



# LINGUAGEM DE PROGRAMAÇÃO I

## NOTAS DE AULA

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNIESP

# Orientação de Envio

O envio deve ser feito pelo link do GitHub.  
Estruture cada nota de aula conforme o exemplo abaixo:



Esses exercícios  
valerão 3 pontos  
para a nota da prova!!!

# **NOTA DE AULA 1**

## Questão 1

Elabore um programa em java que informe se um aluno ele está aprovado, reprovado ou na final.

O sistema deve solicitar o nome e as notas do aluno.

Em seguida, realiza o cálculo da média e informa a situação acadêmica.

Se a média for  $\geq 70$  está aprovado, abaixo de 40 está reprovado, senão, está na final.

## Questão 2

Crie um sistema para controle de um **produto** com atributos **código, nome, tamanho/peso, cor, valor e quantidade no estoque**.

O sistema deve permitir o cadastro dos dados, além de realizar a venda e fazer o controle do estoque.

Para o pagamento, temos os seguintes benefícios:

Pix, Espécie, Transferência ou Débito, 5% de desconto.

Crédito, parcelar em 3x sem juros.

Observação: Espécie, caso o valor pago seja maior que o valor do produto, informar o troco.

# **NOTA DE AULA 2**

# Questão 1

Elabore um programa para um sistema bancário básico com Conta Corrente e Conta Poupança. Cada conta bancária terá um nome de titular e um saldo inicial zerado.

Inicialmente, a pessoa usuária deve selecionar o tipo de conta.

Após selecionada, o sistema deve exibir o menu de opções:

## **1. Conta Corrente:**

Depositar, sacar, usar cheque especial, exibir dados da conta

## **2. Conta Poupança:**

Depositar, sacar, calcular rendimento, exibir dados da conta

continua →

**Observação 1)** Cheque especial de 1.000,00

**Observação 2)** Calcular rendimento deve seguir a seguinte regra:

- Quando a taxa básica de juros (Selic) está acima de 8,5% ao ano, a poupança rende 0,5% ao mês.
- Quando a Selic está igual ou abaixo de 8,5% ao ano, a poupança tem um rendimento de 70% da Selic.

```
if (selic > 8.5) {  
    rendimento = 0.005 * saldo;  
}  
else {  
    rendimento = 0.007 * selic * saldo;  
}
```



## Questão 2

Elabore um sistema que deverá calcular o valor final de uma corrida de um Uber com base na distância percorrida, no tempo de espera e na tarifa base do serviço. O cálculo do valor também deve considerar um fator de demanda que aumenta o preço em horários de pico.

Atributos:

- `distancia` (double): Distância percorrida em quilômetros.
- `tempoEspera` (int): Tempo de espera em minutos.
- `tarifaBase` (double): Valor cobrado pela corrida.
- `fatorDemanda` (double): Valor do preço em horários de alta demanda

Métodos:

- *`calcularValorCorrida()`*:

Calcula o valor da corrida com base na fórmula:

$$\text{ValorFinal} = (\text{distancia} \times 2) + (\text{tempoEspera} \times 0.5) + \text{tarifaBase} \times \text{fatorDemanda}$$

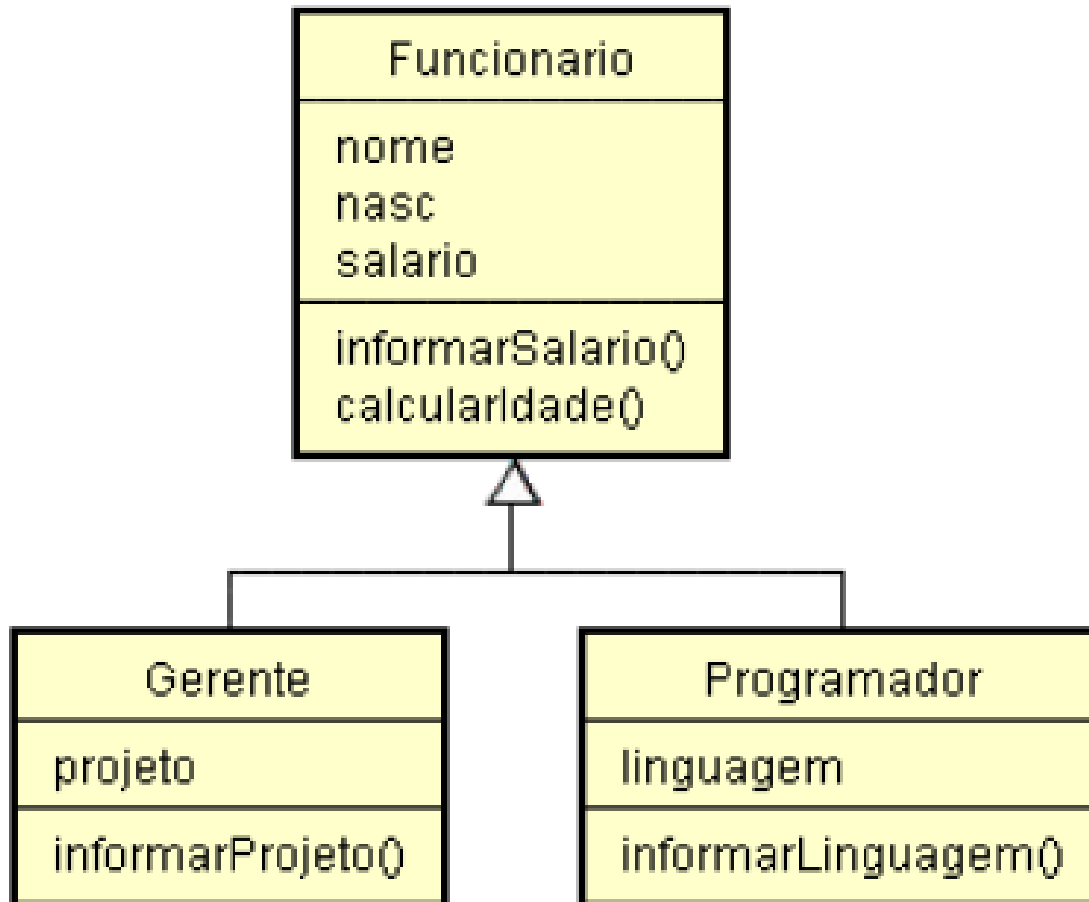
- *`exibirDetalhesCorrida()`*:

Exibe o valor final da corrida e os detalhes (distância, tempo de espera, tarifa base, fator de demanda).

# **NOTA DE AULA 3**

## Questão 1

Implemente o programa com base na modelagem:



## Questão 2

Implemente o programa com base na modelagem:

