

MC302
Primeiro semestre de 2018

Laboratório 2

Professora: Esther Colombini (esther@ic.unicamp.br)

PEDs: Nathana Facion (nathanafacion@gmail.com), Rafael Tomazela (sohakes@gmail.com), Luis Fernando Antonioli (luisfernandoantonioli@gmail.com))

PAD: Anderson Cotrim (ander.cotrim@gmail.com)

1 Descrição Geral

Neste laboratório, continuaremos construindo as classes base de nosso sistema de caronas online, cujo objetivo é propor uma solução de busca e oferecimento de caronas on-line capaz de conectar pessoas que fazem o mesmo trajeto para o trabalho, escola, faculdade ou eventos de lazer, através de uma rede social.

2 Objetivo

O objetivo desta atividade consiste na familiarização do aluno com a construção de classes que possuam atributos e métodos estáticos, bem como a utilização de sobrecarga de métodos e de construtores.

3 Atividade

3.1 Definições

- Caroneiro: Aquele que necessita de carona
- Caronante: Aquele que oferece a carona

Nesta atividade, adicionaremos ao projeto do Lab1 mais duas classes: Caroneiro e Caronante. Estas classes contém atributos e métodos estáticos capazes de auxiliar, futuramente, no cômputo de estatísticas do sistema. Na figura 1 apresentamos o diagrama UML destas classes.

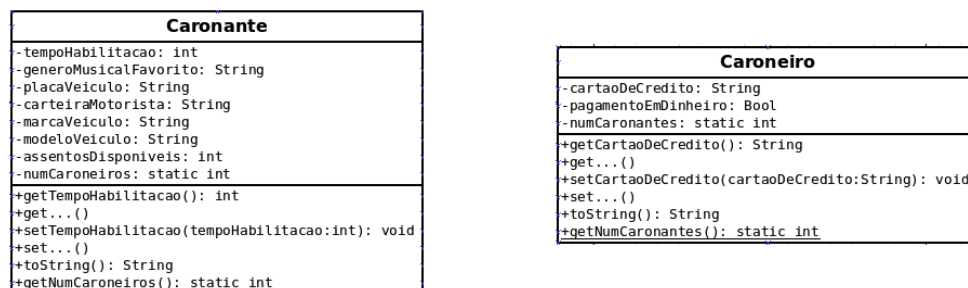


Figura 1: Diagrama UML das classes

Implemente as novas classes conforme apresentado no diagrama UML.

3.2 Observações

- Apesar do sistema ser de caronas, as mesmas poderão ser cobradas. Informações sobre cartão de crédito ou o dinheiro foram colocados no sistema a fim de dividir os custos da viagem, como pedágios e combustível, por exemplo.
- A classe Caroneiro necessitará de três construtores: um para os clientes que desejam informar seu cartão de crédito no sistema, outro para os clientes que preferem pagamento em dinheiro e um construtor sem parâmetros. A classe Caronante deverá necessita apenas de um construtor sem parâmetros.

Após a criação de cada classe do diagrama UML, crie um `main` onde você instanciará objetos do tipo Caroneiro e Caronante, utilizando cada um dos construtores criados anteriormente. Você deve utilizar `System.in` para fazer com que a instanciação das classes seja interativa, ou seja, o usuário deverá definir os valores dos parâmetros via entrada padrão. Em seguida temos um exemplo de como deve ser feito

```
1 // — dentro da classe Main
2 public static void main(String[] args) {
3     // instanciando objetos
4     Scanner sc=new Scanner(System.in);
5
6     System.out.println("[Dados do caronante]");
7     System.out.println("Tempo de habilitacao: ");
8     int tempoHabilitacao = sc.nextInt();
9
10    ...
11
12    Caronante caronante1 = new Caronante(tempoHabilitacao , ...);
13
14    ...
15
16    // impressao dos dados dos objetos
17    System.out.println("Caronante 1:\n"+caronante1);
18    // ...
19 }
```

print.java

Posteriormente, você deve imprimir os dados de cada instância criada utilizando o método `toString`, como realizado no laboratório anterior.

4 Questões

Sobre a atividade realizada, responda como comentário no início do código da classe que contém o `main`.

- É possível escrever uma classe sem escrever nenhum construtor? Por quê?
- Um método estático pode acessar uma variável (atributo) não estático da classe? Por quê?
- Um método não estático pode acessar uma variável (atributo) estático da classe? Por quê?

5 Submissão

Para submeter a atividade utilize o Moodle (<https://www.ggte.unicamp.br/ea>). Salve os arquivos dessa atividade em um arquivo comprimido no formato .tar.gz ou .zip e nomeie-o **Lab2-000000.[tar.gz | zip]** trocando '000000' pelo seu número de RA. Submeta o arquivo na seção correspondente para esse laboratório no moodle da disciplina MC302.

Datas de entrega

- Dia **19/03** Turma **ABCD** até às 23:55