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

Programação Modular
1º Semestre de 2015
Prof.: Douglas G. Macharet
Monitor: André H. C. Silva

Trabalho Prático 1 – Rede Social de Pesquisadores

Valor: 10 pontos
Data de devolução: 15/04/2015

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos de Orientação a Objetos e com a programação utilizando a linguagem Java. Deverá ser implementada uma estrutura de classes que permita armazenar e manipular os dados de uma rede social (fictícia) de pesquisadores.

Instruções:

Considerando os arquivos de entrada apresentados nas próximas seções, o aluno deve criar uma hierarquia de classes, com os atributos e métodos necessários, tendo como objetivo a extração de um conjunto de informações a partir de uma rede social de pesquisadores.

Deverão ser implementados métodos específicos para se calcular:

- a popularidade de cada pesquisador;
- o fator de impacto de cada veículo de publicação;
- e a qualidade de cada artigo (publicação).

Todas as informações deverão ser cadastradas no sistema a partir de um grupo de arquivos de entrada que serão disponibilizados na mesma pasta de execução do programa. Após a leitura dos arquivos de entrada, o sistema deverá gerar arquivos de saída para os resultados de cada um dos métodos, ordenados pelo *id*, no padrão:

- “popularidade_pesquisador.txt” <idPesquisador;popularidade>
- “fatorImpacto_veiculo.txt” <idVeiculo;fator_impacto>
- “pontuacao_artigo.txt” <idArtigo;pontuacao>

Todos os possíveis casos de exceção deverão ser tratados na implementação e relatados na documentação.

Para guardar as informações relativas às coleções, você pode utilizar qualquer uma das classes que implementa a interface `Collection<E>`. Você pode obter mais informações em:
http://docs.oracle.com/javase/7/docs/api/java/util/Collection.html.

Informações:

1. Um *pesquisador* pode ser *graduado*, *mestre* ou *doutor*. Ele deve ser identificado por um número único (id). Um pesquisador possui um conjunto de artigos (publicações) associados a ele. Todo pesquisador possui um peso de importância *w*. Este peso (*w*) é calculado baseado na ordem de importância do pesquisador nas suas publicações. Se ele sempre for primeiro autor, ele sempre terá peso elevado.

A Equação para efetuar o peso da importância é apresentada logo abaixo:

$$w(p) = \sum_{k=1}^{n} \frac{1}{o(p,k)}$$

onde *o(p,k)* é a posição (relativa à autoria) em que o pesquisador *p* aparece no artigo *k* e *n* é o número de artigos publicados pelo pesquisador *p*.

Por exemplo: considere que o pesquisador publicou 6 artigos no total (*n = 6*). Se ele publicou 3 artigos como **segundo** autor, 2 como **primeiro** e 1 como **terceiro**, seu peso (w) será:

$$w = (1/2) + (1/2) + (1/2) + (1/1) + (1/1) + (1/3) = 3.8333$$

Para calcular a popularidade de um *pesquisador*, de forma geral, deve-se basear na seguinte equação:

$$pesquisador{:}popularidade(p) = \mathbf{w(p) + c(p) + a(p)}$$

onde *w(p)* é o peso de importância do pesquisador *p*, *c(p)* é o número de vezes que os artigos do pesquisador *p* foram citados e *a(p)* é o número de artigos publicados pelo pesquisador *p*.

Se o pesquisador for *graduado* o cálculo de popularidade deve ser feito da seguinte maneira:

$$popularidade(p) = w(p) + c(p) + a(p) + \mathbf{hIC(p) + hED(p)}$$

onde *hIC(p)* é quantidade de horas que o pesquisador *p* dedicou à iniciação científica e *hED(p)* é a quantidade de horas que o pesquisador *p* dedicou ao estágio docência.

Caso o pesquisador seja um *mestre*, a popularidade deve ser calculada da seguinte forma:

$$popularidade(p) = w(p) + c(p) + a(p) + hIC(p) + hED(p) + 10*\mathbf{g(p)}$$

onde *g(p)* é quantidade de alunos de graduação orientados pelo pesquisador *p*.

Se o pesquisador for *doutor*, o cálculo de popularidade deve ser realizado conforme a equação abaixo:

$$popularidade(p) = w(p) + c(p) + a(p) + hIC(p) + hED(p) + 10*g(p) + \mathbf{20*m(p) + 30*d(p)}$$

onde *m(p)* e *d(p)* é a quantidade de alunos de mestrado e doutorado, respectivamente, orientados pelo pesquisador *p*.

2. Neste trabalho são considerados dois tipos de *veículo de publicação*: *revistas (R)* e *conferências (C).* Cada *veículo de publicação* também deve possuir um identificador único *(*id*).* Para calcular o *fator de impacto* de um veículo *v*, de maneira geral, deve-se basear na seguinte equação:

$$fat(v) = \boldsymbol{c(v) / n(v)}$$

onde *c(v)* é o número de vezes que os artigos presentes no veículo *v* foram citados e *n(v)* é o número de artigos publicados nesse veículo.

Se o veículo for uma *conferência* (C), o cálculo do *fator de impacto* deve ser feito da seguinte forma:

$$fat(v) = [c(v) / n(v)] + \boldsymbol{1}$$

Caso contrário, se o veículo for uma *revista* (R), o *fator de impacto* deve ser calculado da seguinte maneira:

$$fat(v) = [c(v) / n(v)] + \boldsymbol{2}$$

3. Os artigos também devem ser representados. Cada artigo possui um identificador único (*id*). Todo artigo *pertence* a um *veículo* de publicação. Para calcular a qualidade de um artigo, deve ser utilizado o *fator de impacto* do veículo ao qual pertence este artigo, bem como a quantidade de citação dele, conforme apresentado abaixo:

$$qualidade(k) = fat(v,k) * c(k)$$

onde *fat(v,k)* é o fator de impacto do veículo *v* ao qual o artigo *k* pertence e *c(k)* é o número de vezes que o artigo *k* foi citado.

