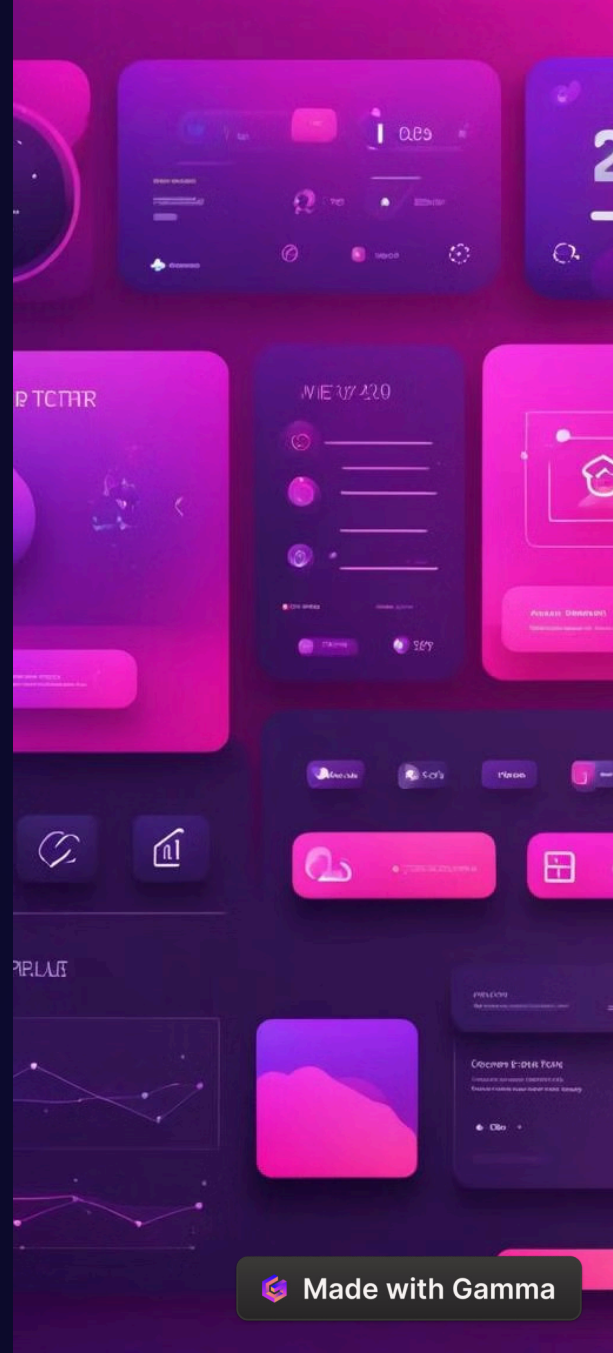


GRUPO DE ESTUDO - JAVA(LÓGICA DE PROGRAMAÇÃO)



Objetivo dos Grupos de Estudo

Ambiente Colaborativo

Criar um espaço para o aprendizado de tecnologia, promovendo o compartilhamento de conhecimentos e experiências entre os alunos.

Comunicação Eficaz

Canal exclusivo para troca de informações e apoio, facilitando a comunicação entre os participantes.

Organização e Estrutura

Orientações para a organização dos encontros, garantindo um fluxo estruturado e eficiente.

Aprendizado Prático

Encontros para compartilhar experiências e receber feedbacks, promovendo o aprendizado prático e a troca de ideias.

HEBERT BARBOSA

1

Desenvolvedor Backend na **WEG Motores**

2

Natural de Araraquara-SP, atualmente em **Joinville-SC**

3

Antes da tecnologia, minha paixão era a **medicina veterinária**

SUA VEZ!



Nome

- 1 — **Profissão atual**
- 2 — **De onde você é**
- 3 — **Curiosidade**

Configurando seu ambiente Java

1. Baixe um JDK. Escolha uma versão Java(21 ou 23) compatível com o seu sistema operacional.
2. Crie uma variável **\$JAVA_HOME**. Adicione **\$JAVA_HOME/bin** ao seu **\$Path** nas variáveis de ambiente.
3. Instale um IDE. Recomendo o IntelliJ Community.

Entradas e Saídas

Saídas

O mais comum para exibir mensagens ou resultados no terminal é o uso de **`System.out.println(expression);`**, mas existem outras opções disponíveis.

Entradas

O **input** permite que o programa leia valores do console, tornando-o mais dinâmico.

- **`String[] args`**: Utilizado para ler argumentos passados ao programa na linha de comando.
- **`java.util.Scanner`**: Uma classe que permite ler valores do console. Por exemplo, para ler um valor do tipo **`double`**, usamos o método **`nextDouble()`**.

Tipos, Variáveis e Constantes em Java

Tipos Primitivos

Java é **fortemente tipada**. Cada variável deve ter um tipo definido.

- Inteiros: **byte, short, int, long**
- Decimais: **float, double**
- Caractere: **char**
- Lógico: **boolean**

Variáveis

Variáveis armazenam valores. Use **camelCase** para nomes compostos.

