

Trabalho Prático

Objetivo

O objetivo desse trabalho é consolidar e praticar o conteúdo visto na disciplina no decorrer do semestre letivo. Para isso os alunos devem se dividir em grupos de 3 pessoas para desenvolver o trabalho proposto nesse documento (**Todos os grupos devem ter exatamente 3 integrantes**).

Descrição do Problema

Considere que seu grupo foi contratado para desenvolver um programa para gerenciar o processo seletivo da Universidade Stark. Nesta universidade, os cursos possuem números restritos de vagas, mas possuem infinitas posições na fila de espera. Para seleção dos candidatos deve ser utilizada a média simples das notas da Redação, prova de Matemática e prova de Linguagens (somar as três notas e dividir por três). Caso ocorra empate de notas entre alunos, o critério de desempate será a nota da redação. (Para simplificar o trabalho não teremos outros critérios de desempate. Assim vamos considerar que dois alunos com a mesma média sempre terão as notas da Redação diferentes). Na Universidade Stark, cada candidato deve selecionar duas opções de curso. O candidato selecionado em sua primeira opção não deverá ser colocado em nenhuma fila de espera. O candidato selecionado para sua segunda opção deverá ser colocado na fila de espera da primeira opção. Um candidato que não for selecionado para nenhuma de suas opções deverá ser colocado na fila de espera de ambos os cursos (opções 1 e 2). Se um candidato for selecionado para suas duas opções de curso, ele deve ser incluído apenas na lista de candidatos selecionados da sua primeira opção de curso (isto é, deve liberar a vaga da segunda opção).

Para simplificar o trabalho considere que a Universidade Stark aceitará, no máximo, 1000 candidatos no processo seletivo.

O programa deverá ler informações de um arquivo de entrada e ao final do processamento deverá criar um arquivo texto contendo as seguintes informações:

- Nome e nota de corte de cada curso (menor nota média dos selecionados para o curso).
- Lista de candidatos selecionados (nome e nota média de cada candidato), em ordem decrescente de nota média.
- Fila de espera (nome e nota média de cada candidato), em ordem decrescente de nota média.

Arquivo de Entrada (entrada.txt)

A primeira linha do arquivo de entrada conterá dois inteiros N e M:

N : número de cursos

M: número de candidatos

As N linhas seguintes terão as seguintes informações sobre os N cursos, separadas por ponto e vírgula: código do curso (inteiro), nome do curso (String) e quantidade de vagas disponíveis no curso (int).

Após as informações dos cursos, as próximas M linhas terão as informações dos M candidatos. Cada linha terá as seguintes informações separadas por ponto e vírgula: nome do candidato (String), nota obtida pelo candidato na redação (double), nota obtida pelo candidato na prova de matemática (double), nota obtida pelo candidato na prova de linguagens (double), código da primeira opção de curso (int) e o código da segunda opção de curso.

Arquivo de Saída (saida.txt)

Para cada curso, na mesma ordem de entrada, deverá ser escrito em uma mesma linha o nome do curso e a nota de corte (com duas casas decimais), separados por um espaço.

Na próxima linha deverá ser escrita somente a String “*Selecionados*”. Em seguida deverão ser escritas S_i linhas contendo o nome do candidato e sua nota média, com duas casas decimais (S_i é o número de candidatos selecionados para o curso

i). Os candidatos deverão estar em ordem decrescente de nota média, seguindo o critério de desempate especificado anteriormente.

Por fim, deverá ser escrita a String “*Fila de Espera*”. Em seguida, E_i linhas contendo os nomes e as notas médias dos candidatos, com duas casas decimais, que estão na fila de espera do curso (E_i é o número de candidatos na fila de espera do curso i). Os candidatos deverão estar em ordem decrescente de nota média, seguindo o critério de desempate. Para separar as informações de cada curso, escreva uma linha em branco após o último nome da fila de espera.

