Universidade Federal do Espírito Santo
Centro Tecnológico
Departamento de Informática

Prof. João Paulo A. Almeida (adaptado do Prof. Vítor E. Silva Souza)

Especificação do Trabalho Prático (1)

O trabalho prático da disciplina consiste em desenvolver um sistema computacional para solução do problema descrito abaixo na linguagem de programação Java.

1. Descrição do problema

Uma revista de informática, a *EngeSoft*, deseja um novo sistema para gerenciar suas atividades. *EngeSoft* é publicada mensalmente, sendo que uma edição tem diversos artigos, todos versando sobre um mesmo tema. Por exemplo, a edição deste mês é sobre o tema "Qualidade de Software", tendo oito artigos. De uma edição deseja-se saber o volume, número, data de publicação, tema e artigos submetidos e selecionados.

Autores submetem artigos para uma edição específica. De um artigo deseja-se saber os autores e o título. Os autores devem informar, além de seus nomes, e-mails e as instituições a que pertencem com endereço. Para artigos com mais de um autor, deve ser indicado um autor como contato.

Para avaliar os artigos submetidos à publicação, a *EngeSoft* possui um conjunto de colaboradores que avaliam artigos (chamados revisores). Dos revisores deseja-se saber o nome, e-mail, instituição e temas para os quais está habilitado a avaliar artigos.

Essas informações são usadas para distribuir os artigos para os colaboradores. Cada artigo é obrigatoriamente avaliado por três revisores, todos habilitados ao tema da edição correspondente, que atribuem notas de 0 a 10 (com até uma casa decimal) a três itens: originalidade, conteúdo e apresentação. Com base nessas avaliações é que se decide se um artigo será publicado ou não. Para simplificar, considere que um revisor é obrigado a avaliar um artigo quando este lhe é atribuído (ou seja, não pode haver cancelamento de atribuição).

Artigos que já foram avaliados pelos seus três revisores estão prontos para a seleção de quais artigos serão publicados na edição em questão, caso contrário encontram-se ainda em avaliação. Apenas quando todos os artigos submetidos para uma edição tiverem sido avaliados é que a seleção pode ser efetuada. Esta seleção é feita pelo editor-chefe da edição, escolhido previamente no conjunto de colaboradores. Finda a seleção, sabem-se quais artigos foram selecionados para publicação e quais foram rejeitados.

A diretoria da revista *EngeSoft* gostaria de um sistema para auxiliar a tarefa do editor-chefe de cada edição. O sistema deverá, dadas as avaliações dos artigos, calcular as médias das avaliações e apresentar ao editor-chefe um relatório dos artigos avaliados, por ordem de média das avaliações. São pedidos, ainda, alguns relatórios complementares, descritos nas próximas seções.

O projeto de construção de um sistema para a revista *EngeSoft* se dará em duas etapas: na primeira etapa, os dados serão mantidos em planilhas eletrônicas e, quando todos os dados estiverem prontos, eles serão passados ao programa, que produzirá relatórios de acordo com os dados fornecidos (vide seção 2).

2. Especificação

Para a primeira etapa, deverá ser construído um software que efetua a leitura dos arquivos de entrada e produz automaticamente os relatórios de saída. Para uma transição mais lenta entre o sistema atual (manual) e um sistema completamente informatizado, foi combinado que os cadastros seriam feitos em planilhas eletrônicas ao longo do mês e, assim que estivessem completos (todas as informações da edição), eles seriam passados ao software para geração dos relatórios.

Para o processamento destes dados e geração dos relatórios desejados, um funcionários da *EngeSoft* irá exportar os dados das planilhas para arquivos de texto simples com valores separados por vírgulas, conhecido como CSV (*Comma Separated Values*). No entanto, para evitar conflito com representação de valores decimais (ex.: 3,9), os dados serão exportados utilizando ponto-e-vírgula como separador (ex.: 123;456;9,5;7,8;8,0 – representando que o revisor 123 avaliou o artigo 456 e atribuiu-lhe as notas 9,5; 7,8; e 8,0).

