



Laboratório 1

Monopoly

MC322 - Programação Orientada a Objetos

Professora: Esther Colombini

PEDs: Cristiano Gabriel de Souza Campos / Wladimir Arturo
Garces Carrillo



1. Descrição Geral

Monopoly é um dos jogos de tabuleiro mais populares do mundo, em que propriedades como bairro, casas, hotéis, empresas são compradas e vendidas, enquanto jogadores adquirem diversos recursos e outros vão à falência. No Brasil, inspirou o jogo Banco Imobiliário. Existem mais de 1.200 edições do Monopoly licenciadas oficialmente em todo o mundo, cada uma oferecendo um toque único no jogo clássico. Essas edições vão desde franquias e cidades populares até edições especiais comemorando aniversários ou eventos

1.1. Regras do Jogo

As regras do jogo seguirão as regras oficiais, como descrito em <https://estrela.vteximg.com.br/arquivos/Manual-Banco-Imobiliario-App.pdf>. Entretanto, cada um é livre para definir sua versão do Banco, desde que as regras sejam seguidas. As cartas de sorte/revés e os locais (tipos e valores) devem ser definidos pela pessoa e dever ser passíveis de reconfiguração.

No Jogo existem:

- 1 Banco detentor inicialmente de:
 - 32 casas
 - 12 hotéis
 - 28 títulos de propriedades
 - 30 cartões SORTE/REVÉS
 - 380 notas (Verificar quantidades de cada tipo. Subtrair a quantidade entregue a cada jogador)
- 2 a 6 jogadores
- 2 dados (com valores de 1-6)
- 1 tabuleiro contendo 40 locais, sendo que um local pode ser:
 - Uma propriedade - que pode ser adquirida pelo jogador (são 28 no total), sendo que cada propriedade pode ser:
 - um terreno (com até 4 casas ou 1 hotel) que possui valor para aluguel simples, com casas, hotel e hipoteca
 - um negócio (companhia de táxi, aeroporto, etc) que possui valor básico a ser pago ao proprietário
 - Um local de utilidade (1 lucro/dividendos e 1 imposto de renda)
 - Um local de sorte/revés (6 no total)
 - Um ponto de passagem (1 prisão, 1 parada livre, 1 vá para prisão e 1 ponto de partida)

Uma partida tem de 2 a 6 jogadores e cada jogador recebe inicialmente:

- 8 notas de \$1
- 10 notas de \$5
- 10 notas de \$10

- 10 notas de \$50
- 8 notas de \$100
- 2 notas de \$500

Todos os jogadores partem do mesmo ponto de início. A ordem das jogadas deve ser realizada por sorteio. Os valores obtidos pelos dados devem ser valores aleatoriamente sorteados. O mesmo acontece para a carta de sorte/revés retirada pelo jogador.

A decisão de comprar deve seguir os limites do jogo, mas cada um é livre para implementar o seu próprio modelo de decisão. O mesmo acontece para vendas, trocas e hipotecas.

Se um jogador falir, é preciso removê-lo da partida. O número de jogadores da partida deve ser um parâmetro.

Com base nessas informações, o tópico a ser trabalhado para este semestre na disciplina de Programação Orientada a Objetos (MC322) é a modelagem de um jogo de Monopoly. Ao longo dos laboratórios, seu sistema deverá ser incrementado para comportar novas funcionalidades à medida que o seu conhecimento for ampliado.

2. Lab 01

Neste primeiro laboratório, você deve implementar o sistema de cadastro de cartas de sorte, jogador e peça. Sendo assim, inicialmente o sistema deve compreender quatro (4) classes: Peça, CartaSorte, Jogador e Main. A Figura 1 apresenta um diagrama de classes das classes a serem implementadas.

Um diagrama de classes é um tipo de diagrama UML (*Unified Modeling Language*). Esse diagrama representa a estrutura de um sistema orientado a objetos, demonstrando assim classes, interfaces, propriedades, métodos e suas associações dentro do sistema. O objetivo principal do diagrama de classes é fornecer uma visão geral da arquitetura do sistema, mostrando como as diferentes partes do sistema se relacionam entre si.

