Programação Orientada a Objeto

INSTITUTO FEDERAL
Triângulo Mineiro
Campus Paracatu

Trabalho Prático – **Estacionamento**

Parte 02

Professor: Edwar Saliba Júnior Valor: 15 pontos

Contextualização:

O software que sua equipe e você desenvolveu para controle de estacionamento foi adotado pelo cliente e este o vem usando com frequência. Porém, o governo alterou as regras para que os estacionamentos possam funcionar. A partir de agora todos os dados dos veículos (Figura 1) que utilizaram o estacionamento deverão permanecer numa base de dados por, no mínimo, 5 anos. Com o objetivo de atender esta nova lei e também melhorar o *software* já construído, você deverá:

- corrigir as falhas que por ventura ainda existam na versão anterior e
- implementar a gravação dos dados manipulados pelo software em uma estrutura de Banco de Dados.

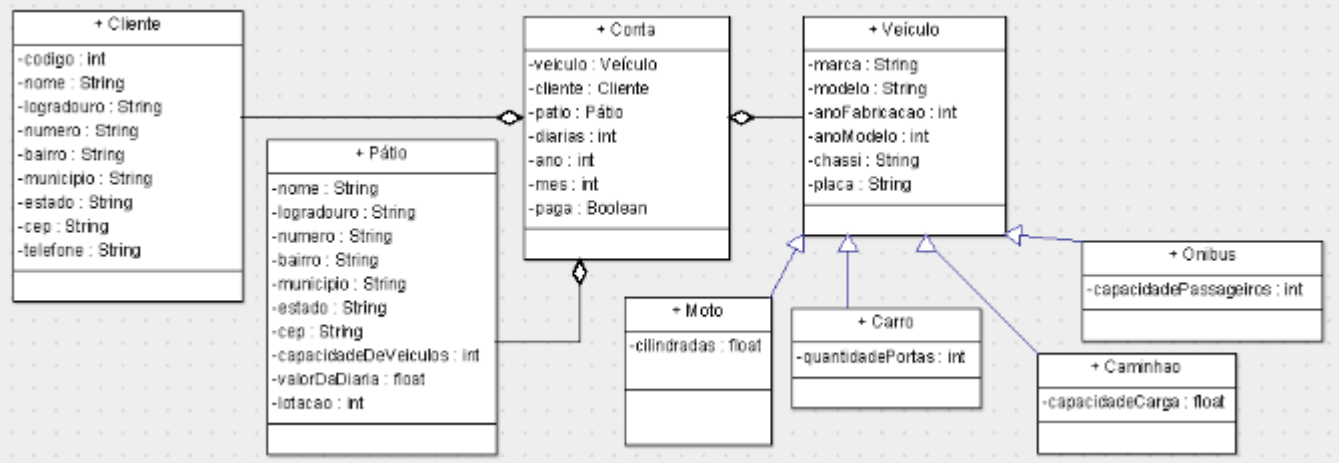

Figura 1: Diagrama de classes.

1) Caso seja necessário, os grupos poderão criar mais atributos nas classes. Os métodos deverão ser criados de acordo com a necessidade de cada *software*, por este motivo não foram especificados no diagrama UML (Figura 1);

2) Neste *software*, o usuário deverá ser capaz de cadastrar clientes, veículos, pátios e contas por pátio. Não deverá haver limite de cadastrados;

3) Crie um *menu* principal com os itens: Cliente, Veículo, Pátio, Conta e Sair. E para cada um destes itens, com exceção do item “Conta”, deverá existir as seguintes opções (*submenu*):
   - Cadastro,
   - Alteração
     - a operação de alteração deverá dar a possibilidade do usuário ver os valores já cadastrados para um determinado item do vetor. Item este, escolhido pelo usuário para que o mesmo possa alterá-lo,
     - atenção! Excluir um objeto e criar um novo para seu lugar não é alteração e será contado como erro no *software*;
   - Exclusão

    - antes de excluir um item do vetor, deverá ser emitida uma mensagem de confirmação da operação para o usuário. Caso o usuário confirme a operação, então o item será apagado. Caso contrário o item não será apagado;
- Consulta
    - deverá possibilitar ao usuário do *software* a visualização de um item do vetor;
- Relatório
    - deverá possibilitar ao usuário do *software* a visualização de todos os itens do vetor;
- Voltar ao *menu* principal
    - deverá possibilitar ao usuário a volta ao *menu* principal.