Funcionalidades:

1. Criar objetos representando cada um dos Pesquisadores, levando em conta a titulação: Graduado (G), Mestre (M) ou Doutor (D)
2. Criar objetos representando veículos de publicação: revistas (R) ou conferências (C)
3. Adicionar um conjunto de pesquisadores
4. Adicionar um conjunto de veículos de publicação
5. Adicionar um conjunto de artigos
6. Extrair a popularidade de cada pesquisador e salvar em um arquivo
7. Extrair o fator de impacto de cada veículo de publicação e salvar em um arquivo
8. Extrair a qualidade de cada artigo (publicação) e salvar em um arquivo

Arquivos de Entrada e Saída:

Exemplo dos arquivos de entrada:

- **Pesquisadores: id_pesquisador;tag_titulacao;horas_ic;horas_estagio_docencia;num_grad_orient;num_M_orient;num_D_orient**

pesquisadores.txt

```
1;G;150;190;0;0;0
2;G;127;271;0;0;0
3;G;218;299;0;0;0
4;G;201;209;0;0;0
5;M;129;99;13;0;0
```

- **Grafo de Pesquisadores: id_pesquisadorA;id_pesquisadorB**
  - Este é um grafo não-direcionado que indica a relação entre os pesquisadores. Por exemplo: os pesquisadores 1 e 2 possuem pelo menos uma publicação em comum.

grafo_pesquisadores.txt

```
1;2
1;4
1;5
2;3
2;5
3;5
4;5
```

- **Grafo Bipartido (artigos X pesquisadores): id_artigo;id_pesquisador;ordem_autoria**
  - Este é um grafo que apresenta a relação entre artigos e seus respectivos autores; bem como a *ordem de autoria* dos autores.
  - Por exemplo: o **artigo 25** tem como **primeiro** autor o **pesquisador 4** e como **segundo** autor o **pesquisador 5**.

grafo_artigos_pesquisadores.txt

```
1;5;1
2;3;1
3;5;1
4;5;1
…
23;5;1
24;5;1
25;4;1
25;5;2
26;1;1
```

- **Grafo de Citações: id_artigoA;id_artigoB**
  - Este grafo indica quais artigos são citados e quem os cita.
  - Exemplos:
    - o **artigo 25** é citado pelo **artigo 13** e pelo **artigo 21**
    - o **artigo 21** é citado pelo **artigo 3**
    - o **artigo 19** não é citado por nenhum artigo

grafo_citacoes.txt

```
3;19
6;10
11;17
13;5
13;23
21;3
25;13
25;21
```

- **Veiculos: id_veiculo;tag_veiculo**
  - indica qual o tipo de veículo: revista (R) ou conferência (C).
  - Por exemplo: o **veículo 1** é uma **revista** e os **veículos 2 e 3** são **conferências**.

veiculos.txt

```
1;R
2;C
3;C
```

- **Mapeamento (artigos X veiculos): id_artigo;id_veiculo**
  - indica a qual veículo pertence um determinado artigo
  - Exemplos:
    - o **artigo 2** pertence ao **veículo 3** (conferência)
    - o **artigo 5** pertence ao **veículo 1** (revista)

artigos_veiculos.txt

```
1;1
2;3
3;2
4;2
5;1
6;2
7;3
8;2
…
20;3
21;3
22;3
23;1
24;1
25;1
26;2
```

Exemplo dos arquivos de saída:

Os arquivos de saídas serão relativos às características extraídas da rede social de pesquisadores. Deverão ser produzidos 3 arquivos de saída. O arquivo "popularidade_pesquisador.txt" deverá ter a popularidade de cada pesquisador. O arquivo "fatorImpacto_veiculo" deverá ter o fator de impacto de cada veículo de publicação (revista ou conferência). E, por fim, o arquivo "pontuacao_artigo.txt" deverá apresentar uma métrica de qualidade (pontuacao) para cada artigo. Todos estes arquivos devem ordenados, crescentemente, de acordo com *id*. A seguir são mostrados exemplos dos 3 arquivos de saída, considerando os arquivos de entrada apresentados anteriormente.

Formato: `idPesquisador;popularidade`

popularidade_pesquisador.txt

```
1;344.5000
2;401.0000
3;532.5000
4;418.0000
5;502.0000
```

Formato: `idVeiculo;fator_impacto`

fatorImpacto_veiculo.txt

```
1;2.3333
2;1.4444
3;1.1250
```

Formato: `idArtigo;pontuacao`

pontuacao_artigo.txt

```
1;0.0000
2;0.0000
3;1.4444
4;0.0000
5;0.0000
6;1.4444
7;0.0000
8;0.0000
9;0.0000
10;0.0000
11;2.3333
12;0.0000
13;2.8889
14;0.0000
15;0.0000
```

```
16;0.0000
17;0.0000
18;0.0000
19;0.0000
20;0.0000
21;1.1250
22;0.0000
23;0.0000
24;0.0000
25;4.6667
26;0.0000
```

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.

O que deve ser entregue:

Envie um arquivo ZIP com o nome no formato 'tp1-nome1-nome2.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', 'make run' e 'make clean'.
- Ao executar o 'make run' devem ser produzidos os três (3) arquivos de saída solicitados: *"popularidade_pesquisador.txt"*, *"fatorImpacto_veiculo.txt"* e *"pontuacao_artigo.txt"*.
- A documentação do trabalho bem escrita e detalhada.

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.
- Serão fornecidos alguns casos de teste, porém na correção outros casos de testes serão utilizados.
- 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.

Critérios de avaliação:

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