



Quem se prepara, não para.

Programação Orientada a Objetos

Prática 3

3º período

Professora: Michelle Hanne

Orientações

- 1) Criar uma conta no GitHub - <https://github.com/>
- 2) Criar um repositório Público no GitHub com o nome **Prática3_OO**
- 3) Subir os arquivos para o repositório criado
- 4) Enviar o link do repositório na tarefa do Canvas.

OBS: O envio deverá ser realizado individualmente

Questão 1 - Carnaval

Esse projeto deverá conter as seguintes classes:

- 1- Crie uma classe chamada Ingresso que possui um valor em reais.
 - Crie o construtor dessa classe e os métodos Set e Get.
- 2- Crie uma classe Normal , que herda
 - Crie o método que imprime o valor do ingresso com a mensagem "Ingresso Normal".
- 3- Crie uma classe VIP , que herda Ingresso e possui um valor adicional.
 - Crie o construtor dessa classe com o valor do ingresso
 - Crie um método que imprime o valor do ingresso com o adicional incluído e a mensagem "Ingresso VIP".

Questão 1 - Carnaval

4. Crie uma classe Camarote , que herda Ingresso e possui a localização do camarote e um valor adicional.

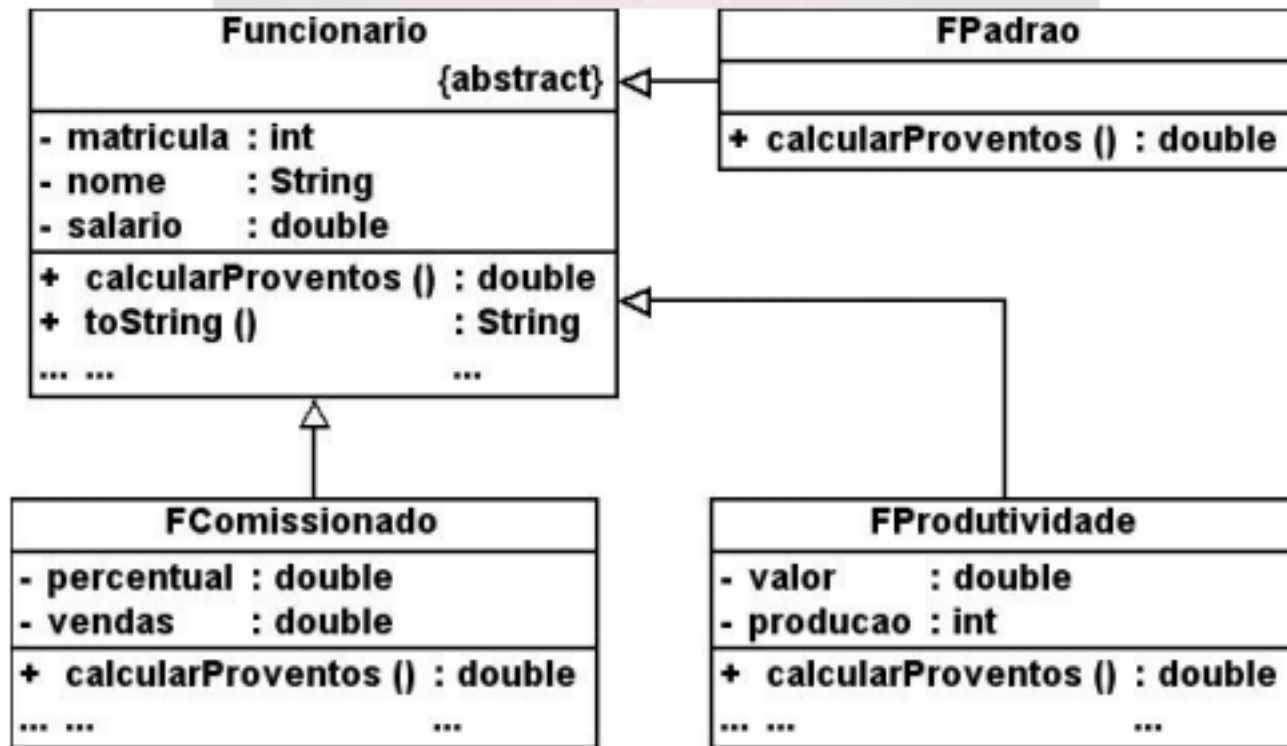
Crie o construtor dessa classe com o valor do ingresso, valor adicional e localização do camarote.

