

Exercício 1: Classe Pessoa

Objetivo: Praticar criação de atributos privados e métodos de acesso.

Descrição:

Crie uma classe `Pessoa` com os atributos privados:

- `nome` (String)
- `idade` (int)

Implemente os métodos:

- `getNome()` e `setNome(String nome)`
- `getIdade()` e `setIdade(int idade)`

Crie uma classe principal (`Main`) que:

- Cria um objeto da classe `Pessoa`
 - Define nome e idade
 - Imprime os dados no console
-

Exercício 2: Conta Bancária

Objetivo: Praticar encapsulamento com validação.

Descrição:

Crie uma classe `ContaBancaria` com os atributos privados:

- `numeroConta` (String)
- `saldo` (double)

Métodos:

- `getNumeroConta()` e `setNumeroConta(String numeroConta)`

- `getSaldo()` (não criar `setSaldo`)
- `depositar(double valor)` (só aceita valores positivos)
- `sacar(double valor)` (só permite sacar se houver saldo)

Na classe `Main`, simule:

- Um depósito
- Um saque válido
- Um saque inválido

Exercício 3: Produto em Estoque

Objetivo: Encapsular atributos e proteger regras de negócio.

Descrição:

Crie uma classe `Produto` com:

- `nome` (String)
- `preco` (double)
- `quantidadeEstoque` (int)

Todos os atributos devem ser privados. Crie:

- Getters e setters apropriados
- Métodos `adicionarEstoque(int qtd)` e `removerEstoque(int qtd)` (verifique se a remoção é possível)

No `Main`, crie um produto, adicione e remova estoque, e exiba as informações atualizadas.

Exercício 4: Aluno e Média

Objetivo: Encapsular lógica de negócio em métodos.

Descrição:

Crie a classe `Aluno` com atributos:

- `nome` (String)
- `nota1`, `nota2` (double)

Apenas `nome`, `nota1`, `nota2` devem ser setados via setters.

Crie o método:

- `getMedia()` que retorna a média das notas
- `isAprovado()` que retorna `true` se a média for ≥ 6.0

No `Main`, crie alunos com diferentes notas e mostre o nome, média e se foi aprovado.

Exercício 5: Livro com Faixa Etária

Objetivo: Usar encapsulamento para controlar acesso a dados sensíveis.

Descrição:

Crie uma classe `Livro` com:

- `titulo` (String)
- `autor` (String)
- `faixaEtariaMinima` (int)

Crie um método `podeSerLidoPor(int idadeLeitor)` que retorna `true` se a idade for maior ou igual à faixa etária mínima.

No `Main`, crie alguns livros e teste se diferentes idades podem lê-los.