**Universidade Federal de Minas Gerais**

**Disciplina**: Progamação Modular

**Tarefa**: Estacionamento Modular

**Alunos**: Rodrigo Augusto Gontijo

Caio Godoy

**Introdução:**

Este trabalho consiste na implementação de um software que irá organizar um estacionamento vertical de um prédio , organizando por exemplo: disponibilidade de vagas, indicando a vaga que o automóvel poderá estacionar, calculando valores a serem pagos, mostrando tempo de permanência.

Devido à diversidade de vagas com suas diferentes quantidades e diferentes preços/hora, o software tornou-se a melhor solução para este problema, pois com todos esses requisitos, a implementação manual seria muito demorada.

O programa tem um funcionamento simples para o usuário, ele irá entrar com os seguintes dados:

* Tipo de atendimento ( Entrada ou saída).
* Horário que o automóvel entrou ou saiu do estabelecimento.
* Placa.
* Tipo de veículo.

Feita essa entrada, o sistema irá processar os dados informados, e emitir uma saída, sendo ela de dois tipos:

* Caso o automóvel esteja entrando, simplesmente irá mostrar a vaga que ele pode estacionar ( caso haja disponibilidade ).
* Caso o automóvel esteja saindo, o sistema irá mostrar o tipo de vaga, o tempo de uso da mesma e o preço a ser pago.

**Implementação:**

Para a implementação desse software, foi feito um TAD para o tipo de vaga, mudando somente os valores e constantes que são inseridos/calculados nesse TAD. Ele é composto por :

* Hora de saída.
* Hora de entrada.
* Vaga está ocupada.
* Andar.
* Número da vaga.
* Número do carro que irá ocupar essa vaga.

E também pelas funções que iram dar entrada nesses dados e por funções que iram exibir esses dados.

Como todas as vagas possuem os mesmos métodos e tipos, foi criada uma interface a ser implementada por todas as classes de vagas. Essa escolha foi feita para deixar o código flexível para se criar futuros novos tipos de vaga.

O funcionamento do programa ocorre da seguinte maneira:

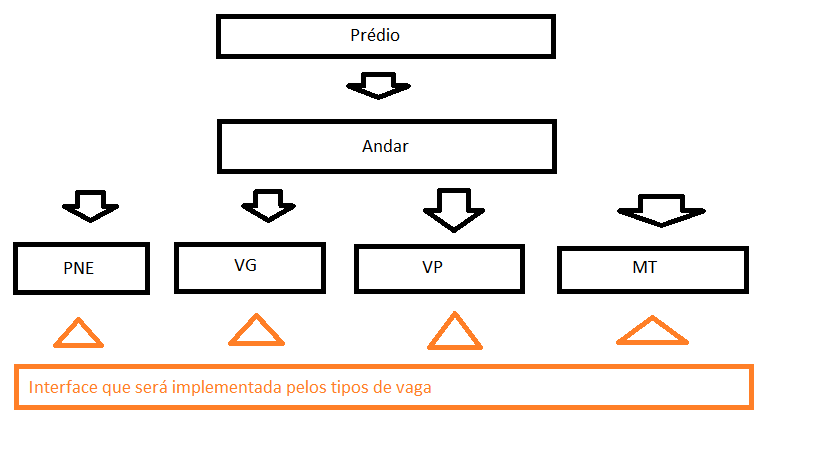
* É criado uma coleção de andares, pois deixamos o código para se criar uma quantidade variável de andares.
* Dentro de cada coleção de andares, é criada uma coleção das vagas existente, também com quantidades variáveis.
* Todos os valores fixos do programa estão no arquivo de constantes, como quantidade de andares, quantidades de uma determinada vaga por andar, preço da hora de um determinado tipo de vaga e o código de um determinado tipo de vaga.

Feito os passos acima, o código está pronto para receber os dados, sendo que esses serão lidos por meio de um arquivo txt.

Após a leitura desses dados, as informações são dividas e interpretadas. Sempre manipulando as classes de Andar e as classes do tipo de vaga, e também sempre verificando, lendo e guardando as informações.

Destaco que foi implementada as opções diferenciadas requeridas no enunciado, com um certo tipo de automóvel estacionando em uma vaga que não foi , primeiramente, reservada para ele, está implementada juntamente com a verificação da tipo de vaga inicial ( quando verifica-se a disponibilidade daquele tipo de vaga, caso não haja essa disponibilidade, é verificada a disponibilidade do outro tipo de vaga que esse veículo pode estacionar).

Foto ilustrativa de como funciona o programa:



**Testes:**

Além do teste passado no enunciado, foram feitos alguns testes, todos funcionaram perfeitamente na implementação feita

**Conclusão:**

O trabalho foi de suma importância para fixar os ensinamentos passados em sala. Ambos os integrantes possuem contato diário com desenvolvimento de software e é de vital importância a fixação desses conceitos.

Algumas dificuldades encontradas:

* Manipulação de arquivos em Java é um pouco mais complicada do que imaginávamos.
* Elaborar a melhor implementação possível para o trabalho, possibilitando, por exemplo: flexibilidade de código, modularização do projeto, fácil alteração futura, etc.

**Bibliografia:**

<http://www.devmedia.com.br/leitura-e-escrita-de-arquivos-de-texto-em-java/25529>

<http://pt.stackoverflow.com/questions/88345/passagem-de-par%c3%a2metro-usando-o-string-args-declarado-na-main>

<http://www.devmedia.com.br/convencoes-de-codigo-java/23871>