4) Para o item “Conta”, deverá existir as seguintes opções (*submenu*):
    - Cadastro de Conta
    - Exclusão de Conta
        - uma conta só poderá ser excluída se não houver débitos;
    - Inclusão de Diária
        - deverá dar a possibilidade, ao usuário do *software*, de incrementar o atributo “diárias” de um Cliente/Veículo/Pátio;
    - Exclusão de Diária,
        - deverá dar a possibilidade, ao usuário do *software*, decrementar o atributo “diárias” de um Cliente/Veículo/Pátio;
    - Total a Pagar
        - deverá mostrar quanto um Cliente/Veículo/Pátio está devendo em um determinado mês/ano, conta não paga;
        - nesta tela deverá haver também, uma opção que permita realizar o pagamento das diárias (todas de uma única vez) ou não;
    - Relatório de Contas a Receber
        - deverá dar a opção de apresentar o relatório:
            - baseado no mês / ano escolhido pelo usuário ou
            - contendo todas as contas já recebidas;
    - Relatório de Conas Recebidas
        - deverá dar a opção de apresentar o relatório:
            - baseado no mês / ano escolhido pelo usuário ou
            - contendo todas as contas já recebidas;
    - Voltar ao *menu* principal
        - deverá possibilitar ao usuário a volta ao *menu* principal.

5) Caso seja necessário, os grupos poderão criar mais itens no *submenu* de Conta;

6) **Os grupos não poderão criar mais atributos nas classes sem consulta prévia ao professor.** Os métodos deverão ser criados de acordo com a necessidade de cada *software*, por este motivo não foram especificados no diagrama UML (Figura 1);

7) Devem ser criadas as classes: Moto, Carro, Caminhão e Ônibus. Todas herdeiras de Veículo.

8) O atributo veículo da classe conta e toda e qualquer outra variável/atributo que trate de armazenamento de veículos, deverão ser do tipo Veículo obrigatoriamente. Para que não haja problema na manipulação destas variáveis/atributos, você deverá usar o recurso de *Polimorfismo* para armazenamento e recuperação dos objetos nelas eventualmente guardados.

9) Todas as informações do *software*, que até então ficavam apenas na memória principal do computador, agora deverão ser gravadas em banco de

dados. Para isto deverá ser criada uma classe DAO (*Data Access Object*) para cada uma das classes apresentadas na Figura 1.

10) O *software* deverá possuir tratamento para exceções em todos os métodos onde houver a possibilidade de se ter um erro fatal que gere a finalização do mesmo.

11) Deverá ser usado o Sistema Gerenciador de Banco de Dados (SGBD) conhecido como "PostgreSQL".

12) Deverão ser mantidas todas as características do *software* criadas na primeira fase do trabalho.

Como o seu *software* deverá funcionar

- A estrutura de *menus*, deverá proporcionar ao usuário:
  - a possibilidade de navegar entre o *menu* principal e seus *submenus* sem efetuar qualquer operação;
  - o usuário só poderá sair do programa através do *menu* principal, ou seja, acessando a opção "Sair";
- Um *software* deve ter uma boa aparência e ser de fácil utilização, para agradar e facilitar a vida de quem o utilizará.

Regras para a entrega do trabalho

- Deverá ser apresentado e entregue, o projeto (compactado) do código-fonte. Antes de compactar o projeto, exclua a pasta "dist" se ela existir.
- **O código-fonte que será entregue e apresentado não deverá possuir nenhum tipo de comentário.**
- **Deverá ser enviado, junto com o código fonte, o script para geração do banco de dados. Deverá ser enviado também um back-up do banco de dados (arquivo).**
- Deverá ser enviado para o *e-mail*: eddiesaliba2@yahoo.com (de acordo com as regras a seguir).
- **Não serão recebidos trabalhos após a data marcada para entrega.**
- **Para a apresentação no laboratório deverá ser levado pelo grupo, em *pendrive*, uma cópia dos arquivos que foram enviados por *e-mail*. Caso o grupo possua alguma restrição ou dificuldade no cumprimento desta regra, então, deverá avisar ao professor com antecedência mínima de 24 horas da data de apresentação.**

Regras para envio do *e-mail* com o trabalho

- No assunto do *e-mail* deve constar apenas o título: **IFTM – POO – Paracatu – ADS3PA – Fase 02**
- No corpo do *e-mail* deverá conter, única e exclusivamente, o nome completo de todos os integrantes do grupo (**um em cada linha**).
- Só será aceito UM *e-mail* por grupo. Portanto, verifique se está tudo certo com seu *e-mail* e trabalho antes de enviá-lo.

- **O *e-mail* deverá ser enviado, no máximo, até UM dia antes da data marcada para apresentação.**

Obs.: O desrespeito a qualquer das regras acima implicará na perda de créditos para o grupo.

Critérios de Avaliação no Laboratório:

- Conformidade do *software* em relação ao solicitado;
- Legibilidade do código (**organização, endentação e etc.**);
- Usabilidade das interfaces de interação com o usuário;
- **Entendimento <u>individual</u> a respeito do código-fonte apresentado.**