

REVISÃO – EXERCÍCIO DOS VEÍCULOS

ETAPA 1 – PREPARAÇÃO

1. Baixe o Material de Apoio disponível no Moodle com o exemplo das classes “Placa” e “Veiculo”
2. Abra o exemplo no VSCode, estude o código e responda as perguntas que seguem:
 - a. Quais são os atributos (variáveis) da classe “Placa”?
 - b. Porque os atributos da classe “Placa” são privados?
 - c. A classe “Placa” tem método construtor? O que ele faz? Quando ele é executado?
 - d. Toda a classe deve ter um método construtor? O que acontece quando uma classe não tem método construtor?
 - e. O que são métodos “getter” e “setter”? O que os caracteriza?
 - f. A classe “Placa” tem métodos do tipo “setter”? Por quê?
 - g. Quais as implicações em se criar métodos “setter” para todos os atributos de uma classe?
 - h. Quantos atributos tem a classe “Veiculo”? De que tipo são? Algum dos atributos de “Veiculo” indica um relacionamento entre as classes?
 - i. Porque os atributos da classe “Veiculo” são privados? Qual a vantagem de se manter os atributos das classes privados?
 - j. A classe “Veiculo” tem método construtor? O que ele faz? Quando é executado?
 - k. Porque não é necessário criar uma instancia da classe “Placa” no construtor da classe “Veiculo”?
 - l. A classe “Veiculo” tem métodos do tipo “setter”? Por quê?
 - m. Para que serve o modificador “final” no atributo “consumoPorLitro”?
 - n. De que maneira funcionam os métodos “getCombustivelNoTanque”, “abastece” e “dirige”?
 - o. Quantas instâncias de que classes são criadas no “main”?
3. Compile e execute o código.
4. Certifique-se de ter dominado todo o processo de edição, compilação e execução de programas proposto para este semestre.

ETAPA 2 – EXERCÍCIOS DE PROGRAMAÇÃO

1. Do jeito que foi projetada, a classe “Veiculo” modela um veículo cujo tanque de combustível tem capacidade infinita. Altere a classe “Veiculo” de maneira que o tanque de combustível passe a ter uma capacidade finita.
2. Reescreva o método “main” de maneira que o mesmo receba pelos argumentos da linha de comando a placa do veículo, a quantidade de combustível que o mesmo deve ser abastecido e a distância que ele deve percorrer e imprima: os dados do veículo, a distância efetivamente percorrida (com o combustível disponível) e o combustível remanescente no tanque. Faça consistência sobre a quantidade e o tipo dos parâmetros informados exibindo uma mensagem adequada se for o caso.
3. Reescreva o método “main” da seguinte forma: crie um arranjo e armazene no mesmo 5 veículos diferentes. Apresente para o usuário um menu com três opções: “1) Abastecer”, “2) dirigir” e “3) Fim”. A opção 1 solicita a placa do veículo a ser abastecido e a quantidade de combustível e, ao final da operação, exibe na tela a quantidade total de combustível no veículo após o abastecimento. A opção 2 solicita a placa do veículo que o usuário pretende dirigir e a distância a ser percorrida e, ao final da operação, imprime na tela a distância efetivamente percorrida e a quantidade de combustível remanescente no tanque. Tanto a opção 1 como a opção 2 devem exibir a mensagem “Veiculo não encontrado” caso a placa informada não conste do cadastro (arranjo). A opção 3 deve imprimir os dados de todos os veículos cadastrados e encerrar o programa.
4. Escreva um novo código para o método “main”: crie uma instancia da classe “Placa” e informe a esta mesma placa para duas instancias de “Veiculo”. Exiba na tela as informações de cada veículo a partir do método “toString”. Em seguida, usando o método “setPais” da classe “Veiculo”, altere o país do segundo veículo e imprima novamente as informações dos dois veículos. Explique: por que as informações de ambos os veículos foram alteradas?