As tabelas apresentam o modelo (esquerda) e um exemplo de entrada e saída (direita):

entrada.txt	entrada.txt
qtdCursos;qtdCandidatos codCurso1;nomeCurso;qtdVagas codCurso2;nomeCurso;qtdVagas ... codCursoN;nomeCurso;qtdVagas nomeCandidato1;notaRed;notaMat;notaLing;codCursoOp1;codCursoOp2 nomeCandidato2;notaRed;notaMat;notaLing;codCursoOp1;codCursoOp2 ... nomeCandidatoM;notaRed;notaMat;notaLing;codCursoOp1;codCursoOp2	4;8 1;Matemática;2 2;Física;2 3;Química;2 4;Estatística;5 Bob Esponja;600;700;800;1;2 Pato Donald;700;700;800;1;2 Mickey Mouse;800;700;800;1;2 Peppa Pig;700;700;500;3;2 Super Mario Bros;600;600;500;4;3 Peter Parker;700;850;800;3;4 Deadpool;900;700;800;3;4 Tio Patinhas;700;700;600;4;2

saida.txt	saida.txt
nomeCurso1 notaCorte Selecionados nomePrimeiroClassificado notaMedia nomeSegundoClassificado notaMedia ... nomeUltimoClassificado notaMedia Fila de Espera nomePrimeiroClassificado notaMedia nomeSegundoClassificado notaMedia ... nomeUltimoClassificado notaMedia nomeCurso2 notaCorte Selecionados nomePrimeiroClassificado notaMedia nomeSegundoClassificado notaMedia ... nomeUltimoClassificado notaMedia Fila de Espera nomePrimeiroClassificado notaMedia nomeSegundoClassificado notaMedia ... nomeUltimoClassificado notaMedia	Matemática 733.33 Selecionados Mickey Mouse 766.67 Pato Donald 733.33 Fila de Espera Bob Esponja 700.00 Física 633.33 Selecionados Bob Esponja 700.00 Peppa Pig 633.33 Fila de Espera Química 783.33 Selecionados Deadpool 800.00 Peter Parker 783.33 Fila de Espera Peppa Pig 633.33 Estatística 566.67 Selecionados Tio Patinhas 666.67 Super Mario Bros 566.67 Fila de Espera

Estruturas de Dados e Métodos

Um trabalho desse tamanho pode ser difícil de implementar sem uma devida organização por parte dos alunos. Sendo assim, o código deve ser obrigatoriamente segmentado em classes e métodos para fazer as diferentes funcionalidades do programa.

Além disso, o programa deve ter obrigatoriamente, no mínimo:

- **Lista simples flexível:** uma estrutura de dados Lista deve ser implementada e usada no processamento. Caso o grupo opte por usar várias listas, no mínimo uma deve ser implementada pelo grupo e as demais podem ser estruturas de dados Lista nativas do Java. Na estrutura de dados Lista podem ser implementados novos métodos, caso o grupo julgue necessário.
- **Fila flexível:** pode-se tanto usar a estrutura de dados fila do Java quanto implementar a própria estrutura de dados fila.
- **Método de ordenação:** o grupo deve implementar e usar no processamento algum método de ordenação. Escolha um método eficiente.

Entrega e Avaliação

Cada grupo deverá entregar o código Java e um relatório com as seguintes seções:

1. Desenvolvimento
2. Testes

No desenvolvimento os alunos devem explicar resumidamente cada parte da implementação do código, utilizando “printScreen” para mostrar tais partes. Justifique a escolha do método de ordenação.

Na parte de testes os alunos devem mostrar testes realizados com código desenvolvido. Utilize “printScreen” para mostrar o funcionamento do programa.

A avaliação dos trabalhos será feita em duas etapas. A primeira etapa será uma entrevista com cada grupo, onde perguntas serão feitas direcionadas a cada integrante do grupo. Como as perguntas não vão levar em consideração quem implementou o código, **TODOS** os integrantes do grupo devem entender o código por **COMPLETO!**

A segunda etapa será uma avaliação de funcionalidade do código e boas práticas de programação (código bem organizado, uso de comentários explicativos, variáveis com nomes intuitivos e escolha adequada das estruturas de dados). O funcionamento dos programas será avaliado por meio de um sistema automático, portanto respeitem os formatos dos arquivos de entrada e saída para evitar erros nos testes automáticos.

A nota será **INDIVIDUAL** apesar do trabalho ser em grupo. Alunos serão penalizados na entrevista de forma individual caso não consigam responder às perguntas. O grupo será penalizado por igual na fase de testes caso o programa não funcione corretamente. Trabalhos onde o plágio (cópia de colegas ou Internet) for identificado serão anulados.

Apenas um integrante de cada grupo deverá entregar no Canvas o código (arquivo compactado com todo o projeto) e o relatório (pdf).

Data de entrega (via Canvas): 26/06 até às 14h

Entrevistas com grupos: 26/06, 27/06 e 29/06

Referências

Adaptação do material elaborado pelos professores Raquel Prates e Luiz Chaimowicz