

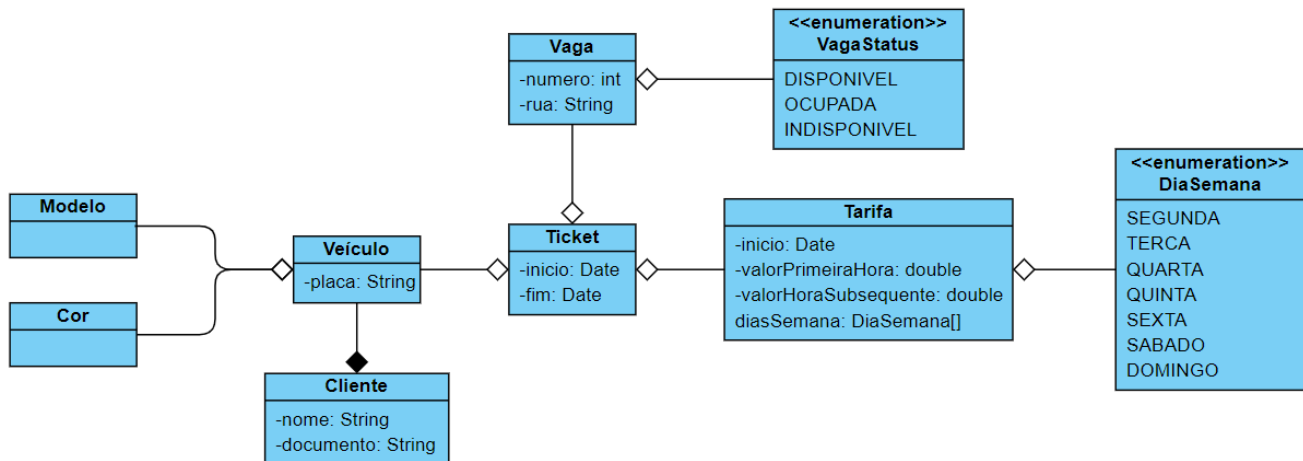


Aluno(a): _____ RGA: _____

Trabalho 1 (15/04/2024)

Descrição geral

Bruno e seu pai compraram um terreno e irão inaugurar um estacionamento. O estacionamento “Estaciona Bem” necessita de um sistema para manter e gerenciar informações sobre as vagas disponíveis/ocupadas, bem como de seus clientes e veículos. Para isso, a irmã de Bruno elaborou o seguinte projeto (Diagrama de Classes) básico:



No “Estaciona Bem”, há diversos tipos de vagas, as quais podem ser utilizadas por determinado tipo de veículo, desde que estejam disponíveis no momento que o cliente chegar para estacionar. Portanto, o *ticket* a ser gerado deve conter os dados de quando o veículo entrar no estacionamento com a hora de entrada gerada automaticamente, além de quem é o cliente e qual de seus carros ele está querendo estacionar.

Ao retornar ao estacionamento, o cliente deve devolver o *ticket*, em que o tempo de permanência deverá ser calculado ao registrar automaticamente a hora de saída. O “Estaciona Bem” possui uma tabela de preços diferentes de acordo com o dia da semana, a qual considera o tempo de permanência e o tipo de veículo estacionado. Essa tabela pode variar de acordo com a época do ano ou promoções lançadas por Bruno e seu pai.

Tanto os veículos quanto as vagas devem ter a identificação do tipo feitas no momento do cadastrado. Como existem tipos diferentes nessas classes supracitadas, é necessário implementar construtores diferentes para cada uma delas, de modo que a instanciação dos objetos seja feita de maneira adequada ao seu tipo de dado correspondente.

A classe “Estacionamento” deverá conter os atributos para armazenamento das informações sobre os dados que serão utilizados dentro do sistema. Recomenda-se que as funcionalidades estejam relacionadas a classes específicas de acordo com o contexto de cada classe. Além disso, o diagrama inicial deve ser complementado com os atributos e métodos utilizados. Outras classes podem ser incluídas, mas a estrutura inicial deve ser mantida.

Interface de interação do sistema

Criar uma classe específica que contenha métodos para geração dos menus, bem como gerenciar a forma de entrada de dados no sistema pelo usuário. O gerenciamento da interface pode ser feito em mais de uma classe, caso seja necessário.

