

Em grupos de até 2 alunos. O trabalho é extraclasse – **NÃO DEVE SER FEITO DURANTE AS AULAS.**

Neste trabalho vamos simular uma pequena aplicação, Trata-se de uma intermediária entre proprietários de carros e pessoas que desejam alugar carros. O sistema irá viabilizar as locações, cobrando dos proprietários um percentual sobre o preço de cada locação. Seria a transferência para a locação de veículos, do sistema de locação de imóveis controlados por imobiliárias.

Regras do negócio

O sistema envolve dois tipos importantes de usuários: proprietários e locadores.

Proprietário – pessoa que cadastra seu carro no sistema para permitir que outras o aluguem e estipula o preço da diária.

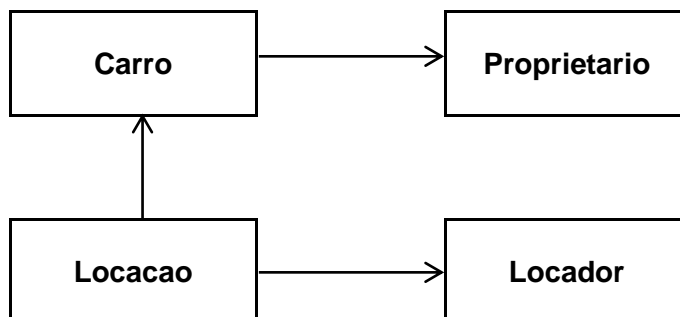
Locador – é o usuário que se cadastra no sistema para poder alugar carros. Cada vez que ele paga uma locação, ele ganha 1 ponto para cada R\$ 10,00 do pagamento. Com esses pontos acumulados ele poderá ganhar algum desconto em operações futuras.

A locação é o nome da transação que ocorre quando um locador aluga um carro de um proprietário, por um ou mais dias. A única modalidade de locação é por diárias.

O preço de uma locação é o número de diárias que durou a locação multiplicado pelo preço da diária. Deste valor, 85% vai para o proprietário e 15% fica com a administradora do sistema.

A administradora “banca” um possível desconto a cada locação, cujas regras estão descritas mais adiante.

Diagrama UML abaixo mostra o relacionamento entre as classes.



PROGRAME em Java as quatro classes, de acordo com as especificações abaixo.

Classe Proprietario

Atributos privados (somente estes):

- nome
- endereço
- total faturado – acumula os rendimentos auferidos com os alugueis



Construtor: apenas um com dois parâmetros, para nome e endereço. O total faturado deve ser inicializado com zero

Métodos:

+ *credita* – será chamado ao final de uma locação e receberá, como parâmetro, o valor que corresponde à parte do proprietário, para acumular no atributo que guarda o total já faturado.

+ *exibeDados* – mostra na tela os valores dos atributos, com títulos adequados para facilitar a leitura.

+ Métodos de acesso (get) e de configuração (set) que o aluno julgar necessários para o funcionamento do sistema.

Classe Carro

Atributos privados (somente estes):

- placa
- descrição – marca, modelo, ano etc.
- dono – variável objeto do tipo **Proprietario**
- preço da diária
- maior locação – registra a quantidade de dias que durou a maior locação já feita pelo carro
- quilometragem – corresponde à leitura do Hodômetro (aquele instrumento do painel que marca a quilometragem do veículo)



Construtores:

- O primeiro deve ter 5 parâmetros para: placa, descrição, um objeto do tipo **Proprietario** que indica o dono do carro, preço da diária e quilometragem atual do carro.
- O segundo deve ter os seguintes parâmetros: placa, descrição, nome do proprietário, endereço do proprietário, preço da diária e quilometragem atual do carro. O objeto **Proprietario** neste caso deve ser instanciado dentro do construtor.

Métodos:

+ *maiorLocacao* – será chamado ao final de uma locação e recebe via parâmetro a quantidade de diárias que durou uma locação e, se necessário, atualiza o atributo maior locação.

+ *exibeDados* – mostra na tela os conteúdos dos atributos, um em cada linha, com títulos.

