

Trabalho do Grau B

Sistema para Locadora de Veículos

Apresentação: Os trabalhos serão apresentados pelos grupos diretamente ao professor na aula do dia **22/11/2022**. Não haverá prorrogação de data. O tempo máximo para apresentação por grupo será de 20 minutos. A ordem de apresentação será definida no dia da apresentação.

Instruções para envio do trabalho: Enviar somente os arquivos-fonte do projeto para a atividade aberta no **Canvas até as 19h30min do dia 22/11/2022**. Envios após esse horário terão desconto de 1 ponto. Apenas um integrante do grupo deve enviar os arquivos informando o nome de todos os integrantes do grupo.

Grupos: no máximo 4 integrantes.

Objetivo: Desenvolver um programa para gerenciamento básico de uma locadora de veículos.

Cenário: Um cliente acessa o sistema da locadora e solicita a locação de um carro, informando a cidade de origem. Ele consulta a lista de veículos disponíveis na cidade e escolhe um deles. Depois informa a quantidade de dias que pretende ficar com o veículo. Nesse momento, o sistema calcula e mostra para o cliente o valor total estimado das diárias e pergunta se o usuário quer reservar o carro. Caso afirmativo, o sistema registra a locação. Quando o cliente devolver o veículo, o sistema solicita seu nome, a cidade de devolução, e a quilometragem percorrida. O sistema também mostra a quantidade de diárias contratadas, e pede para o cliente confirmar ou informar uma quantidade diferente da reservada. Nesse instante, o sistema calcula e informa o valor referente aos quilômetros rodados e uma possível diferença no valor das diárias. Ainda na devolução, o sistema atualiza a cidade e o odômetro do carro, e o libera para nova locação.

Estrutura de Classes:

A estrutura abaixo é uma proposta que serve como ponto de partida para o desenvolvimento do sistema. Adicione novos métodos e atributos caso seja necessário.

Locadora
- veiculos: list (Veiculo) - locacoes: list (Locacao)
+ carrega_dados() + salva_dados() + consulta_veiculo() + realiza_locacao() + consulta_locacao() + realiza_devolucao() + relatorio_resumo()

Veiculo
- codigo: int - modelo: string - cor: string - ano: int - odometro: int - cidade: string - disponivel: bool - valor_diaria: float - valor_km_rodado: float

Locacao
- veiculo: Veiculo - cliente: string - origem: string - destino: string - km_rodado: int - qt_dias_reserva: int - qt_dias_realizado: int
+ valor_diarias(): float + valor_km_rodado(): float

Menu:

1. **Consultar veículos:** Usuário informa o nome de um atributo (modelo, cor, ano ou cidade) seguido do valor do atributo, e o sistema mostra uma tabela com todos os dados dos veículos que se enquadram na condição.
2. **Realizar locação:** Cliente informa a cidade de origem e o sistema mostra todos os veículos disponíveis para a cidade e seus respectivos atributos. Em seguida, o cliente indica qual o código do veículo que deseja alugar, informa seu nome e a quantidade de diárias. O sistema mostra o valor total das diárias e realiza a locação caso o cliente aceite. O sistema não deve permitir que o cliente tenha mais de um carro alugado simultaneamente.
3. **Realizar devolução:** Cliente informa seu nome, a cidade de devolução e a quilometragem percorrida. Nesse instante, o sistema informa a quantidade de dias contratados e o cliente confirma ou informa a quantidade real de dias que utilizou o carro. Caso tenha devolvido o carro antes, o sistema dá um desconto de 20% sobre as diárias não utilizadas, mas se devolveu depois, tem uma multa de 30% sobre as diárias extras. Em seguida, o sistema calcula e informa o valor referente aos quilômetros rodados. Para concluir o processo de devolução, o sistema atualiza a cidade e o odômetro do carro, e o libera para nova locação.
4. **Consultar locações:** Usuário informa nome do cliente ou modelo do veículo, e o sistema mostra todos os dados dos veículos e das locações correspondentes, ativas e finalizadas.
5. **Resumo:** Mostrar na tela um resumo de todas as locações já finalizadas, contendo os somatórios de km rodados, quantidade de dias contratados e realizados, valor das diárias contratadas, valor das diárias extras, valor dos km rodados e valor total das locações.
6. **Salvar:** Salva os dados dos veículos no arquivo "veiculos.txt". As locações tanto ativas quanto finalizadas devem ser salvas no arquivo "locacoes.txt". Atributos sem dados devem ser salvos como *string* vazia. Utilizar o caractere <TAB> como separador dos dados.
7. **Sair:** Encerra o sistema, liberando os objetos instanciados dinamicamente. Caso algum dado não tenha sido salvo em arquivo, salvar automaticamente.

Observações:

- Carregar automaticamente os arquivos "veiculos.txt" e "locacoes.txt" ao iniciar o programa, instanciando dinamicamente os objetos. Os formatos dos arquivos de exemplo devem ser obedecidos à risca;
- Todas as ações devem ser realizadas sobre os objetos em memória;
- Todas as funcionalidades descritas nesta definição devem ser implementadas;
- Os nomes de classes, atributos e métodos especificados acima na estrutura das classes devem ser mantidos na implementação do código (ou seja, não os renomeie). Novos métodos e atributos devem ser nomeados de acordo com a sua respectiva função;
- Se necessário, implemente métodos **getters** e **setters** para os atributos das classes. Métodos setters devem verificar o valor sendo setado ao atributo quando necessário;
- Para padronizar a persistência de dados, implementar os métodos "serializar()" e "deserializar()";