

Lista de Exercícios – Criação de Classes e Pacotes em Java

Exercício 1: Controle de Conta Bancária

Descrição:

Crie um programa que simule uma conta bancária, permitindo depósitos e saques.

Requisitos:

1. Criar o pacote dominio contendo a classe ContaBancaria:
 - Atributos: titular (String), saldo (double).
 - Métodos:
 - depositar(double valor): adiciona um valor ao saldo.
 - sacar(double valor): subtrai um valor do saldo, se houver saldo suficiente.
 - exibirSaldo(): exibe o saldo atual.
 2. Criar o pacote controle contendo a classe GerenciadorConta:
 - Permitir que o usuário cadastre uma conta e realize operações (depositar, sacar, consultar saldo).
 - Utilizar um while para continuar as operações até o usuário escolher sair.
-

Exercício 2: Cálculo de Salário com Bônus

Descrição:

Crie um programa que calcule o salário final de um funcionário, considerando um bônus de acordo com sua categoria.

Requisitos:

1. Criar o pacote dominio contendo a classe Funcionario:
 - Atributos: nome (String), salarioBase (double), categoria (String).
 - Método:
 - calcularSalarioFinal(): adiciona um bônus ao salário base:
 - "A" → +20%
 - "B" → +10%

- "C" → sem bônus

2. Criar o pacote controle contendo a classe GerenciadorFuncionario:

- Permitir que o usuário cadastre um funcionário e consulte seu salário final.
- Utilizar switch-case para aplicar a regra do bônus.

Exercício 3: Simulação de Um Elevador

Descrição:

Implemente um programa que simule o funcionamento básico de um elevador, permitindo subir, descer e exibir o andar atual.

Requisitos:

1. Criar o pacote dominio contendo a classe Elevador:

- Atributos: andarAtual (int), totalAndares (int).
- Métodos:
 - subir(): aumenta o andar atual, se não for o último.
 - descer(): reduz o andar atual, se não for o térreo.
 - exibirAndar(): exibe o andar atual.

2. Criar o pacote controle contendo a classe SimuladorElevador:

- Permitir que o usuário suba e desça os andares de acordo com sua escolha.
- Garantir que o elevador não ultrapasse os limites (térreo e último andar).
- Utilizar um do-while para manter a simulação em execução até o usuário escolher sair.

Exercício 4: Validação de Nota de um Aluno

Descrição:

Crie um programa que leia uma nota de um aluno e verifique sua validade. Se for válida, classifique o aluno conforme sua média.

Requisitos:

1. Criar o pacote dominio contendo a classe Aluno:

- Atributos: nome (String), nota (double).
- Métodos:
 - validarNota(): verifica se a nota está entre 0 e 10.
 - classificarAluno(): classifica o aluno:
 - Nota $\geq 7 \rightarrow$ "Aprovado"
 - Nota ≥ 5 e $< 7 \rightarrow$ "Recuperação"
 - Nota $< 5 \rightarrow$ "Reprovado"

2. Criar o pacote controle contendo a classe GerenciadorAluno:

- Solicitar o nome e a nota do aluno e verificar sua validade antes de exibir a classificação.
- Utilizar um while para garantir que o usuário informe uma nota válida.

Exercício 5: Simulação de Um Caixa Eletrônico

Descrição:

Crie um programa que simule um caixa eletrônico, permitindo saques apenas em múltiplos de R\$ 10 e garantindo que o saldo seja suficiente.

Requisitos:

1. Criar o pacote dominio contendo a classe CaixaEletronico:

- Atributos: saldo (double).
- Métodos:
 - sacar(double valor): verifica se o valor é múltiplo de 10 e se há saldo suficiente.
 - exibirSaldo(): exibe o saldo restante.

2. Criar o pacote controle contendo a classe SimuladorCaixa:

- Permitir que o usuário realize múltiplos saques, verificando a validade do valor.
- Utilizar um while para continuar as operações até o usuário decidir sair.