

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO SUL - CAMPUS OSÓRIO
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

Avaliação de Programação Orientada a Objetos I

Orientações Para a Realização da Prova Prática

- Esta atividade de avaliação é individual.
- Ao concluir a atividade, zip o projeto completo do Netbeans e renomeie o arquivo com o seu nome completo.
- Envie o arquivo zipado através do link disponível no Moodle.

Descrição do Problema e Tarefas

Crie a classe Pessoa, que tem como atributos nome e cpf.

Crie uma classe Veículo abstrata. Todo veículo tem uma pessoa que é a proprietária do mesmo, um registro, modelo, marca e quilometragem (que sempre começa zerada). Cabe a você definir o tipo que melhor representa cada um desses atributos.

Crie a classe Registro, do tipo record, que tem como atributos o número (inteiro), local (String), data (String).

Sabendo que a relação entre Veículo e Registro é de agregação, crie um construtor que respeite esse tipo de relacionamento e crie, também, o método abstrato mover (distância), que retorna o sentido do movimento.

Especialize a classe Veículo, criando as subclasses:

- Avião: além do proprietário principal, pode ter mais três pessoas proprietárias do avião. Tem como atributo a finalidade do avião (de passageiros, de carga ou agrícola - crie uma enum para isso) e como método extra a função getConteudo (o método deve retornar uma String como uma mensagem dizendo o que o avião está transportando: mercadorias, passageiros ou veneno). Um avião de carga pode virar avião de passageiros e vice-versa, mas avião agrícola não pode mudar de finalidade. O avião se move apenas para frente, sendo que a distância percorrida é acrescentada à sua quilometragem. Deve ser possível, também, verificar se uma Pessoa é proprietária de um Avião.
- CicloBike: tem apenas uma pessoa proprietária. A relação entre CicloBike e Pessoa é de composição. A CicloBike tem como atributo a cor e se move para frente ou para trás, dependendo se a distância é positiva ou negativa. Essa distância, mesmo se negativa, deve ser contabilizada na quilometragem do veículo.

Dicas

- Implemente as classes na seguinte ordem: Pessoa, Finalidade, Registro, Veiculo, Aviao, CicloBike.
- Utilize Math.abs(int value) para obter o valor absoluto da quilometragem.
- Utilize o atalho CTRL6 para ver a lista de TODOs, indicando o que precisa ser feito no código
- Utilize as classes de teste, conforme as instruções abaixo.

Classes de Teste:

As classes de testes estão dentro do TestPackages. Repare que, para cada entidade, existe uma classe de testes correspondente. Execute os testes na seguinte ordem:

1. TestaPessoa
2. TestaRegistro
3. TestaCicloBike
4. TestaAviao

Em cada classe de teste você encontrará, ao lado ou abaixo de cada linha, na forma de comentário de código, o resultado esperado da execução da linha acima. Por exemplo, ao executar o código. Por exemplo:

```
Pessoa p1 = new Pessoa("Marco", "1234-00");
```

```
Aviao av2 = new Aviao(p1, "", "boing", Finalidade.CARGA);
```

```
System.out.println(av2);
```

```
//Exception in thread "main" java.lang.IllegalArgumentException: O  
modelo nao pode ser nulo ou vazio
```

Avaliação

- enum Finalidade - até 1 ponto
- record Registro - até 0,9 ponto
- classe Veiculo - até 2,8 pontos
- classe CicloBike - até 1,6 pontos
- classe Aviao - até 3,7 pontos