

Documentacao Trabalho Pratico 1 Programação Modular

Nélio Cezar Muniz Sampaio, Louise Saturnino

Abril 2017

1 Introdução

O banco imobiliário - Monopoly, em inglês- é um jogo de tabuleiro em que a dinâmica se baseia na compra e venda de propriedades, pagamento de aluguéis e etc. O jogo foi criado nos Estados Unidos em 1935 por Charles B. Darrow, e veio para o Brasil em 1944 através da empresa Brinquedos Estrela. O sucesso alcançado, tornou Darrow multi-milionário, e, no Brasil, é o brinquedo mais bem sucedido em vendas no país, além de que estima-se que seja o jogo mais vendido no mundo.

Neste trabalho, o objetivo é recriar o jogo virtualmente como forma de aplicar os conhecimentos adquiridos na disciplina de Programação Modular pelos alunos de graduação em ciência da computação da Universidade Federal de Minas Gerais. A Modelagem do jogo será feita através da linguagem de programação Java, utilizando-se dos conceitos de Orientação a Objetos.

O programa recebe como entrada dois arquivos. O primeiro arquivo descreve o tabuleiro do jogo, contendo o número de posições e, para cada posição, o tipo (começo do jogo (start), passa a vez e imóvel), o tipo do imóvel, valor do imóvel e taxa de aluguel. O segundo arquivo contém o número de jogadores e o saldo inicial dos jogadores, além das jogadas. Como saída, o programa gera um outro arquivo que mostra as estatísticas do jogo simulado.

2 Implementação

O projeto foi implementado utilizando a linguagem de programação Java, utilizando conceitos de Orientação a objetos. Assim, para descrever a implementação do trabalho, será descrito a seguir as classes criadas e seus principais métodos.

2.1 Classe Tabuleiro

A classe tabuleiro é responsável por armazenar o tabuleiro que será utilizado no jogo, assim como as operações relacionadas à ele.

2.1.1 Principais métodos

- void leTabuleiro (String arquivo) : Faz a leitura do tabuleiro do arquivo cujo nome é recebido por parâmetro. No trabalho prático, a leitura do tabuleiro deve ser sempre feita do arquivo "tabuleiro.txt", mas o método foi mantido recebendo o nome do arquivo por parâmetro para obter-se reutilização para possíveis alterações futuras;
- void sort () : Esse método ordena os imóveis do tabuleiro. Isso é necessário pois, no arquivo de entrada, as posições não estão necessariamente ordenadas;

2.2 Classe Imovel

Essa classe é responsável por armazenar as informações particulares de cada posição do tabuleiro.

2.3 Classe Jogador

Representa os jogadores de cada partida. Contém informações como saldo, posição e imóveis, que são utilizadas durante a partida, e outras como número de voltas, total recebido de aluguel, total de aluguel pago, gasto com compra de imóveis e números de passa a vez, que são informações apresentadas nas estatísticas do jogo.

2.4 Classe Jogo

Essa classe é responsável pela execução do jogo e guardar as informações referentes à ela, como a lista de jogadores e o tabuleiro.

2.4.1 Principais Métodos

- void leTabuleiro() : Chama o método de leitura do objeto tabuleiro;
- void startGame() : Faz a leitura das informações do jogo e processa as jogadas provenientes do arquivo "jogadas.txt";
- void imprimeEstatistica() : Imprime os dados de saída (estatísticas do jogo);

2.5 Outras estruturas

2.5.1 ArrayList

A estrutura ArrayList do Java foi utilizada com o objetivo de facilitar a implementação onde era necessário organizar um conjunto de dados como na lista de jogadores, na classe Jogo, e na lista de posições, na classe Tabuleiro. Essa estrutura foi escolhida por permitir fácil manipulação dos dados nela armazenados.

3 Testes Práticos

Logo abaixo segue um exemplo de teste realizado com o programa gerado:

3.1 Arquivos de Entrada

3.1.1 Tabuleiro - "tabuleiro.txt"

```
1 10
2 1;1;1
3 2;2;3;2;150;20
4 3;3;3;1;100;10
5 4;4;3;4;350;30
6 5;5;3;1;100;10
7 6;6;3;2;150;20
8 7;7;3;3;100;10
9 8;8;3;5;500;10
10 9;9;2
11 10;10;3;1;100;10
```

Figure 1: Arquivo de entrada "tabuleiro.txt"

3.1.2 Jogadas - "jogadas.txt"

```
1 13%3%3000
2 1;1;3
3 2;2;4
4 3;3;6
5 4;1;3
6 5;2;2
7 6;3;1
8 7;1;1
9 8;2;4
10 9;3;4
11 10;1;6
12 11;2;3
13 12;3;4
14 DUMP
```

Figure 2: Arquivo de entrada "jogadas.txt"

3.2 Arquivo de Saída

3.2.1 Estatísticas - "estatisticas.txt"

```
1 1:4
2 2:1-1;;2-1;;3-1;
3 3:1-3195;2-3285;3-2670
4 4:1-105;2-0;3-70
5 5:1-60;2-115;3-0
6 6:1-350;2-100;3-900
7 7:1-0;2-0;3-0
8
```

Figure 3: Arquivo de saída "estatisticas.txt"

4 Conclusão

Foram aplicados no projeto diversos conceitos de orientação a objetos, tais como herança, interface, sobrescrita, etc. Assim, podemos dizer que o trabalho cumpriu seu objetivo possibilitando a prática dos conceitos estudados e o contato com as dificuldades, tanto em implementá-los quanto por não terem sido devidamente aplicados.

O projeto também ocasionou o contato com uma linguagem de programação poderosa e e demandada no mercado, sendo assim uma oportunidade de conhecê-la melhor e, a partir de agora, tê-la como uma possível solução para problemas futuros.

References

- Angelfire.com. (2017). *Banco imobiliário — uma história de sucesso?* Retrieved from <http://tinyurl.com/m9sr6v4>
- Devmedia.com.br. (2017). *Criando e gravando dados em txt com java*. Retrieved from <http://tinyurl.com/kd98bjf>
- Guj.com.br. (2017). *Dúvida em como criar um arquivo txt em java*. Retrieved from <http://tinyurl.com/m6j5v39>
- stackoverflow.com. (2017). Retrieved from <http://tinyurl.com/mmyqb49>
- Wikipedia.com. (2017). *Banco imobiliário*. Retrieved from <http://tinyurl.com/nxofg7a>
- Youtube.com, E. R. D. P. . (2017). *Tutorial 25 java netbeans - ordenar objetos list, arraylist, linkedlist*. Retrieved from <http://tinyurl.com/kbyaapz>