

Desenvolvimento de Programas Estruturados e Modularização

**UNIVERSIDADE DO VALE DO SAPUCAÍ – UNIVAS – POUSO ALEGRE/MG
PROFESSOR: RAFFAEL CARVALHO**

Aula Prática Fixação de Funções

Exercício 1: O Dobro

- **Missão:** Crie uma função chamada `calcularDobro`.
- **Entrada (Parâmetro):** Ela deve receber um número inteiro (`int`).
- **Saída (Retorno):** Ela deve devolver um número inteiro que seja o dobro do valor recebido.
- **No `main`:** Peça para o usuário digitar um número, chame a função e imprima o resultado.

Exercício 2: Calculadora de Área

- **Missão:** Crie uma função chamada `calcularAreaRetangulo`.
- **Entradas (Parâmetros):** Ela deve receber dois valores decimais (`double`): a `base` e a `altura`.
- **Saída (Retorno):** Ela deve devolver um `double` contendo a área (`base * altura`).
- **No main:** Teste sua função passando valores fixos ou lidos pelo teclado.

Exercício 3: Mensagem de Boas-vindas (void)

- **Missão:** Crie uma função chamada `exibirBoasVindas`.
- **Entrada (Parâmetro):** Ela deve receber uma `String` chamada `nome`.
- **Saída (Retorno):** Nenhum (`void`).
- **O que ela faz:** Esta função tem permissão para usar o `System.out.println`! Ela deve imprimir na tela: "Olá [nome], seja muito bem-vindo ao sistema!"
- **No main:** Leia o nome do usuário e chame a função.

Exercício 4: Par ou Ímpar?

- **Missão:** Crie uma função chamada `verificarParidade`.
- **Entrada (Parâmetro):** Recebe um número inteiro (`int`).
- **Saída (Retorno):** Devolve um texto (`String`).
- **Regra:** Se o número for divisível por 2 (use o operador `%`), a função retorna a palavra "Par". Caso contrário, retorna "Ímpar".

Exercício 5: O Maior de Dois

- **Missão:** Crie uma função chamada `descobrirMaior`.
- **Entradas (Parâmetros):** Recebe dois números inteiros (`int a, int b`).
- **Saída (Retorno):** Devolve o maior número entre os dois (`int`).
- **Regra:** Use um `if/else` dentro da função para comparar os parâmetros. Se eles forem iguais, pode retornar qualquer um deles.

Exercício 6: Validador de Maioridade

- **Missão:** Crie uma função chamada `isMaiorDeIdade`.
- **Entrada (Parâmetro):** Recebe a idade da pessoa (`int`).
- **Saída (Retorno):** Devolve um valor Verdadeiro ou Falso (`boolean`).
- **Regra:** Retorna `true` se a idade for maior ou igual a 18, e `false` caso contrário. Teste no `main` imprimindo "Pode entrar" ou "Acesso negado" baseado no retorno.

Exercício 7: Aprovador de Empréstimo

- **Missão:** Crie uma função aprovarEmprestimo.
- **Entradas (Parâmetros):** Salário do cliente (double) e o valor da parcela do empréstimo (double).
- **Saída (Retorno):** Verdadeiro ou Falso (boolean).
- **Regra:** Um empréstimo só é aprovado (retorna true) se o valor da parcela não ultrapassar 30% do salário do cliente. Caso contrário, retorna false.

Exercício 8: Calculadora de Tarifa de Água

- **Missão:** Crie uma função chamada `calcularContaAgua`.
- **Entrada (Parâmetro):** O consumo do mês em metros cúbicos (`double`).
- **Saída (Retorno):** O valor total da conta em reais (`double`).
- **Regras de Negócio (Use if/else if):**
 - Até 10 m³: Taxa mínima fixa de R\$ 35,00.
 - De 11 m³ a 20 m³: R\$ 5,00 por metro cúbico consumido.
 - Acima de 20 m³: R\$ 7,00 por metro cúbico consumido.

Exercício 9: Conversor de Tempo

- **Missão:** Crie uma função chamada `formatarSegundos`.
- **Entrada (Parâmetro):** Um tempo total dado apenas em segundos (`int`). Exemplo: 3665.
- **Saída (Retorno):** Um texto (`String`) formatado.
- **Regra:** A função deve converter esse valor gigante de segundos em horas, minutos e segundos restantes.
 - Exemplo de retorno: "1 horas, 1 minutos e 5 segundos".
 - Dica matemática: Lembre-se que 1 hora tem 3600 segundos, e 1 minuto tem 60 segundos. Use divisão `/` e o resto da divisão `%`.

Bibliografia Básica

- FURGERI, Sérgio. Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações. Editora Saraiva, 2015. E-book. ISBN 9788536519340. Disponível em:
- FINEGAN, Edward; LIGUORI, Robert. OCA Java SE 8: guia de estudos para o exame 1Z0-808. Grupo A, 2018. E-book. ISBN 9788582604779. Disponível em:
- FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação. São Paulo: Makron Books, 2000.