Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Programação Modular
1º Semestre de 2017
Prof. Douglas G. Macharet

Trabalho Prático 1 – Banco Imobiliário Modular

Valor: 10 pontos
Data de entrega: 17/04/2017

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos da Orientação a Objetos e com a programação utilizando a linguagem Java. Para realização do trabalho o aluno deverá implementar um jogo (Banco Imobiliário) seguindo os princípios da Orientação a Objetos.

1. Instruções

Considerando os arquivos de entrada apresentados nas próximas seções, o aluno deverá implementar um código fonte que construa um tabuleiro do Banco Imobiliário e processe uma sequência de jogadas.

O tabuleiro é composto de uma sequência de posições, sendo que em cada posição do tabuleiro será associado um determinado imóvel. Deve ser implementada uma estrutura em memória que possa armazenar o imóvel de cada posição. Sendo que os imóveis podem ser dos seguintes tipos:

- Residência
- Comércio
- Indústria
- Hotel
- Hospital

Por meio da leitura de um arquivo (tabuleiro.txt), disponibilizado na mesma pasta de execução do programa, deverá ser construído um tabuleiro que tem para cada posição um imóvel pré-determinado. Esse arquivo ainda descreve para cada imóvel um valor de compra e uma taxa de aluguel. Após a construção do tabuleiro, seu sistema deverá ler um segundo arquivo (jogadas.txt), também disponibilizado na mesma pasta de execução do programa. Esse arquivo descreve as instruções que correspondem às jogadas dos participantes no Banco Imobiliário. O arquivo jogadas.txt indicará os jogadores e seus respectivos números tirados em um dado de seis faces. Cada linha deste arquivo será uma jogada.

Seu sistema deverá implementar as regras de negócios básicas do jogo, a fim, de computar as jogadas dos participantes para o tabuleiro construído e dar continuidade à execução da partida. Além disso, o sistema deve tratar possíveis casos de exceção simples, por exemplo, tentativa de retirar a taxa e aluguel de um jogador com saldo inferior a taxa cobrada.

O programa deverá ser feito baseado no JAVA SDK 8. Para guardar as informações relativas às coleções, você pode utilizar uma das classes que implementa a interface `Collection<E>`. Você pode obter mais informações em: http://docs.oracle.com/javase/8/docs/api/java/util/Collection.html.

1.1. Funcionalidades

Resumo das funcionalidades esperadas do sistema:

1. Criar os objetos necessários para construção do tabuleiro;
2. Implementar as regras do jogo necessárias computar as jogadas dos participantes;
3. Imprimir as estatísticas do jogo.

1.2. Regras do Jogo

O jogo a ser implementado tem algumas simplificações se comparado à sua versão de tabuleiro original. A seguir são descritas algumas regras específicas:

1. Cada imóvel inicialmente pertence ao Banco;
2. O Banco tem valor infinito em caixa;
3. Caso a posição do tabuleiro que o jogador caiu, mediante o número lido do arquivo de jogadas, for um imóvel do Banco a ação será: **jogador deverá comprar o imóvel.** Desta forma, o jogador deverá pagar para o banco a quantia referente ao imóvel (desde que possua saldo suficiente).
4. Caso o imóvel seja de outro jogador a ação será: **jogador deverá pagar o valor do aluguel**. Desta forma, o jogador deverá pagar para outro jogador, dono do imóvel, a quantia referente ao aluguel do imóvel. Se o jogador cair em um imóvel que seja dele próprio não deverá pagar e nem receber nada.
5. Caso o jogador tenha caído em uma posição do tabuleiro que tem um passe a vez, o jogador não paga nada e uma nova instrução deverá ser lida;
6. Se o jogador tiver saldo menor que zero, as jogadas do arquivo de Jogadas correspondente a ele deverão ser ignoradas (jogador não está mais participando da partida). Nesse caso, suas propriedades retornam ao Banco e podem ser compradas pelos outros jogadores;
7. Cada vez que um jogador passar pela a posição inicial deverá receber um valor de 500 reais do Banco;
8. O sistema deverá terminar o jogo quando restar apenas um jogador com saldo positivo, ou quando for lido no arquivo de entrada a instrução **DUMP**;
9. Após o jogo ser terminado, os dados estatísticos deverão ser impressos em um arquivo de saída.

2. Arquivos de Entrada e Saída

2.1. Exemplo dos arquivos de entrada:

- **Tabuleiro:** Este arquivo indica como será o tabuleiro do jogo. Uma instância de exemplo do tabuleiro é descrita na Figura 1.

tabuleiro.txt

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

Figura 1 - Instância de um tabuleiro.

A primeira linha do arquivo corresponde ao número de posições que o tabuleiro deverá ter. As demais linhas descrevem qual o tipo da posição: 1) Start, 2) Passe a vez e 3) Imóvel.