Crie um método que imprime o valor do ingresso com o adicional incluído, a localização do camarote e a mensagem "Ingresso Camarote".

No seu main, você deverá mostrar as opções: **(1) Ingresso Normal (2) VIP e (3) Camarote**. Cada opção deverá instanciar o seu respectivo construtor e chamar o respectivo método de impressão. Solicite a entrada de dados pelo teclado através do scanner ou **showMessageDialog**.

Questão 2 – Folha de Pagamento

Construir a seguinte hierarquia



Questão 2 – Folha de Pagamento

1- Objetivo: cadastrar os funcionários de uma empresa e calcular os proventos devidos a cada um deles por mês de trabalho.

2- Proventos: representam quaisquer valores devidos aos funcionários a título de remuneração.

Serão considerados apenas **três tipos de proventos:**

- **Salário fixo:** devido a todos os funcionários
- **Comissão:** adicional devido pelas vendas realizadas pelos funcionários que estão comissionados.
- **Produtividade:** adicional devido aos funcionários que trabalham diretamente na produção e que recebem determinando valor por unidade produzida.

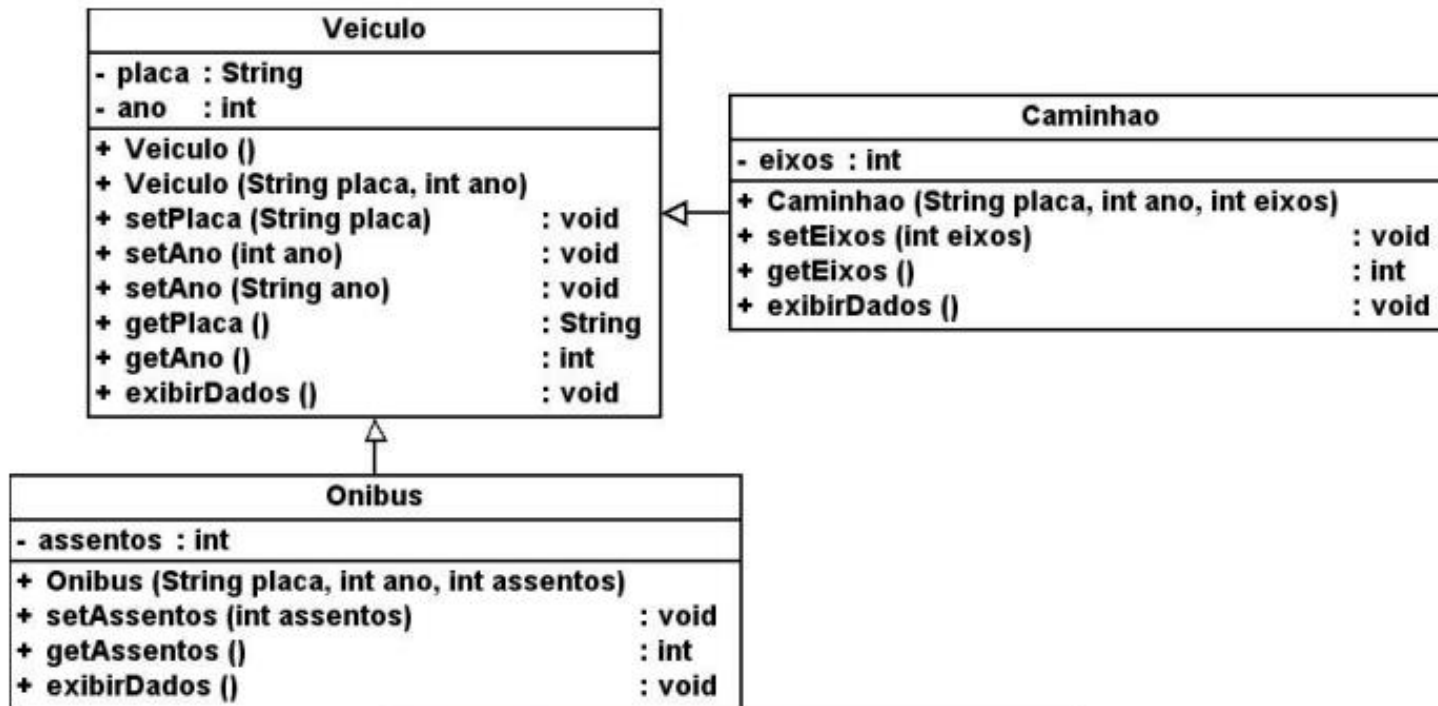
Questão 2 – Folha de Pagamento

3- Criar o construtor, os métodos Set e Get de cada classe.