O acesso às funcionalidades do sistema se dará pelo menu principal:

1 - Gerenciar clientes

- 2 - Gerenciar vagas
- 3 - Gerenciar estacionamento
- 4 - Cadastros gerais
- 5 - Consultar total faturado em um período
- 6 - Sair do programa

Ao acessar a **opção 1** do menu principal, deve disponibilizar o seguinte menu:

- 1 - Cadastrar
- 2 - Consultar por documento
- 3 - Excluir
- 4 - Editar
- 5 - Gerenciar veículos
- 6 - Listar todos os cadastros
- 7 - Voltar

Ao acessar a **opção 2** do menu principal, deve disponibilizar o seguinte menu:

- 1 - Cadastrar
- 2 - Consultar por número
- 3 - Excluir
- 4 - Editar
- 5 - Alterar disponibilidade
- 6 - Voltar

Ao acessar a **opção 3** do menu principal, deve disponibilizar o seguinte menu:

- 1 - Estacionar
- 2 - Retirar
- 3 - Listar todas as vagas disponíveis
- 4 - Gerenciar tarifas
- 5 - Voltar

O retorno dos métodos relacionados ao cadastros devem corresponder aos objetos referentes a cada funcionalidade. Por exemplo, o método para cadastrar clientes, deve-ser possibilitar que se entre com as informações dos clientes do estacionamento e, ao final, deve ser retornar um novo objeto do tipo criado.

Observe a separação entre as classes de negócio e a classe de interface. As classes de negócio não devem possuir nada que crie uma dependência da interface. Desta forma, se posteriormente você decidir mudar a interface, por exemplo, usar uma interface gráfica, as classes de negócio não precisam ser alteradas. A classe Estacionamento irá atuar como uma classe gerente, fazendo a interligação das classes de negócio com a classe de interface. Em um projeto mais profissional, provavelmente existiriam várias interfaces e classes gerente.

Requisitos

- Deve-se priorizar a declaração de todos os atributos com acesso privativo. Atributos que forem declarados com outros modificadores de acesso deverão ter breves comentários justificando a escolha.
- O cliente, vaga e/ou tarifa não pode ser excluído se existir algum *ticket* cadastrado.
- O estacionamento de um veículo somente poderá ser realizado se a vaga estiver disponível.
- O *status* de indisponível significa que a vaga não pode ser ocupada em nenhum momento, enquanto o status de ocupada/disponível vai depender da existência do *ticket*.
- Um cliente pode ter vários veículos, mas um veículo tem apenas um único proprietário.

- As vagas devem ser destinada a um tipo de veículo e apenas este tipo de veículo pode ser estacionado.
- O veículo poderá estar estacionado em apenas uma vaga no estacionamento por período (intervalo considerando a data de início e fim do *ticket*).
- O valor final do *ticket* deve ser calculado de acordo com o registro da tarifa, considerando o seu período.
- Um carro pode ser deixado estacionado por vários dias. Neste caso, o calculo de valores deve considerar as nuances do período envolvido.
- As entradas e saídas ficam a cargo do grupo (terminal ou interface gráfica – janelas de pop-up).
- O atendimento dos campos listados na especificação de cada funcionalidade deve ser seguido.
- Comentários sobre o funcionamento dos métodos, com exceção de *getters*, *setters* e construtores deverão estar presentes no código-fonte.
- Não serão contabilizados na nota comentários óbvios como:

```
//Este método calcula a média  
public double calculaMedia()
```

Considere descrever de forma mais específica:

```
/* Este método percorre cada disciplina cadastrada para um aluno,  
e para cada disciplina, percorre todas as notas cadastradas,  
acumula as notas e então calcula a média */  
public double calculaMedia()
```

Critérios de avaliação

O trabalho deverá ser entregue no AVA até o dia 15/04/2024. Deve ser entregue um arquivo compactado contendo o projeto do Netbeans (ou outra IDE que esteja utilizando). A avaliação será dividida em 8 (oito) etapas e em cada etapa será avaliada a codificação e a execução. As etapas são:

Critério	Pontuação
Estacionamento e classes de funcionalidades	1,5
Cliente	1,0
Veículo e derivados	1,5
Ticket e Tarifa	2,0
Vaga	1,0
Interface	1,0
Relatórios de faturamento em um período	1,5
Comentários	0,5