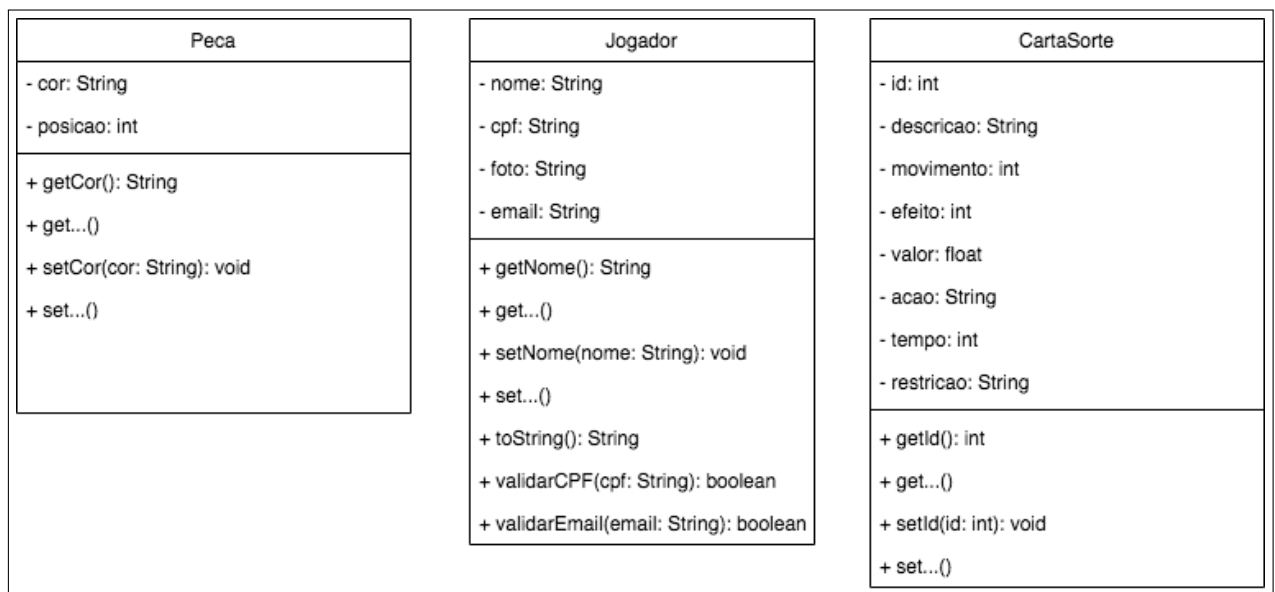


Figura 1: Diagrama de Classe

3. Objetivos

Os principais objetivos deste laboratório consistem em:

- Reconhecimento e familiaridade com o ambiente de desenvolvimento (IDE) e a linguagem a ser utilizada nesta disciplina (Java);
- Desenvolvimento das classes e seus respectivos atributos e métodos;
- Desenvolvimento da visibilidade dos atributos e métodos.

4. Atividades

As seguintes atividades são esperadas nesse laboratório:

- Criação de um projeto na IDE a sua escolha: Dê preferência para o Eclipse como IDE, contudo, outras IDEs (VSCode, Sublime, Atom, etc) podem ser utilizadas;
- Elaboração das classes **Peça**, **Jogador**, **CartaSorte**: seu projeto deve compreender estas três classes além de 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.
- Método *validarCPF()*: defina o método de validar o CPF de um jogador. Para isso você pode seguir os seguintes passos:
 1. Remova todos os caracteres não numéricos do CPF usando o método **replaceAll**;
 2. Verifique se o CPF tem 11 dígitos. Se não tiver, retorne `false`;
 3. Verifique se todos os dígitos são iguais. Se forem, retorne `false`;
 4. Calcule os dígitos verificadores usando o algoritmo apropriado;
 5. Verifique se os dígitos verificadores calculados são iguais aos dígitos verificadores do CPF. Se forem, retorne `true`. Caso contrário, retorne `false`.
- Método *validarEmail()*: defina o método de validar o email de um jogador. Você pode usar métodos disponíveis na literatura para esse fim.