Para facilitar a leitura dos relatórios produzidos pelo programa, será feita a importação dos dados dos relatórios do formato CSV para planilha eletrônica. Portanto, seu programa deve ser capaz de ler dados neste formato e gerar os relatórios também no mesmo formato.

O restante desta seção especifica em detalhes como o software deve funcionar na primeira etapa. A especificação encontra-se dividida em quatro partes: leitura dos dados, processamento, escrita dos relatórios e tratamento de exceções.

2.1. Leitura dos dados

São cinco os arquivos de entrada de dados:

- Informações da edição;
- Cadastro de temas;
- Cadastro de pessoas;
- Cadastro de artigos;
- Cadastro de revisões.

Os nomes dos arquivos são especificados durante a execução do programa (vide Seção 3). Abaixo encontra-se especificada a ordem que os dados devem aparecer em cada um destes arquivos e os tipos de dados esperados:

Informações da edição

```
<Nome do tema>
<Nome do editor-chefe>
<Volume>
<Número>
<Data de publicação>
```

Ao contrário dos cadastros, o arquivo de informações da edição não encontra-se em formato CSV. Ele contém informações de uma única edição (a que está sendo produzida) e cada informação encontra-se em uma linha diferente. Volume e número são numéricos

(inteiros), data de publicação é uma data no formato dia/mês/ano-com-4-dígitos e os demais campos devem ser lidos como texto.

Cadastro de temas

```
<código>;<nome>;<códigos dos revisores capacitados para o tema>
```

Tanto o código do tema como dos revisores é numérico (inteiro). Os códigos dos revisores são separados por vírgula. Nome deve ser lido como texto.

Cadastro de pessoas

```
<código>;<nome>;<email>;<senha>;<instituição>;<endereço>;<tipo>
```

Código é numérico (inteiro), tipo deve ser A (para autor) ou R (para revisor) e os demais campos devem ser lidos como texto.

Cadastro de artigos

```
<código>;<título>;<códigos dos autores>;<código do autor contato>
```

Todos os códigos são numéricos (inteiros) e os códigos dos autores são separados por vírgula. O código do autor contato só é obrigatório quando o artigo tem mais de um autor. O título deve ser lido como texto.

Cadastro de revisões

```
<código do artigo>;<código do revisor>;<nota originalidade>;<nota
conteúdo>;<nota apresentação>
```

Todos os dados são numéricos, sendo os códigos inteiros e as notas números reais com até 1 casa decimal.

2.2. Processamento

Lidos todos os dados, o programa deve criar objetos em memória representando as informações contidas nos arquivos de entrada. Tais objetos devem estar ligados adequadamente, conforme as associações entre as classes de objetos.

Antes de proceder para os cálculos e escrita dos relatórios, é preciso efetuar uma verificação de consistência dos dados. A tabela abaixo lista os possíveis problemas de consistência que podem ocorrer e o que deve ser incluído no relatório de resumo (vide próxima subseção) no caso dos dados não estarem consistentes.

#	Possível inconsistência	A incluir no resumo
1	O nome do tema em *Informações da edição* não existe no *Cadastro de temas*	O tema "<nome do tema>" não foi encontrado no cadastro.
2	O nome do editor-chefe em *Informações da edição* não existe no *Cadastro de pessoas* como um revisor.	O editor-chefe "<nome do editor-chefe>" não foi encontrado no cadastro.