Exemplo: **int idade = 22;**

Exemplo **camelCase**: **String nomeCompleto = "João Silva";**

Java atribui um valor padrão se não forem inicializadas.

Constantes

Constantes armazenam valores fixos. Declaradas com **final**, em letras maiúsculas e underscores.

Exemplo: **final int MAXIMO_TENTATIVAS = 3;**

Constante existente: **Math.PI**

Operadores em Java

Operadores Aritméticos

- Soma (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/)
- Mod (%)

Realizam cálculos matemáticos com variáveis.

Operadores de Comparação

- Maior que (>)
- Maior ou igual que (>=)
- Menor que (<)
- Menor ou igual que (<=)
- Igual (==)
- Diferente (!=)

Comparam valores.

Operadores Lógicos

- Negação (!)
- E (&&)
- OU (||)

Avaliam expressões booleanas.

A precedência define a ordem de execução dos operadores. Consulte a tabela para evitar surpresas!

Estruturas de Controle de Fluxo

Estruturas Condicionais

Use **if**, **else if** e **else** para executar blocos de código com base em condições.

Avalie expressões booleanas para controlar o fluxo de execução.

Instrução Switch

A instrução **switch** escolhe caminhos de código com base no valor de uma variável.

Ela suporta os tipos primitivos, Wrapper Classes e Enums, como `byte`, `short`, `char`, `int`, `String` e `Enum`.

Estruturas de Repetição - For e While

While Loop

Os loops **while/do while** executam um bloco de código. Isso é feito com base em uma condição booleana.

Ele se repete enquanto a condição for **verdadeira**.

For Loop

O loop **for** serve para iterar um número conhecido de vezes.

É especialmente útil para arrays.

Arrays e Matrizes

Arrays

Arrays são como uma fileira de gavetas. Cada um armazena um valor com um índice numérico.

- Arrays são úteis para armazenar listas de dados.
- O índice começa em 0.

Matrizes

As Matrizes são como uma tabela com linhas e colunas.

- Matrizes são úteis para armazenar tabela de dados.
- Cada elemento é acessado por dois índices, linha e coluna

Mão na massa!

1

Exibição na Tela

Aprender a exibir mensagens e resultados utilizando **`System.out.println()`**.

3

Operadores de Controle de Fluxo

Aplicar operadores de controle de fluxo (**`if`**, **`else`**) para criar programas que tomam decisões.

5

Agora sua vez!

Desafio: Não tenha medo, mostre que você domina o código!

Tarefa: Crie um programa que exiba os números de 1 a 10 e, para cada número, indique se é par ou ímpar.

2

Criação de Variáveis

Entender como declarar e inicializar variáveis para armazenar e manipular dados.

4

Estruturas de Repetição

Aprender a usar estruturas de repetição (**`for`**, **`while`**) para executar blocos de código múltiplas vezes.

6

Desafios de Fixação

Vamos realizar exercícios para consolidar o aprendizado.

Desafio 1: Validação de Elegibilidade para o Programa Joinville Mais Tec



Objetivo

Crie um programa que verifique se o usuário é elegível para o programa **Joinville Mais Tec**, com base em sua **idade** e **cidade**.



Descrição do Desafio

- O programa deve pedir ao usuário para informar sua **idade** e **cidade**.
- O programa deve verificar se o usuário **tem 16 anos ou mais** e se mora em **Joinville**.
- Se ambas as condições forem verdadeiras, o programa deve informar que o usuário **está apto** a participar.
- Se alguma das condições não for atendida, o programa deve informar que o usuário **não está apto**.



Entrada

O programa deve pedir ao usuário para **informar sua idade** e a **cidade** onde mora.



Saída Esperada

Quando o usuário digitar a idade e a cidade, o programa deve verificar se ele é **apto** ou **não apto**.

Exemplo de saída para o usuário com 17 anos e morador de Joinville:

Digite sua idade: 17

Digite sua cidade: Joinville

Você está apto a participar do Programa Joinville Mais Tec!

Exemplo de saída para o usuário com 15 anos e morador de outra cidade:

Digite sua idade: 15

Digite sua cidade:

Florianópolis

Você não está apto a participar do Programa Joinville Mais Tec.

Desafio 2: Soma de Números



Objetivo

Crie um programa que leia uma quantidade de números digitados pelo usuário e calcule a soma de todos os números informados. Utilize uma **estrutura de repetição** para ler os números e somá-los.



Descrição do Desafio

- O programa deve pedir ao usuário para informar a **quantidade de números** que deseja somar.
- Em seguida, o programa deve **ler cada número** fornecido pelo usuário e **adicionar** esse número à soma total.
- Ao final, o programa deve **exibir o resultado da soma** de todos os números fornecidos.



Entrada

- O programa deve pedir ao usuário para **informar a quantidade de números** que ele deseja somar.
- Depois, o programa vai ler **cada número individualmente** e somá-los.



Saída Esperada

O programa deve exibir o resultado da **soma total** dos números fornecidos.

