

## Trabalho 1 – Algoritmos II – UNISUL – Ricardo Ribeiro Assink

Nomeie o arquivo como Trabalho1.java e traga o mesmo em um **pendrive ou CD/DVD**.

O trabalho deve ser entregue e apresentado pela dupla no dia **28/09/2015 na Pedra Branca ou 23/09/2015 na Dib Mussi, na sala de aula a partir das 18:15.**

### Observações:

O arquivo deve ser compilado em linha de comando **javac Trabalho1.java** no computador do professor.

- **Observações:**

- **NÃO “EMPRESTE” seu trabalho para outros colegas, se identificado plágio é nota ZERO para todos os envolvidos.**
- **LEIA** cada questão antes de começar.
- O arquivo deve ser compilado em linha de comando **javac Trabalho1.java** no computador do professor.
- A avaliação será feita por meio de análise do código e perguntas aos alunos sobre a construção do mesmo.
- O trabalho é em dupla, mas a nota é individual mediante apresentação do trabalho.
- Implemente **UMA QUESTÃO POR VEZ** e separe cada uma delas no código fonte usando comentários copiando e colando o enunciado no topo da mesma.
- **NENHUM OUTRO COMENTÁRIO SERÁ PERMITIDO DENTRO DO CÓDIGO.**  
(1 ponto a menos)
- **A INDENTAÇÃO SERÁ COBRADA! ORGANIZE SEU CÓDIGO.**  
(1 ponto a menos)

### PROBLEMA:

Tiburcio é um mega empresário dono do grupo Tiburcio's Corporate Group. Ele está lançando seu novo negócio que é o Tiburcio's PARK, uma rede de estacionamentos de veículos alocada em prédios compactos adaptados ao pouco espaço disponível em grandes centros de negócios. O Tiburcio só atende grandes executivos, não julgue a pessoa pelo carro que ela tem... :P

Para a implantação do projeto piloto, será necessário implementar um software para testes e para isso ele resolveu contratar programadores da UNISUL.

Seu estacionamento piloto tem 3 andares com apenas 4 vagas em cada andar.

O serviço VIP do Tiburcio controla também a kilometragem dos veículos, já que as vagas são apenas para mensalistas será importante manter relatórios na versão final, por hora, é importante apenas prever este item no software piloto.

**DICA 1 : para implementar o software piloto crie 4 MATRIZES 3x4 (3 linhas e 4 colunas).**

**1 – com o número das vagas, cadastre previamente os valores (101,102, etc) (STRING)**

**2 – com os nomes dos donos(STRING)**

**3 – com os modelos dos veículos(STRING)**

**4 – com as leituras de kilometragem(INT)**

Cada VAGA segue numeração padrão.

Andar 1: 101, 102, 103, 104

Andar 2: 201, 202, 203, 204

Andar 3: 301, 302, 303, 304

Os nomes dos donos dos carros seguem a lista abaixo:

101 – Ricardo

102 – João

103 – Maria

104 – José

201 – Julia

202 – Marcos

203 – Paulo

204 – Pedro

301 – Bruno

302 – Isabela

303 – Rafaela

304 – Gabriel

Os modelos dos veículos seguem a lista abaixo:

101 – Ferrari

102 – Mercedes

103 – Fusca

104 – Lamborghini

201 – Evoque

202 – BMW

203 – Fusca

204 – Fiat 147

301 – Dodge

302 – Mustang

303 – Ferrari

304 – Fusca

## **CADASTRE TODOS OS DADOS ANTERIORES DIRETAMENTE NO CÓDIGO.**

Faça um programa que manipule as kilometragens de cada veículo, o algoritmo deve calcular e mostrar na tela os resultados das questões abaixo:

**1)** Solicite ao usuário a kilometragem de cada veículo, a mensagem que aparece para o usuário deve seguir o seguinte modelo:

"Olá (nome do dono do Veículo), sua vaga é a (número da vaga), insira a kilometragem do seu veículo (Modelo do Veículo) : "

Os campos entre parênteses devem ser substituídos pelos respectivos nomes e modelos.

**2)** Solicite ao usuário que informe um modelo de veículo(STRING) e utilizando PESQUISA LINEAR mostre o número de veículos com o mesmo modelo no estacionamento piloto da Tiburcio's PARK.

**3)** Mostre a média de Kilometragem dos carros do estacionamento.

**4)** Mostre o nome do dono, o número da vaga e o modelo do veículo que MAIS andou.

**5)** Mostre o número da vaga do veículo que MENOS andou e o nome do dono deste veículo.

**6)** Solicite que o usuário informe o número de uma das vagas(STRING), mostre o nome do dono, modelo e kilometragem deste veículo.

**7)** Mostre todas as leituras de kilometragem ordenadas. Para isso coloque todas as leituras em um VETOR e ordene com o BubbleSort.

**8)** Mostre o número de veículos que possuem kilometragem superior a média calculada na opção 3.

**9)** Solicite que o usuário informe o nome de um dono, mostre o modelo, a kilometragem deste veículo e o número da vaga.

**10)** Considerando as vagas com final 4 (104,204,304), mostre o nome do dono, o modelo e a kilometragem.