#	Possível inconsistência	A incluir no resumo
3	Revisores referenciados no *Cadastro de temas* não correspondem a revisores no *Cadastro de pessoas*.	O código <código do revisor> associado ao tema "<nome do tema>" não corresponde a um revisor cadastrado.
4	Não há ao menos 3 revisores no *Cadastro de temas* para o tema da edição em questão.	O tema "<nome do tema>" possui apenas <X> revisores. São necessários no mínimo 3 revisores.
5	Uma das pessoas no *Cadastro de pessoas* possui tipo diferente de A ou R.	O tipo de "<nome da pessoa>" não é um tipo válido: <tipo>.
6	Autores referenciados no *Cadastro de artigos* não correspondem a autores no *Cadastro de pessoas*.	O código <código do autor> associado ao artigo "<título do artigo>" não corresponde a um autor cadastrado.
7	O autor de contato no *Cadastro de artigos* não é um dos autores do artigo em questão.	O código <código do autor de contato> informado como autor de contato não corresponde a um dos autores do artigo "<título do artigo>".
8	Revisores referenciados no *Cadastro de revisões* não correspondem a revisores no *Cadastro de pessoas*.	O código <código do revisor> encontrado no cadastro de revisões não corresponde a um revisor cadastrado.
9	Artigos referenciados no *Cadastro de revisões* não se encontram no *Cadastro de artigos*.	O código <código do artigo> encontrado no cadastro de revisões não corresponde a um artigo cadastrado.
10	Revisores referenciados no *Cadastro de revisões* não estão aptos a revisar artigos do tema da edição.	O revisor "<nome do revisor>" avaliou o artigo "<nome do artigo>", porém ele não consta como apto a revisar o tema "<nome do tema>", desta edição.
11	Não há exatamente três revisões para cada artigo no *Cadastro de revisões*.	O artigo "<título do artigo>" possui <X> revisões. Cada artigo deve ter exatamente 3 revisões.

Alguns problemas de consistência podem ocorrer mais de uma vez (ex.: se no cadastro de revisões houver mais de um artigo com um número de revisões diferente de 3, o problema #11 ocorrerá uma vez para cada artigo). Nesse caso devem ser incluídas no relatório múltiplas mensagens, uma para cada ocorrência do problema.

Caso o programa passe a verificação de consistência sem erros, ele deverá calcular as estatísticas e as médias das revisões dos artigos de modo a possibilitar a escrita dos dados nos relatórios. Os dados exatos a serem calculados podem ser vistos na próxima seção, que descreve em detalhes o que deve ser escrito em cada relatório.

2.3. Escrita dos relatórios

São três os arquivos de saída de dados:

- Resumo;
- Relatório de revisões;
- Relatório de revisores.

Os nomes dos arquivos são especificados durante a execução do programa (vide Seção 3). Abaixo encontra-se especificada a ordem que os dados devem aparecer em cada um destes arquivos e os formatos que devem ser utilizados:

Resumo

```
EngeSoft, num. <número>, volume <volume> - <data>
Tema: <tema>
Editor-chefe: <editor-chefe>

Consistência dos dados:
<consistência>

Artigos submetidos: <número de artigos submetidos>
Revisores capacitados: <número de revisores capacitados ao tema>
Revisores envolvidos: <número de revisores que fizeram revisão>
Média artigos/revisor: <média artigos/revisor>
```

Ao contrário dos relatórios, o resumo não deve ser escrito em formato CSV. Ele deve ser escrito exatamente como acima, substituindo os termos entre < e > pelos valores apropriados:

- Número de volume da edição devem ser impressos normalmente como números, sem formatação especial;
- A data deve ser impressa no formato "<mês-por-extenso> de <ano-com-4-dígitos>". Por exemplo: Setembro de 2014;
- Tema e editor-chefe devem ser impressos normalmente como texto, sem formatação especial;
- O campo <consistência> deve ser substituído pelas mensagens de erro geradas durante a validação da consistência dos dados. As mensagens devem ser prefixadas com "- Erro <#>: <mensagem>", onde <#> deve ser substituído pelo número do erro (vide 1ª coluna da tabela de possíveis inconsistências, apresentada anteriormente) e <mensagem> deve ser substituída pela mensagem de erro. Por exemplo:

```
Consistência dos dados:
- Erro 8: O código <código do revisor> encontrado no cadastro de
revisões não corresponde a um revisor cadastrado.
- Erro 11: O artigo "Exemplo de Artigo" possui 2 revisões. Cada
artigo deve ter exatamente 3 revisões.
- Erro 11: O artigo "Outro Exemplo de Artigo" possui 4 revisões.
Cada artigo deve ter exatamente 3 revisões.
```