4- Criar o Main() que permite entrar com os dados de um Funcionário e escolher o tipo: padrão, produtividade e comissionado. Mostrar o respectivo calcularProventos()

Questão 3 – Veículo

Criar a Hierarquia abaixo:



Questão 3 – Veículo

Classe Veiculo.java

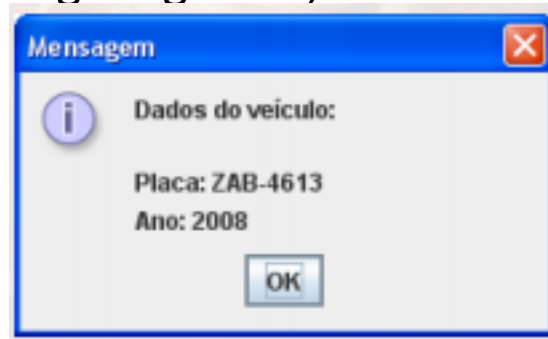
Construtor Veiculo()

Invocar o construtor Veículo (String placa, int ano)

Inicializar atributos com "" e 0

Método exibirDados()

Produzir uma mensagem gráfica, conforme imagem



Questão 3 – Veículo

Classe Onibus.java

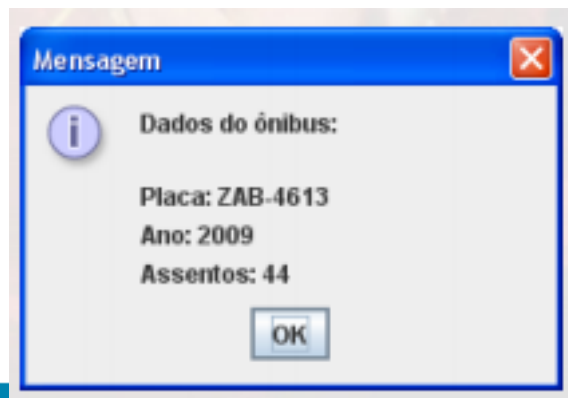
- **Construtor Onibus(String placa, int ano, int assentos)**

Invocar o construtor da superclasse: Veículo (String placa, int ano)

Complementar com a inicialização do atributo assentos

- **Método exibirDados()**

Produzir uma mensagem gráfica, conforme imagem



Questão 3 – Veículo

Classe Caminhao.java

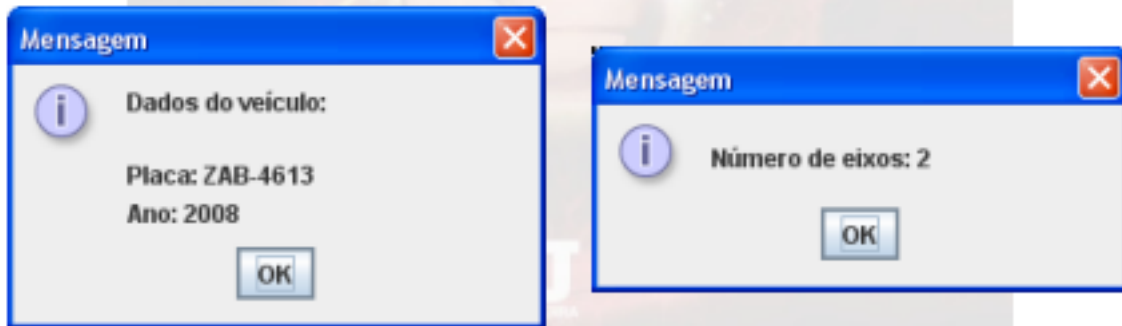
Construtor aminhao(String placa, int ano, int eixos)

Invocar o construtor da superclasse: Veículo (String placa, int ano)

Complementar com a inicialização do atributo assentos

Método exhibirDados()

Produzir separadamente as duas mensagens gráfica, conforme imagem



Questão 3 – Veículo

Main

- **Crie uma instância da classe Onibus**

Solicite os dados pelo teclado e utilize o construtor alternativo.

Execute o método `exibirDados()`

- **Crie uma instância da classe Caminhao**

Solicite os dados pelo teclado e utilize o construtor alternativo.

Execute o método `exibirDados()`

Questão 3 – Veículo

Faça as seguintes transformações na classe

Public abstract class Veiculo

Public final class Onibus extends Veículo

Public final class Caminhao extends Veiculo

Teste novamente o main, ocorreu algum problema?