

PROVA = 9.0
TRAB = 12/12

⇒ PA = 9.3



AVALIAÇÃO

NOTA

9,0

CURSO: Ciências da Computação

5º

PERÍODO:

5º

DISCIPLINA: Programação Funcional

COMP5N

TURMA:

COMP5

PROFESSOR(A): Carley Nobrega

☒ P1 ☐ 2ª Ch ☐ P2 ☐ PS

ALUNO (A): Farciso Deschamps Silva

MATRÍCULA: 201310038

OBSERVAÇÕES / ORIENTAÇÃO DE PROVA

- A prova terá início às 18:30h, com entrada permitida até às 19:45h. (só será permitida a saída da prova a partir das 19:45h).
- O material de consulta, quando autorizado é de uso individual, não pode ser emprestado / compartilhado.
- As respostas à lápis não serão submetidas à revisão de prova. (o aluno deverá manter sobre a mesa apenas material autorizado).
- Não é permitido o uso de quaisquer equipamentos. (exceto calculadora).

- É permitida a consulta **EXCLUSIVAMENTE** a materiais impressos. Está autorizado o uso de livros, apostilas, cópias dos slides e anotações do aluno.

QUESTÕES

- 2,0 1ª Questão (2,0 pontos): Explique por que funções são valores de 1ª classe. Justifique a sua resposta com um exemplo. Pois ela pode ser passada como parâmetro de outra função; pode retornar uma função e pode ser o valor de uma variável. Ex: $f(g(x))$
- 2,0 2ª Questão (2,0 pontos): O Máximo Divisor Comum de dois números inteiros positivos a e b , pode ser calculado pelo método *mdc* da classe *CalcMDC* abaixo.

```
class CalcMDC{
    def mdc(n1:Int, n2:Int):Int = {
        var a = n1
        var b = n2
        while(b!=0){
            var r = a % b
            a = b
            b = r
        }
        return a
    }
}
```

Entretanto, essa solução não é adequada do ponto de vista do paradigma de Programação Funcional. Mostre quais são os princípios de Programação Funcional que são violados pela implementação do método *mdc*. Para cada princípio listado, apresente uma pequena explicação sobre ele.

- Ele não utiliza uma função recursiva para obter o mdc
- Uso desnecessário de variável onde poderia usar o próprio parâmetro ao invés de colocá-lo em uma variável

1,5 3ª Questão (2,0 pontos): Escreva uma função em Scala que recebe como parâmetros 4 números reais e retorna o cálculo da variância.

Obs.: $\text{var} = \frac{1}{4} \sum_{i=1}^4 (x_i - \mu)^2$, onde x_i é o i -ésimo parâmetro e μ é média dos 4 parâmetros.

1,5 4ª Questão (2,0 pontos): Dada a seguinte função $A(m, n)$:

$$A(m, n) = \begin{cases} n+1 & \text{se } m=0 \\ A(n-1, 1) & \text{se } m>0 \text{ e } n=0 \\ A(m-1, A(m, n-1)) & \text{se } m>0 \text{ e } n>0 \end{cases}$$

válida para valores inteiros não negativos de m e n , implemente uma versão recursiva da função em Scala e mostre o resultado da execução de $A(1, 2)$.

2,0 5ª Questão (2,0 pontos): Usando **obrigatoriamente herança** e **sobrescrita** de métodos, crie uma subclasse de Conta que cobra uma tarifa de R\$ 1,00 de cada depósito ou saque e imprime na tela o saldo anterior, a operação (se foi depósito ou saque) e o saldo atual.

```
class Conta(saldoInicial: Double) {
  private var saldo = saldoInicial
  def deposito(valor: Double) = { saldo += valor; saldo }
  def saque(valor: Double) = { saldo -= valor; saldo }
}
```

1,5 3ª) def var(a: Double, b: Double, c: Double, d: Double) = {
 var m = (a+b+c+d)/4
 var n1 = (a-m)(a-m)
 var n2 = (b-m)(b-m)
 var n3 = (c-m)(c-m)
 var n4 = (d-m)(d-m)
 var res = (n1+n2+n3+n4)/4
 res }

Var RESCONTAR
0,5 POR NÃO TER UTILIZADO
0 WHILE
BOA PROVA!

1,5 4ª) def A(m: Int, n: Int) = {
 if (m == 0) RETURN n+1
 else if (m > 0 && n == 0) RETURN A(m-1, 1)
 else if (m > 0 && n > 0) RETURN A(m-1, A(m, n-1))
 } // aplicando A(1,2)

5ª) class Tarifa extends Conta {

var taxa = 1

override def deposito(valor: Double) = { println(Conta.saldo)

Conta.saldo += valor - taxa

println("depósito de: " + valor + "efetuado \n Novo saldo")

+ Conta.saldo)

override def saque(valor: Double) = { println("saldo anterior: " + Conta.saldo)

Conta.saldo -= valor - taxa

println("saque de: " + valor + "efetuado \n Novo saldo: " + Conta.saldo)

}

execução	m	n
1ª)	1	2
2ª)	0	2
3ª)	0	3

m=0 m=3