Exemplo de saída para a soma de 3 números (**5, 10, 20**):

Digite a quantidade de números que você deseja somar: 3

Digite o número 1: 5

Digite o número 2: 10

Digite o número 3: 20

A soma total é: 35

Desafio 3: Tabuada de um Número



Objetivo

Crie um programa que **calcule e exiba a tabuada** de um número fornecido pelo usuário. Utilize uma **estrutura de repetição** para realizar as multiplicações.



Descrição do Desafio

O programa deve pedir ao usuário para **informar um número**.

O programa deve então **exibir a tabuada** desse número, de **1 até 10**.

A tabuada deve ser exibida de forma clara, por exemplo, "Número X multiplicado por Y é igual a Z".



Entrada

- O programa deve pedir ao usuário para **informar o número** para o qual a tabuada será calculada.



Saída Esperada

O programa deve exibir a **tabuada** do número fornecido de **1 até 10**.

Exemplo de saída para o número **5**:

Digite um número: 5

Tabuada de 5:

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

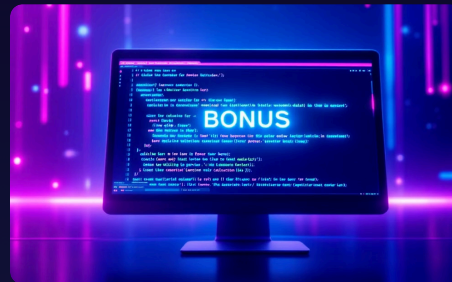
$$5 \times 10 = 50$$

Desafio 4: Cálculo de Bônus Anual para Funcionários



Objetivo

Crie um programa que calcule o **bônus salarial** anual de um funcionário, aplicando um aumento no bônus a cada ano de trabalho. O bônus inicial é de **R\$1000,00**, e a cada ano o valor do bônus **umenta R\$200,00**. O programa deve calcular o **total acumulado** de bônus ao longo dos anos e exibir esse total ao final.



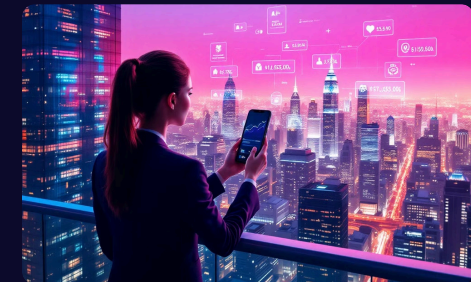
Descrição do Desafio

- O **bônus inicial** é de **R\$1000,00**.
- A cada ano trabalhado, o bônus **umenta R\$200,00**.
- O programa deve calcular e mostrar o bônus **acumulado a cada ano**.
- Ao final, o programa deve exibir o **total de bônus acumulado** após o número de anos informados pelo usuário.



Entrada

O programa deve pedir ao usuário para **informar o número de anos** que ele trabalha na empresa.



Saída Esperada

Quando o usuário digitar a quantidade de anos (por exemplo, **5 anos**), o programa deve exibir o cálculo do bônus acumulado ano a ano. Exemplo de saída para 5 anos:

Digite o número de anos que você trabalha na empresa: 5

Ano 1: Bônus acumulado = R\$1000

Ano 2: Bônus acumulado = R\$2200

Ano 3: Bônus acumulado = R\$3600

Ano 4: Bônus acumulado = R\$5200

Ano 5: Bônus acumulado = R\$7000

Total de bônus após 5 anos:
R\$7000

Sugestões para as Próximas Etapas

Programação Orientada à Objetos

Aprender a modelar entidades do mundo real como objetos. Entender classes, herança e polimorfismo.

Testes

Aprender testes unitários para garantir que o código esteja funcionando corretamente e de forma eficiente.

Versionamento de Código (Git)

Compartilhe seu código com outras pessoas.

Protocolo HTTP

Compreender como o protocolo HTTP funciona é essencial para trabalhar com APIs.

Sua opinião é muito importante!

Tem algo em mente que gostaria de explorar em nossos estudos? Compartilhe suas ideias!



Próxima Aula

O assunto a ser abordado será:

Programação Orientada à Objetos

Aprender a modelar entidades do mundo real como objetos. Entender classes, herança e polimorfismo e tirar dúvidas no geral.

Links de Referências

- [Java Tutorial - Println](#): Entenda como imprimir no console.
- [Baeldung - Scanner](#): Aprenda sobre como obter dados do usuário.
- [Java Tutorial - Números e Strings](#): Domine tipos de dados e strings.
- [Java Tutorial - Autoboxing](#): Compreenda o conceito de autoboxing em Java.
- [Java Tutorial - Operadores](#): Estude os operadores Java.
- [Java Tutorial - Condicionais](#): Declarações condicionais explicadas.
- [Controle de fluxo - parte 1](#): Uma introdução ao controle de fluxo.
- [Java Tutorial - While](#): Explore o loop while.
- [Java Tutorial - For](#): O essencial do loop for.
- [Estruturas condicionais e de repetição](#): Mergulho profundo em estruturas de loop.