



Ciência da Computação
Programação Orientada a Objetos - POO
Professor: Anderson Elias

LISTA – 15 Herança / Thread

1) Crie uma classe chamada **Veiculo**, que conterá os seguintes atributos privados: chassi (String), tipo (carro ou bicicleta - String) e velocidadeMaxima (double). Essa classe deverá possuir um construtor vazio e um construtor que inicialize todos os atributos.

2) Crie duas subclasses para **Veiculo**:

a) Uma chamada **Bicicleta**, que possuirá o atributo privado nrMarchas (int);

b) Uma chamada **Carro**, que possuirá o atributo privado combustível (double).

Crie para as duas subclasses um construtor vazio e um construtor que inicializa todos os atributos. Lembrando que como elas herdam de **Veiculo**, também possuem os atributos que **Veiculo** possui.

3) Adicione à classe **Veiculo** os seguintes métodos:

a) **limpar**: método público que ao ser chamado imprime a seguinte mensagem: "O veículo TIPO_DO_VEICULO" está sendo limpo;

b) **ajustar**: método público que ao ser chamado imprime uma mensagem dizendo: "O veículo TIPO_DO_VEICULO" está sendo ajustado;

c) **incrementarVeloc**: método público que recebe um valor inteiro e que soma esse inteiro à velocidadeMaxima do veículo.

4) Adicione à classe **Carro** o seguinte método:

a) **trocarOleo**: método público que imprime uma mensagem dizendo: "O veículo TIPO_DO_VEICULO trocou o óleo".

5) Na classe **Bicicleta**, seguinte método:

a) **incrementarVeloc**: método público que recebe um inteiro e soma esse inteiro ao nrMarchas da bicicleta. Além disso, ele aumenta a velocidade da bicicleta no valor da % do número de marchas incrementadas.

6) Crie a classe **Oficina**:

a) A classe **Oficina** terá como atributo um array de **Veiculos** e um double com o valor apurado da oficina.

b) Possuirá os seguintes métodos:

- **adicionarVeiculoFila**(Veiculo v): recebe um veículo e o adiciona no array de veículos;

- **atenderProximo()**: olha o próximo da fila e o atende, isto é, realiza a limpeza(método limpar) e conserta o veículo (método ajustar).

Em seguida, pergunta ao usuário se ele deseja incrementar o veículo, se a resposta for SIM, ele pede um valor e faz o incremento no veículo (método incrementarVeloc).

Em seguida, o método avalia se a classe é um Carro, se for ele pergunta se o mesmo deseja trocar o óleo, caso a resposta seja SIM, troque o óleo do carro (método trocarOleo).

Por fim, como o veículo já foi atendido, ele o remove da fila.

7) Crie uma classe **Teste** com um método main, crie 2 Bicicletas e 2 Carros. Crie uma classe **Oficina**, e em seguida chame o método **adicionarVeiculoFila**, e adicione 1 bicicleta, 1 carro, 1 carro e 1 bicicleta.

Por fim, faça um loop que vá de 1 a 4 e chame o método **atenderProximo** da oficina nesse loop, de modo que os 4 veículos que entraram sejam atendidos.

8) Crie um fluxo paralelo para realizar uma corrida de 5 voltas onde cada veículo serão os competidores e terão tempos de pitStops diferentes: (Carro 1500 e Bicicleta 950) milissegundos. Inicie a corrida informando o nome do veículo que largou e seu tipo. Também informe o nome de cada veículo e seu tipo após realizar uma volta. Caso aconteça algum problema durante a volta relate qual foi o veículo e seu tipo. Fique à vontade para sincronizar, priorizar, adormecer ou até parar as threads. Também se necessário, faça com que eles aguardem por algum momento e os notifique para que voltem a executar. Ao concluir a corrida, informe o veículo que terminou.

Façam com bastante atenção.

Abraços.