+ métodos de configuração (set) para os atributos dono, preço da diária e quilometragem.

+ métodos de acesso (get) para todos os atributos.

Classe Locador

Atributos privados (somente estes):

- nome
- pontos – vão sendo acumulados para dar direito a descontos em futuras locações

Construtor: apenas um com parâmetro para o nome. Os pontos devem ser inicializados com zero.

Métodos:

+ *ganhaPontos* – será chamado quando encerrar uma locação e receberá o valor pago por ela, com a finalidade de acumular pontos. Para cada R\$ 10,00 pagos, o locador ganha 1 ponto.

+ get e set – para cada atributo



Atributos privados (somente estes):

- carro – variável objeto do tipo **Carro**
- locador – variável objeto do tipo **Locador**
- duração – tempo que durou a locação em dias
- preço – valor final pago pela locação

Construtor: apenas um com dois parâmetros, o primeiro do tipo **Carro** e o segundo do tipo **Locador**. A duração e o preço

devem ser inicializados com zero

A instanciação de um objeto desta classe corresponde ao início de uma locação, isto é, o momento em que o locador busca o carro na casa do proprietário.

Métodos:

+ devolve – método (não void) que será chamado quando o locador devolver o carro ao proprietário, encerrando uma locação. Vai receber, via parâmetros, o tempo de duração da locação (em dias) e a distância percorrida em km. Deve atualizar os atributos duração e preço.

Para o preço, o cálculo deve ser feito como segue:

- Calcular o valor de acordo com o valor de diária e as diárias consumidas.
- Creditar ao dono do carro, o valor calculado, menos 15% que é retido para a administradora do sistema.
- Depois disso, calcular possíveis descontos, que são “bancados” pela administradora, de acordo com os pontos que o proprietário acumulou antes desta locação:
 - Se os pontos são iguais ou acima de 250, o locador entra num sorteio (chamar o método fazSorteio para isso). Se ele ganhar o sorteio, terá o desconto no valor de uma diária e seus pontos serão zerados. Se não ganhar, fica sem desconto e mantém os pontos.
 - Se os pontos estão de 150 a 249, o desconto vale 7% do preço da locação e abate 150 pontos do locador;
 - Se os pontos estão de 100 a 149, o desconto será de 4% e abate 100 pontos.
- Abater do preço o desconto calculado anteriormente, se houver, e atualizar o atributo preço, além de chamar o método correspondente da classe **Locador** que irá lançar os pontos referentes ao que ele acaba de pagar pela locação.
- O método deve retornar o valor do desconto.

Obs. não esqueça de atualizar a quilometragem do carro e chamar o método maiorLocacao.

+ fazSorteio – gera um número randômico inteiro de 0 a 99 e pede para o usuário adivinhar a paridade (se é par ou ímpar) do número sorteado, oferecendo um diálogo para isso e lendo a resposta do usuário via teclado. Após a resposta do usuário, exibir o número sorteado e uma frase informando se ele acertou ou não a paridade. O método deve retornar true, se o usuário acertou, ou false, em caso contrário. Usar o tipo char para armazenar a resposta do usuário: por exemplo, P ou p, para indicar “par”; I ou i, para “ímpar”.

+ exhibeDados – mostra na tela todos os dados da locação, contendo:

- valores de todos os atributos do carro e seu proprietário
- valores de todos os atributos do locador
- a duração da locação e o seu preço final, se a locação já foi encerrada, ou a frase *A locação ainda não encerrou*. Uma locação termina quando ocorre a devolução do carro e a duração deixa de ser zero.



Entregar nas datas especificadas no plano de ensino:

- **Em papel**, os diagramas de classes e dois diagramas de objetos, com dados fictícios adequados. Um dos diagramas deve representar uma locação em andamento e o outro uma locação já encerrada.

- **Enviar pelo moodle** as quatro classes descritas acima. Não esquecer o seguinte comentário no início do texto fonte de cada classe:

/** Alunos : xxxxxxxxxxxxxxxxxxxx e xxxxxxxxxxxx Trabalho GA Turma:XX 2016/1 */