- Caso não haja nenhum problema de consistência, o campo <consistência> deve ser substituído por "- Nenhum problema encontrado.";
- Por fim, os números inteiros nas estatísticas devem ser impressos normalmente, sem formatação, enquanto a média de artigos por revisor deve ser impressa com 2 casas decimais.

Relatório de revisões

```
<título do artigo>;<autor de contato>;<média>;<nome do 1º revisor>;<nome do 2º revisor>;<nome do 3º revisor>
```

O relatório de revisões deve ser escrito em CSV e conter as informações de um artigo por linha. O título do artigo, o autor de contato e os nomes dos revisores devem ser escritos como texto, sem formatação especial. A média deve ser formatada para ter apenas 2 casas decimais e usar separação com vírgula (brasileira). Os revisores de uma mesma linha do arquivo devem aparecer em ordem alfabética. O arquivo como um todo deve ser ordenado em ordem decrescente de média. Em caso de empate, ordene alfabeticamente pelo título do artigo.

Relatório de revisores

```
<nome do revisor>;<número de artigos revisados>;<média das notas atribuídas>
```

O relatório de revisores deve conter apenas os revisores que participaram da edição, ou seja, que revisaram artigos. Eles devem ser incluídos no relatório em ordem alfabética dos nomes, que devem ser escritos normalmente, sem formatação. O número de artigos revisados é um número inteiro e também não necessita de formatação. A média das notas atribuídas deve ser formatada com 2 casas decimais e separação com vírgula.

2.4. Tratamento de exceções

Leitura de dados de arquivos, formatação, etc. são fontes comuns de erros e exceções. Seu programa deve tratar **apenas** os seguintes tipos de erro:

1. Erros de entrada e saída de dados como, por exemplo, o arquivo especificado não existir ou o programa não ter permissão para ler ou escrever em um arquivo. Nestes casos, o programa deve exibir a mensagem "Erro de I/O";
2. Erro de formatação dos dados nos arquivos, ou seja, um valor formatado de forma incorreta nos arquivos de entrada (ex.: encontrado caractere onde esperava-se um número), causando erros de *parsing* dos dados. Nestes casos, o programa deve exibir a mensagem "Erro de formatação".

No programa sem interatividade, as mensagens devem ser impressas na tela e o programa deve ser terminado em seguida. No caso do programa com interface gráfica, as mensagens devem ser exibidas de alguma forma na interface com o usuário e o programa deve continuar rodando, permitindo que o usuário modifique os parâmetros.

Quaisquer outras situações de erro possíveis devem ser ignoradas. Pode-se assumir que nos testes feitos durante a avaliação dos trabalhos outros tipos de erros diferentes dos listados acima nunca acontecerão.

3. Execução

Esta seção descreve como o programa deve ser executado.

Seu programa deve ser executado especificando os nomes dos arquivos de entrada como opções de linha de comando, especificadas a seguir (em qualquer ordem):

- -e <arquivo>: informações da edição;
- -t <arquivo>: cadastro de temas;
- -p <arquivo>: cadastro de pessoas;
- -a <arquivo>: cadastro de artigos;
- -r <arquivo>: cadastro de revisões.

Apesar do uso do pacote *default* não ser recomendado, para efeito dos exemplos abaixo vamos supor que a classe do seu programa que possui o método `main()` chama-se `Main` e encontra-se no pacote *default*. Portanto, para executar seu programa lendo os arquivos edicao.txt, temas.csv, pessoas.csv, artigos.csv, revisoes.csv como arquivos de entrada, o comando seria:

```
java Main -e edicao.csv -t temas.csv -p pessoas.csv -a
artigos.csv -r revisoes.csv
```

4. Condições de entrega

O trabalho deve ser feito obrigatoriamente em dupla impreterivelmente até o dia 14/11/2014.

