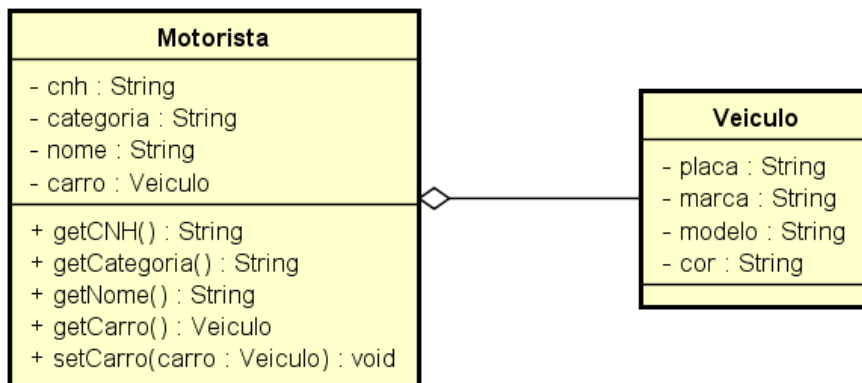


POO - Lista de Exercícios 4 – Prof. Adriano Nakamura

Associação, Agregação e Composição

1. Implemente as classes baseado no diagrama a seguir:



powered by Astah

A classe **Veiculo** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos;
- para cada atributo programe um método para retornar o seu valor.

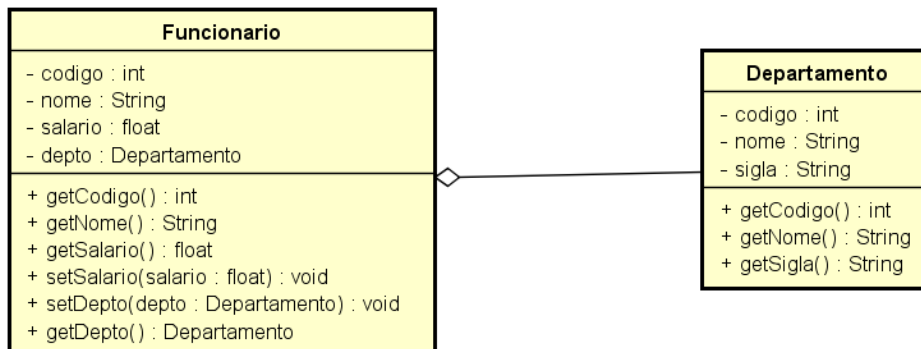
A classe **Motorista** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos, exceto o objeto Veículo;
- para cada atributo programe um método para retornar o seu valor;
- método para setar o veículo;

Implemente a classe **Principal** programando o método **main** da seguinte forma:

- crie 2 objetos da classe Veículo
- crie 1 objeto da classe Motorista
- sete um dos objetos veículo para o motorista
- a partir do objeto motorista, exiba seu nome, nro da CNH, o modelo e o nro da placa que dirige.

2. Implemente as classes baseado no diagrama a seguir:



powered by Astah

A classe **Departamento** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos;
- para cada atributo programe um método para retornar o seu valor.

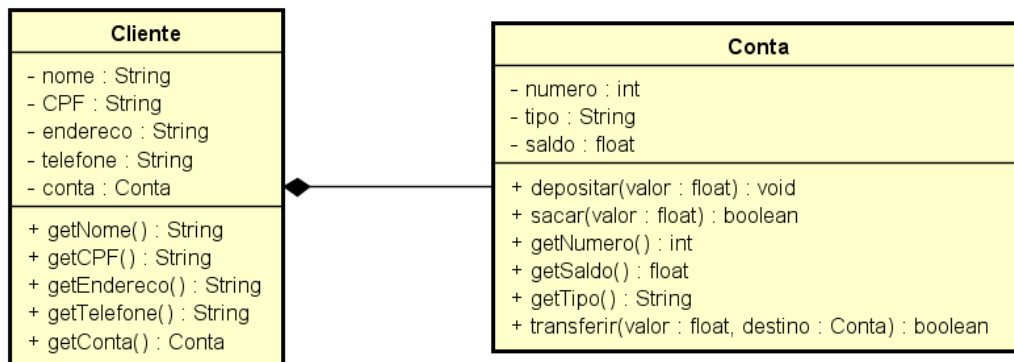
A classe **Funcionário** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos, exceto o objeto Departamento;
- para cada atributo programe um método para retornar o seu valor;
- método para setar o salário;
- método para setar o objeto departamento;

Implemente a classe **Principal** programando o método **main** da seguinte forma:

- crie 2 objetos da classe Departamento
- crie 2 objetos da classe Funcionário
- para cada funcionário sete o departamento para o qual foi alocado
- exiba os nomes do funcionário e do departamento em trabalho aquele que recebe o maior salário.

3. Implemente as classes baseado no diagrama a seguir:



powered by Astah

A classe **Conta** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos;
- para cada atributo programe um método para retornar o seu valor;
- método **depositar** recebe como parâmetro o valor do depósito que deve ser somado ao saldo;
- método **sacar** recebe como parâmetro o valor do saque que deve ser subtraído do saldo, caso seja possível e retornar true, do contrário retornar false;
- método **transferir** recebe como parâmetro o valor da transferência e a conta destino; verificar há saldo suficiente para transferência, subtrair o valor do saldo, depositar na conta destino e retornar true; caso a transferência não puder ser feita por saldo insuficiente retornar false;

A classe **Cliente** deve possuir os métodos:

- método **construtor** deve receber como parâmetro valores para iniciar todos os atributos (inclusive o objeto Conta);
- para cada atributo programe um método para retornar o seu valor;

Implemente a classe **Principal** programando o método **main** da seguinte forma:

- crie 2 objetos da classe Cliente (já passando por parâmetro o objeto Conta)
- faça um depósito e um saque para cada cliente (dê um feedback do saque)
- faça a transferência da conta de um cliente para outro (dê um feedback da operação)