# Qual é a principal função do encapsulamento em Java?

a) Permitir que os atributos de uma classe sejam acessados diretamente por outras classes.

b) Ocultar os detalhes internos de uma classe e fornecer métodos públicos para manipulação segura dos dados.

c) Criar várias instâncias de uma classe com diferentes valores para seus atributos.

d) Estabelecer herança entre classes distintas.

# Qual é a saída do seguinte código se a variável nota tiver valor 5?

**int** nota = 5;

**if** (nota >= 6) {

System.***out***.println("Aprovado");

} **else** **if** (nota >= 4) {

System.***out***.println("Recuperação");

} **else** {

System.***out***.println("Reprovado");

}

a) Aprovado

b) Recuperação

c) Reprovado

d) Erro de compilação

# Qual é a saída correta do seguinte código?

**for** (**int** i = 1; i <= 5; i++) {

System.***out***.print(i + " ");

}

a) 0 1 2 3 4

b) 1 2 3 4 5

c) 1 2 3 4

d) 2 3 4 5 6

# Identifique o erro no seguinte código e sugira uma solução:

**public** **class** Pessoa {

String nome;

**int** idade;

**public** **void** exibirDados() {

System.***out***.println("Nome: " + nome);

System.***out***.println("Idade: " + Idade);

}

}

Qual é o erro neste código?

a) A variável Idade não está declarada corretamente.

b) O método exibirDados() deve ser declarado como static.

c) Os atributos devem ser declarados como private.

d) O método exibirDados() está faltando um retorno return.

# Com base no seguinte diagrama UML, implemente a classe correspondente. Você não precisa criar getters, setters, construtores ou realizar verificações de saldo.



public class ContaBancaria {