Dado que existem várias versões dos compiladores Java, fica determinado o uso das versões instaladas nas máquinas do Labgrad como versões de referência para o trabalho prático. Seu trabalho deve compilar e executar corretamente nas máquinas do Labgrad. Além disso, os arquivos de código-fonte devem estar em arquivos codificados com Unicode (UTF-8) para evitar erros de compilação.

4.1. Entrega do trabalho

Sua solução deverá ser compactada e enviados por e-mail (anexo ao e-mail) para o professor (jpalmeida@inf.ufes.br). Serão aceitos (sem penalidade) trabalhos entregues até as 23h59 da data limite. O assunto do e-mail deverá ser o seguinte:

`PROG3 - Trab1 - *Nomes dos alunos*`

substituindo *Nomes dos alunos* pelos nomes completos dos alunos do grupo, separado por vírgula.

Dada a quantidade de trabalhos que devem ser avaliados, a correção dos trabalhos passará primeiro por um processo de testes automáticos e, em seguida, por uma avaliação subjetiva em entrevista (mais detalhes na seção 4.3).

Para que os testes automáticos funcionem, o arquivo compactado enviado por e-mail deve estar no formato `zip` com o nome `trabalho.zip` e conter o arquivo de *build* (explicações a seguir) e o código-fonte. O arquivo enviado não deve conter nenhuma classe compilada. Os testes automáticos serão executados no diretório onde encontra-se o arquivo de *build*. O código-fonte pode ser organizado da forma que a dupla achar melhor, desde que o arquivo de *build* esteja adequado a esta estrutura.

4.2. Preparação e execução do script de testes

Será disponibilizado oportunamente aos alunos um script para execução de alguns testes automáticos, sendo portanto possível garantir que o trabalho passa nesses testes antes de submetê-lo ao professor. Apesar disto, **os alunos deverão fazer testes adicionais e não apenas aqueles disponibilizados pelo professor**.

5. Critérios de avaliação

Para a avaliação objetiva, todo trabalho possui inicialmente nota 10 e sofre modificações nas situações descritas na tabela abaixo:

Situação	Modificação
Não observou as regras para envio do trabalho 1.	-1
Não compilou nos testes automáticos, mas foi possível corrigir manualmente (ex.: arquivos não codificados em UTF-8).	-3
Não compilou nem manualmente.	-10
Não gerou saídas nos testes automáticos.	-5
Não gerou saídas nem manualmente.	-8
Pequenas diferenças das saídas em relação ao teste automático (ex.: formatação, arredondamentos, etc.).	-1
Grandes diferenças das saídas em relação ao teste automático (ex.: valores sensivelmente diferentes).	-2
Entrega fora do prazo.	-1 por dia

Para avaliação subjetiva, será utilizada a nota da parte objetivo como base e descontados pontos (que variam de acordo com a avaliação feita pelo professor) caso não estejam bem escritos ou organizados. Critérios utilizados na avaliação subjetiva incluem (mas não estão limitados a):

- Uso dos princípios básicos da orientação a objetos, como encapsulamento, abstração e modularização;

- Legibilidade (nomes de variáveis bem escolhidos, código bem formatado, uso de comentários quando necessário, etc.);
- Consistência (utilização de um mesmo padrão de código, sugere-se a convenção de código do Java: http://www.oracle.com/technetwork/java/codeconv-138413.html);
- Eficiência (sem exageros, tentar evitar grandes desperdícios de recursos);
- Uso eficaz da API Java (leitura com Scanner, API de coleções, etc.) e das funcionalidades das novas versões da plataforma (ex.: tipos genéricos, laço *for-each*, *try* com recursos fecháveis, etc.);

6. Observações finais

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, frequentando as aulas ou acompanhando as novidades na página da disciplina na Internet.