Universidade Estadual de Campinas
Instituto de Computação

Tarefa 1

Modelando um Sistema de Táxi

MC322 – Programação Orientada a Objetos

Professor: Marcelo Reis
PEDs: Caio Rhoden / Gilson Soares
PADs: Felipe Araujo / João Takaki

Campinas, 26 de agosto de 2024

---

1. Descrição Geral

Nesta primeira atividade de programação, vamos colocar em prática conceitos vistos na primeira aula teórica, tais como superclasses e subclasses, classe abstrata, herança, polimorfismo, encapsulamento e *overriding*, num probleminha de modelagem de um sistema de táxi. Além disso, vamos desenhar e implementar o sistema utilizando o método de *pair programming* (atividade em duplas), e utilizaremos três pontos importantes vistos na última aula:

- Separar adequadamente interface de implementação;
- Pensar de forma abstrata no problema ("o que tem que fazer" vs. "como fazer");
- Dar o mínimo possível de interface ao usuário programador (e ir fornecendo métodos públicos conforme a necessidade, numa abordagem "bottom-up").

2. Objetivos

Os principais objetivos deste laboratório consistem em:

- Prosseguimento na familiaridade com o ambiente de desenvolvimento (IDE) Visual Studio Code e a linguagem a ser utilizada ao longo desta disciplina (Java);
- Desenvolvimento das classes e seus respectivos atributos e métodos;
- Desenvolvimento da visibilidade dos atributos e métodos (i.e., a separação adequada entre a interface e a implementação).

3. Atividades

As seguintes atividades são esperadas nesse laboratório:

- Criação de um projeto na IDE de sua escolha: dê preferência para o VSCode como IDE; contudo, outras IDEs (Eclipse, Sublime, Atom, etc) podem ser utilizadas;
- Elaboração das classes `Taxi`, `Cabbie` e `Passenger`: seu projeto deve compreender ao menos essas classes e uma classe `Main` para a execução do projeto;
- Definição dos métodos, atributos, bem como suas respectivas visibilidades: você deve definir qual visibilidade (*public, private, protected*) os atributos e métodos possuem;
- Métodos de acesso (*getters e setters*): defina os métodos de acesso aos atributos das classes;
- Métodos de impressão (`toString()`): para visualizar a classe instanciada e seus atributos;

4. Descrição do sistema a ser modelado

O sistema de táxi proposto descreve a dinâmica de uma corrida típica de táxi, situação que envolve três objetos: táxi (veículo), taxista e passageiro. Para utilizar um táxi, um passageiro precisa realizar os seguinte passos e/ou atender aos seguintes requisitos:

- Ter um lugar para ir;
- Chamar um táxi;
- Entrar no táxi;
- Dizer ao taxista aonde ele quer ir;
- Pagar o taxista;
- Dar ao taxista uma gorjeta (opcional);
- Sair do táxi.

Como dito acima, o sistema é composto por três objetos: um táxi, um taxista e um passageiro. Objetos dos três tipos devem ser instanciados no método `static main` da classe `Main`, e trocas de mensagens entre esses objetos devem ser realizadas para efetuar os passos acima. Para cada passo realizado, uma troca de mensagens adequadas entre objetos precisa ocorrer, sendo que a mesma será indicada através de uma mensagem impressa na saída padrão (utilizando o método `System.out.println`). Um exemplo é ilustrado abaixo com uma passageira chamada Alice, que faz uma corrida para Ribeirão Preto e não dá gorjeta ao taxista:

```
Alice define o destino (Ribeirão Preto)
Alice chama um táxi
Taxista atende ao chamado
Alice entra no táxi
Alice informa ao taxista o destino
Taxista dirige até o destino
Alice paga o taxista (sem gorjeta)
Alice sai do táxi
```

4.1. Esquema das classes

Para servir de referência para o início da atividade, segue um exemplo de declaração das principais classes (fora a classe Main), sem ainda a maioria dos atributos e métodos que serão implementados para o desenvolvimento do sistema.

```java
// Taxi.java
public class Taxi {
    private String destination;
    private boolean isHailed;
    public Taxi() {
        this.destination = "";
        this.isHailed = false;
    }
}
```

```java
// Person.java
public abstract class Person {
    public abstract void performRole();
};
```

```java
// Cabbie.java
public class Cabbie extends Person {
    private int money;
    public Cabbie() {
```

```
        this.money = 0;
    }
}
```

```
// Passenger.java
public class Passenger extends Person {
    private String destination;
    private String name;
    public Passenger(String a_name) {
        this.destination = "";
        this.name = a_name;
    }
}
```

5. Avaliação

Além da correta execução do programa (i.e., sem erros ou warnings de compilação e de execução), os seguintes critérios serão utilizados para a composição da nota do laboratório:

- Entrega realizada dentro do prazo estipulado;
- Entrega realizada de todos os arquivos do projeto Java compactados em um único arquivo zip, feita somente por um dos membros da dupla.

Qualidade do código desenvolvido, tanto do ponto de vista funcional quanto de documentação (e.g., tabulação, comentários) será cobrada a partir da Tarefa 2.

6. Entrega

- **A entrega do Laboratório é realizada exclusivamente via Google Classroom** [1];
- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório;
- Em todos os arquivos, colocar o nome completo e o RA de cada membro da dupla.
- **Prazo de Entrega:** 02/09 (segunda-feira), às 23:59, pelo Google Classroom.

[1] classroom.google.com/u/1/c/Njg0OTY4NDI2OTUz