

# MC322 – Programação Orientada a Objetos

## Laboratório 10 – 2s2021

### TESTE PRÁTICO 3

Leonardo Montecchi (Professor)

Thales Eduardo Nazatto  
Leonardo de Sousas Rodrigues

Ângelo Renato Pazin Malaguti

Para perguntas ou dúvidas usem o Discord (<https://discord.gg/Xzpz9epNTG>)

## 1 Submissão

### Data de entrega

- 31/10/2021 até às 23h59.

### Submissão

- Ao criar o projeto Java no Eclipse, selecionar a versão **JavaSE-11** no JRE
- IMPORTANTE: Nomear o projeto na forma **RA\_Lab10** e o pacote base na forma **com.unicamp.mc322.lab10**. Substitua RA com o seu *Registro Acadêmico* (matrícula).
- Submeta o trabalho no link de entrega na página do Classroom da disciplina, em formato de arquivo compactado (zip) com o nome **RA\_Lab10.zip**. Substitua RA com o seu *Registro Acadêmico* (matrícula). Entregas feitas de outras formas não serão consideradas.
- O arquivo compactado deve conter o projeto inteiro (“File / Export” no Eclipse, ou crie o arquivo manualmente).

### Critérios de avaliação

- **Este laboratório VALE nota.**

Faz parte da avaliação saber quais classes devem ser criadas e em quais classes cada método deve ser implementado. Em particular, o código será avaliado de acordo com os seguintes aspectos:

1. Definição de Classes (30%).
2. Aplicação de princípios de Programação Orientada a Objetos (POO) (30%).
  - Ex: *encapsulamento* de atributos, *responsabilidades* de classes.
3. Funcionalidades implementadas (40%).

## 2 Objetivo

Este laboratório é a avaliação prática da unidade 3, sobre os conceitos de associação, agregação e composição. O objetivo é implementar uma plataforma similar a Uber com base na especificação abaixo.

## 2.1 Especificação do Sistema

Desenvolva um software que seja capaz de gerenciar corridas, alocando motoristas para passageiros. O software deve conter as seguintes funcionalidades:

- O software deverá armazenar as informações do usuário, como nome, CPF e data de nascimento. Além disso, deve armazenar número do cartão de crédito caso o usuário seja passageiro; número da habilitação, número do cartão de crédito e uma lista de veículos caso o usuário seja motorista.
- O software deverá ser capaz de cadastrar veículos, mantendo informações placa e ano de fabricação. Deve conter também uma referência para o proprietário do veículo. Podem existir dois tipos de veículos, os comuns e os de luxo.
- O software deverá gerenciar viagens. Para simplificar pode assumir que todas as viagens são feitas em linha reta. Cada viagem pode ter várias paradas ao longo do caminho, conforme o número de paradas é maior, o preço da viagem também aumenta. Pode haver também várias pessoas na mesma viagem, e em cada parada pode subir ou descer um ou mais passageiros.
- Na criação de uma viagem, deverão ser informados: passageiro(s), motorista, carro, e distância a ser percorrida. Eventuais paradas adicionais devem ser informadas separadamente, indicando em quais pontos da viagem as paradas deverão ser feitas e eventualmente passageiros que vão descer ou subir no carro.
- O preço da viagem tem um custo fixo dado o tipo de veículo e varia conforme a distância percorrida. Para veículos comuns o preço fixo é R\$ 3,00 e para os de luxo é R\$ 7,00. Além disso, para cada 100 metros percorridos, veículos comuns cobram R\$ 2,00 e veículos de luxo cobram R\$ 3,50 a mais. A cada parada o valor de R\$ 1,50 deverá ser adicionado na viagem de um carro comum e um valor de R\$ 2,70 deverá ser adicionado na viagem de um carro de luxo.

## 2.2 Fluxo de Execução

- **Configure o software para fazer uma execução padrão mostrando as funcionalidades implementadas**, de forma parecida a quanto feito nos laboratórios anteriores.
- **Não é necessário criar um menu de interação com o usuário**

Considere o seguinte cenário de exemplo:

- Crie os usuários: Marcos, com CPF 145678798, e data de nascimento 15/07/1998, passageiro, número do cartão 369874. Maria, com CPF 248679108, e data de nascimento 12/02/1997, motorista, habilitação 987654, número do cartão 483530, e possui dois carros, um de ano 2009 placa ABC-1234 modelo luxo, e um ano 2013 placa OOP-2020 modelo comum. João, com CPF 654973652, e data de nascimento 03/01/2002, passageiro, número do cartão 785632.
- Faça uma viagem de Marcos com Maria, em seu carro de luxo, distância de 500 metros, com duas paradas.
- Faça uma viagem de Marcos e João com Maria, em seu carro comum, distância de 2 km, com 5 paradas, em que João terminará sua viagem na terceira parada.
- Faça uma viagem de João com Maria em seu carro de luxo, distância de 700m, 3 paradas.

No final de cada viagem, imprima na tela um resumo (summary) de cada viagem. Com informações como passageiro(s), motorista, distância percorrida, paradas e preço final da viagem.

### 3 Boas Práticas

- Nomes de classes devem começar com letra maiúscula;
- Nomes de variáveis e métodos devem começar com letra minúscula;
- Nomes de classes devem ser substantivos;
- Nomes de métodos devem ser verbos ou começar com verbo;
- Nomes compostos por mais de uma palavra devem ser escritos na forma *CamelCase*. Isto é, com a primeira letra de cada palavra maiúscula. Por exemplo: “get something” deve ser escrito como `getSomething` (método), e uma classe que representa *something* poderia ser declarada como `Something`.