior de Três Números
Crie uma função chamada *maiorNumero* que receba três números inteiros como parâmetros e retorne o maior deles.

Exercício 8: Função que Calcula o Fatorial de um Número
Crie uma função chamada *calcularFatorial* que receba um número inteiro positivo e retorne o fatorial desse número.

FUNÇÕES:

EXERCÍCIOS



Exercício 9: Função para Calcular a Potência de Um Número

Crie uma função chamada *calcularPotencia* que receba dois números inteiros, base e expoente, e retorne o resultado de $\text{base}^{\text{expoente}}$ (base elevado ao expoente).

Exercício 10: Função para Calcular o IMC

Crie uma função chamada *calcularIMC* que receba o peso (em double) e a altura (em double) de uma pessoa, e retorne o valor do Índice de Massa Corporal (IMC). A fórmula é: $\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$.

FUNÇÕES:

DESAFIO: FUNÇÕES



Sistema de Cadastro com Menu e Funções.

Baseando-se no desafio da última aula, onde criamos um sistema de cadastro utilizando uma matriz para armazenar dados, agora vamos expandir o programa para torná-lo mais completo e fácil de manter.

FUNÇÕES:

DESAFIO: FUNÇÕES



Sistema de Cadastro com Menu e Funções.

Objetivo: Adapte o programa para incluir um **menu de opções**, permitindo que o usuário escolha as seguintes ações:

- 1. Cadastrar Usuário:** Adicione um novo usuário à matriz.
- 2. Consultar Usuários:** Exiba todos os usuários cadastrados em formato de tabela, incluindo o cabeçalho.

FUNÇÕES:

DESAFIO: FUNÇÕES



Sistema de Cadastro com Menu e Funções.

Objetivo: Adapte o programa para incluir um **menu de opções**, permitindo que o usuário escolha as seguintes ações:

3. Atualizar Usuário: Permita a modificação dos dados de um usuário já cadastrado.

4. Deletar Usuário: Remova um usuário da lista.

FUNÇÕES:

DESAFIO: FUNÇÕES



Requisitos:

- **Modularização:** Utilize o conceito de funções para organizar o código. Cada opção do menu deverá ser implementada em uma função separada, o que facilitará a manutenção e o entendimento do código.
- **Reutilização de Código:** Sempre que possível, reaproveite partes do código já escritas.
- **Boas Práticas:** Use todos os recursos que aprendemos até o momento, como tipo de dados, estruturas de decisão, repetição e vetores/matrizes.



SENAI

DEPARTAMENTO REGIONAL
DE SÃO PAULO

www.sp.senai.br