	PUC-MG: Pontifícia Universidade Católica de Minas Gerais	
	Curso: Engenharia de Software	
	Disciplina: Programação Modular	
	Professor(a): Pedro Pongelupe Lopes	
	Semestre: 2024.1	
	Aluno:	Matrícula:

Exercício de Revisão Prova 1

Questão 1 Uma conta bancária é identificada pelo seu número de 5 dígitos e pelo CPF de seu proprietário. As contas bancárias permitem operações de saque e depósito. A operação de saque tem uma regra especial, permitindo que o cliente faça um saque acima do seu saldo – o que é chamado de limite. Por exemplo, se uma conta tem saldo de R\$200 e limite de R\$100, seu cliente pode sacar até R\$300. O saldo fica negativo e, no próximo depósito, a conta cobrará uma taxa de 3% sobre este valor negativo, descontando do próprio valor de depósito.

Para a situação apresentada, modele uma classe utilizando UML e implemente seu código em Java. É aconselhado testar os métodos implementados para ter certeza do seu funcionamento.

Resposta

Conta
+ <u>TAXA_CHEQUE_ESPECIAL</u> : double = 0.03 – identificador: String – cpf: String – saldo: double – limite: double
+ sacar(valor: double): double + depositar(valor: double): double + getSaldo(): double

```
public class Conta {

    private static final double TAXA_CHEQUE_ESPECIAL = 0.03;

    private String identificador;
    private String cpf;
    private double saldo;
    private double limite;

    public Conta(String identificador, String cpf, double saldo, double limite) {
        this.identificador = identificador;
        this.cpf = cpf;
        this.saldo = saldo;
        this.limite = limite;
    }

    public double sacar(double valor) {
        if (valor <= saldo + limite) {
            this.saldo -= valor;
        } else {
            valor = -1;
        }
        return valor;
    }

    public double depositar(double valor) {
        if (saldo < 0) {
```

```

        this.saldo += valor + (saldo * TAXA_CHEQUE_ESPECIAL);
    } else {
        this.saldo += valor;
    }
    return saldo;
}

    public double getSaldo() {
        return this.saldo;
    }
}

class ContaTest {

    private Conta conta;

    @BeforeEach
    public void setUp() {
        this.conta = new Conta("12345", "algum cpf", 200, 100);
    }

    @Test
    public void testSacarSemChequeEspecial() {
        conta.sacar(50d);
        assertEquals(150d, conta.getSaldo());
    }

    @Test
    public void testSacarComChequeEspecial() {
        conta.sacar(250d);
        assertEquals(-50d, conta.getSaldo());
    }

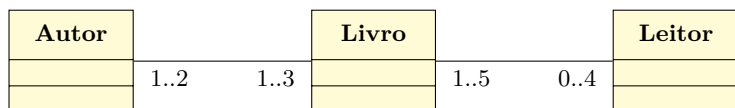
    @Test
    public void testSacarValorInvalido() {
        conta.sacar(2500d);
        assertEquals(200d, conta.getSaldo());
    }

    @Test
    public void testDepositarSemChequeEspecial() {
        conta.depositar(50d);
        assertEquals(250d, conta.getSaldo());
    }

    @Test
    public void testDepositarComChequeEspecial() {
        conta.sacar(250d);
        conta.depositar(100d);
        assertEquals(48.50, conta.getSaldo());
    }
}

```

Questão 2 Considere o seguinte diagrama UML.



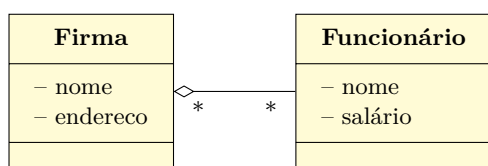
(a) Se existem 6 instâncias de autores, qual é o mínimo e o máximo de livros? E qual o mínimo e máximo de leitores?

Resposta 6 e 18; 0 e 72

(b) Se existem 6 instâncias de leitores, qual é o mínimo e o máximo de livros? E qual o mínimo e máximo de autores?

Resposta 6 e 30; 6 e 12

Questão 3 Considere o seguinte diagrama UML

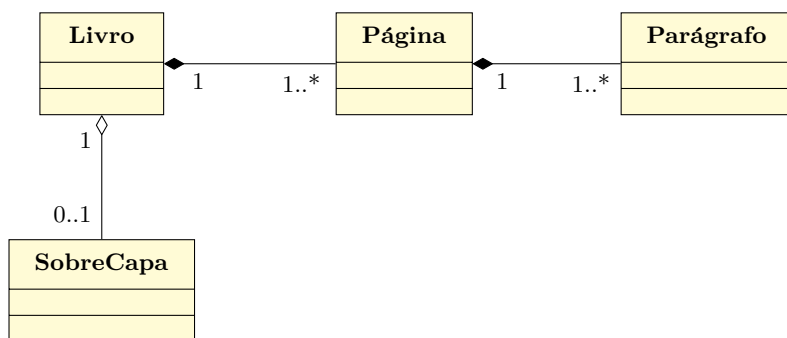


O atributo nome da classe *Firma* não pode repetir. Escreva quais são as alternativas **verdadeiras**.

- (a) Cada funcionário trabalha para pelo menos uma firma
- (b) Cada firma tem que ter pelo menos um funcionário
- (c) Nenhum funcionário pode trabalhar para mais de uma firma
- (d) Dois funcionários com o mesmo nome não podem trabalhar na mesma firma

Resposta a, c

Questão 4 Qual a diferença de interpretação entre os relacionamentos livro-sobrecapa e livro-páginas?



Resposta Livro-página é uma agregação por composição, logo se destruirmos o livro, somem as páginas e os parágrafos. Representamos um livro como algo fisicamente constituído de páginas. Na agregação livro-sobrecapa representamos que um livro pode ser constituído por uma sobrecapa mas que a sobrecapa é um objeto que existe independentemente do livro. Os tempos de vida dos objetos não são semelhantes.