5. Descrição das Classes

Classe - Peça

Classe responsável por armazenar as informações das peças do jogo. A classe é composta pelos seguintes atributos: `cor` e `posição`, do tipo `String` e `inteiro`, respectivamente.

Classe - CartaSorte

Essa classe compreende as informações da carta de sorte. A classe contém os seguintes atributos:

1. Identificador único (`id`): Um identificador único para cada carta, o que pode ser útil para identificar cartas específicas ou para referência durante o jogo;
2. Descrição: o nome da carta;
3. Movimento no tabuleiro: Para cartas que instruem os jogadores a se moverem para uma determinada casa no tabuleiro;
4. Efeito (`positivo`, `negativo`, `neutro`): Um atributo que indica se o efeito da carta é positivo, negativo ou neutro;
5. Valor do pagamento/recebimento: Para cartas que envolvem dinheiro, como "Pague X ao banco." ou "Receba X do banco";
6. Ação específica: Para cartas que têm um efeito único e específico, como "Construa uma casa em uma de suas propriedades sem custo".
7. Tempo de aplicação: Alguns efeitos podem ter condições de ativação, como "Use esta carta imediatamente" ou "Guarde esta carta e use-a mais tarde";
8. Restrições de uso: Alguns efeitos podem ter restrições, como "Esta carta só pode ser usada se você estiver na prisão".

Classe - Jogador

A classe Jogador tem como objetivo ter as informações do jogador. Os atributos dessa classe são: nome, cpf, email e foto. Todos os atributos são do tipo String. A foto deve guardar o caminho para a foto.

```
1 public class Jogador {
2     private String nome;
3     private String cpf;
4     private String email;
5     private String foto;
6
7     // Construtor
8     public Jogador(String nome, String cpf, String email, String foto) {
9         this.nome = nome;
10        this.cpf = cpf;
11        this.email = email;
12        this.foto = foto;
13    }
14
15    // Getters e setters
16    public String getNome() {
17        return nome;
18    }
19
20    public void setNome(String nome) {
21        this.nome = nome;
22    }
23
24    public String getCpf() {
25        return cpf;
26    }
27
28    public void setCpf(String cpf) {
29        this.cpf = cpf;
30    }
31
32    public String getEmail() {
33        return email;
34    }
35
36    public void setEmail(String email) {
37        this.email = email;
38    }
39
40    public String getFoto() {
41        return foto;
42    }
43
44    public void setFoto(String foto) {
45        this.foto = foto;
46    }
47 }
48 }
```

Listing 1: Jogador.java

Classe Main

Para um programa Java funcionar é requerida a existência de um método main que serve de ponto de partida para o programa ser inicializado. Crie uma nova classe chamada Main e escreva o método main.

Na função main realize a instanciação de alguns objetos do tipo Jogador, CartaSorte ou Peça (pelo menos um de cada) com valores de atributos quaisquer conforme sua imaginação. Após instanciar os objetos, imprima seus dados utilizando o método System.out.println(). Veja o exemplo a seguir:

6. Avaliação

Além da correta execução do laboratório, os seguintes critérios serão utilizados para a composição da nota do laboratório:

- Entrega realizada dentro do prazo estipulado;
- Qualidade do código desenvolvido (tabulação, comentários);

- Instanciação dos objetos na classe **Main**;
- Desenvolvimento correto dos métodos de acesso.

7. Entrega

- **A entrega do Laboratório é realizada exclusivamente via Github**¹ Para a submissão no Github, gere um release (tag) com a identificação do laboratório.
- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório.
- **Prazo de Entrega:** 24/03/24 - 23h59.

¹Você deve criar um link e enviar no Google Classroom