Centro Federal de Educação Tecnológica de Minas Gerais

Depto. de Computação – Curso: Engenharia Elétrica – Prof. Eduardo Cunha Campos

Trabalho 02 de Programação de Computadores II

Instruções Gerais

- Esta atividade deve ser realizada **em grupos**;
- Esteja atento às **observações sobre plágio** apresentadas no final desde documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet, gerados por ferramentas de IA (como ChatGPT, Gemini e similares), copiados de trabalhos de semestres anteriores, serão anulados;
- O trabalho deve ser implementado explorando adequadamente os conceitos e recursos apresentados em aula, com os devidos tratamentos de erros e da forma mais eficiente possível;
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Material de Apoio

https://eduardocunha11.github.io/firstblog/aulas/poo/backes/Apostila_Python.pdf

Objetivo Geral

Este trabalho tem como objetivo o desenvolvimento de um aplicativo de console **em Python** explorando os conceitos fundamentais de Programação Orientada a Objetos para gerenciar, em memória, a locação de veículos de uma locadora. O programa deve seguir as especificações a seguir para oferecer as funcionalidades de cadastro de **clientes**, cadastro de **veículos** e registro de **locações** de veículos cadastrados para clientes cadastrados.

Dos Clientes

A locadora presta serviços de locação para dois tipos de clientes: empresas e pessoas físicas. Tanto pessoas como empresas precisam ter no cadastro um código, um nome e um endereço. No caso de pessoas, é preciso ter também o CPF e a idade; no caso de empresas, o CNPj. O código dos novos clientes deve ser gerado automaticamente, de maneira incremental. Ele não deve ser solicitado ao usuário (utilize um atributo estático para contabilizar o número de objetos da classe instanciados).

Dos Veículos

A locadora trabalha com a locação de três tipos de veículos: automóveis, ônibus e motocicletas. Para todo veículo deve ser armazenado o código RENAVAM, o modelo, a quilometragem total, o valor da diária de locação e se o veículo se encontra locado ou não (booleano). Para automóveis, há também a capacidade do porta-malas. Para ônibus, há ainda o número de eixos e para motocicletas, a potência do motor em cilindradas. Deve ser implementado, para todo tipo de veículo, um método de nome **IniciarLocacao**, sem parâmetros, para registrar a informação de que o veículo está atualmente locado. Deve haver também um método de nome **FinalizarLocacao**, com um parâmetro **quilometragemRodada**, para registrar o término da locação do veículo e acrescentar a **quilometragemRodada** na quilometragem total do veículo.

Dos Clientes e Veiculos

Clientes e veículos devem ser "seguráveis", ou seja, deve haver um método para cálculo da diária do respectivo seguro, conforme especificações a seguir:

- Diária para cliente pessoa física: valor fixo de R$ 20,00 + R$ 0,50 por ano de vida;
- Diária para cliente pessoa jurídica: valor fixo de R$ 30,00;
- Diária para automóveis: valor fixo de R$ 20,00 + R$ 0,01 * quilometragem total;
- Diária para ônibus: valor fixo de R$ 50,00 + R$ 10,00 por eixo;
- Diária para motocicletas: valor fixo de R$ 40 + R$ 0,10 por cilindrada.

Das Locações

Para cada nova locação é necessário armazenar um código (gerado automaticamente), o cliente responsável (precisa estar cadastrado), o veículo sendo locado (precisa estar cadastrado), a data e hora da retirada do veículo (DateTime, pesquisar) e o número de diárias. A classe deve ter um método que calcule e retorne o valor total da locação. Esse valor corresponde à diária de locação do veículo multiplicado pelo número de diárias, acrescido dos seguros do cliente e do veículo (calcular para todos os dias locados). A classe deve disponibilizar também um método para encerrar a locação, registrando que o carro não está mais locado (ficando disponível, portanto, para novas locações). Esse método deve ter um parâmetro para indicação da quilometragem percorrida pelo cliente durante o período locado. Essa quilometragem deve ser adicionada à quilometragem total do veículo utilizando o respectivo método do veículo.

Do Repositório de Clientes

Deve ser criada uma classe para gerenciar os clientes cadastrados. A classe deve possuir um método **AdicionarCliente** que receba um objeto do tipo **Cliente** como parâmetro e o adicione em uma lista de clientes da classe. Deve possuir também um método **BuscarPorCodigo**, com parâmetro **codigo**. O método deve fazer uma busca sequencial na lista de clientes utilizando o código passado por parâmetro. Deve retornar o objeto cliente, caso localizado; ou informar que não encontrou o cliente, caso contrário.

Do Repositório de Veículos

De forma similar ao repositório de clientes, crie uma classe para gerenciar os veículos cadastrados. A classe deve ter um método que permita adicionar um novo veículo ao repositório e um único método de busca para buscar um veículo qualquer pelo código RENAVAM.

Do Programa Principal

Deve ser apresentado ao usuário um menu de opções conforme a seguir. O programa deverá voltar ao menu automaticamente depois de cada ação executada, até que o usuário escolha a opção 8 – Sair.

1 - Cadastrar Cliente

2 - Cadastrar Veículo

3 - Iniciar Locação

4 - Finalizar Locações

5 - Listar Clientes

6 - Listar Veículos

7 - Listar Locações

8 - Sair

Quando o usuário escolher a opção **1 – Cadastrar Cliente**, deverá ser solicitado, em um submenu, o tipo de cliente a ser cadastrado (pessoa física ou empresa). Posteriormente, as devidas informações devem ser solicitadas. O objeto referente ao novo cliente deve ser inserido adequadamente no repositório de clientes (utilizando a classe implementada previamente).

Ao escolher a opção **2 – Cadastrar Veículo**, deverá ser solicitado, em um submenu, o tipo de veículo a ser cadastrado. Na sequência, as informações daquele tipo de veículo devem ser solicitadas para efetivação do cadastro. O novo objeto criado deve ser inserido no repositório de veículos utilizando a classe implementada previamente.

Ao escolher a opção **3 – Iniciar Locação**, deverá ser solicitado o código RENAVAM do veículo sendo locado, o código do cliente fazendo a locação e as demais informações da locação (diárias, data e hora). A locação deve ser efetivada apenas se o veículo informado já não estiver locado. Não deve ser permitida a locação para clientes não cadastrados ou de veículos não cadastrados. A nova locação deve ser registrada no repositório de locações (deve haver uma classe para realizar o gerenciamento das locações, de forma similar aos outros repositórios). Após inserção no repositório, deve ser apresentado ao usuário o valor total da locação.

Ao escolher a opção **4 – Finalizar Locações**, o sistema deverá encerrar todas as locações do repositório que estejam abertas (com o veículo no estado de locado).

As opções 5 e 6 deverão listar, respectivamente, todas as informações de todos os clientes e veículos cadastrados no repositório.

A opção 7 deverá apresentar as informações de todas as locações realizadas, incluindo o estado atual da locação (ativa ou encerrada), com base no estado do veículo locado.

Entrega e Avaliação

Todos os trabalhos deverão ser enviados para o SIGAA. O grupo deverá apresentar o trabalho para o professor da disciplina. Além disso, todo o código-fonte do trabalho deverá ser zipado. Por fim, os nomes completos dos alunos bem como os seus respectivos números de matrícula deverão constar no arquivo zip enviado pelo SIGAA.