Os campos de cada linha são separados por ponto e vírgula (**;**). Com exceção da primeira linha, as demais linhas são organizadas da seguinte forma:

1º. **Id**: é um identificador de cada linha no arquivo que figura de **1** até **n**.
2º. **Posição**: corresponde a um identificador da posição que a especificada deve ocupar no tabuleiro, indo de **1** até a posição **n**. As posições não são necessariamente descritas em ordem crescente.
3º. **Tipo da posição:** Start (1), Passe a vez (2) e Imóvel (3).
4º. **Tipo do imóvel**: Corresponde ao tipo do imóvel que pertence a posição, conforme a Tabela 1.
5º. **Valor do imóvel**: Corresponde ao valor do imóvel para compra;
6º. **Taxa do aluguel**: Corresponde à taxa de aluguel do imóvel. Esse valor é descrito em porcentagem. Ou seja, será a porcentagem do valor do imóvel.

Tabela 1 Tipos de Imóveis

Imóvel	Identificador
Residência	1
Comércio	2
Indústria	3
Hotel	4
Hospital	5

- **Jogadas:** Este arquivo informa as jogadas de cada participante do jogo. Uma instância de jogadas é descrita na Figura 2.

A primeira linha do arquivo corresponde ao número de instruções de jogadas que o arquivo contém, o número de jogadores participantes e o valor inicial que cada jogador começa o jogo. Essas informações são separadas pelo caractere (**%**).

As demais linhas correspondem as jogadas de cada jogador. Sendo que, estas linhas tem três campos separados por ponto e vírgula (**;**). Os campos são organizados da seguinte forma:

1º. **Id:** é um identificador de cada linha no arquivo que vai de 1 até n.
2º. **Id do jogador:** corresponde ao identificador do jogador.
3º. **Valor do dado:** indica o valor do dado de seis faces, que o jogador tirou na jogada.

jogadas.txt

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

Figura 2 - Instância de jogadas.

A Figura 3 demostra a execução das instâncias das Figuras 1 e 2. Os jogadores são representados pelo indicativos J1, J2 e J3 representando o Jogador 1, Jogador 2 e Jogador 3 respectivamente.

Posição tabuleiro	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tipo	1	3 (2)	3 (1)	3 (4)	3 (1)	3 (2)	3 (1)	3 (5)	2	3 (1)
Rodada 1				J1	J2		J3			
Rodada 2							J1-J2	J3		
Rodada 3	J2	J3						J1		
Rodada 4				J1-J2		J3				

Figura 3 - Exemplo de execução das instâncias das Figuras 1 e Figura 2.

2.2. Exemplo dos arquivos de saída:

O arquivo de saída será relativo apenas às estatísticas do jogo. Ao logo do jogo deverão ser computadas informações para responder as seguintes questões:

1. Quantas rodadas o jogo teve?
2. Quantas voltas foram dadas no tabuleiro por cada jogador?
3. Quanto de dinheiro cada jogador ficou (colocação)?
4. Qual foi a quantidade de aluguel recebida por cada jogador?
5. Qual foi o valor pago de aluguel por cada jogador?
6. Qual foi o valor gasto na compra de imóveis por cada jogador?
7. Quantos “passa a vez” cada jogador teve?

No arquivo de saída deverão ser informadas as linhas referente as questões acima citadas. Em cada linha o número da questão deverá ser separado da resposta por meio do caractere dois pontos (**:**). A Figura 4 demostra uma instância dos dados estatísticos computados para o jogo descrito nas Figura 1 e Figura 2.

estatística.txt

```
1:4
2:1-1;2-1;3-1
3:1-3195;2-3185;3-2670
4:1-105;2-0;3-70
5:1-60;2-115;3-0
6:1-350;2-100;3-900
7:1-0;2-0;3-0
```

Figura 4 - Instância de Dados Estatísticos.

3. Documentação

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado.
4. Testes: descrição dos testes realizados e listagem da saída (não edite os resultados). Você pode propor outros testes além dos fornecidos com o enunciado.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

3.1. O que deve ser entregue:

Envie um arquivo ZIP com o nome no formato 'tp1-primeironome1-primeironome2.zip', contendo os seguintes arquivos:

- Arquivo README.txt com os nomes completos dos alunos da dupla.
- O código fonte do programa em Java bem endentado e comentado. Deve ser fornecido junto com o fonte um arquivo Makefile com as opções 'make' e 'make run'.
- A documentação do trabalho bem escrita e detalhada.

3.2. Comentários gerais

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, endentação e comentários no programa também serão avaliados.
- O trabalho deverá ser feito em dupla.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.

3.3. Critérios de avaliação:

- Funcionamento correto (3 pts).
- Uso correto dos conceitos de OO (5 pts).
- Documentação (2 pts).