

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 e filas de espera **limitadas a 10 candidatos**. 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. Persistindo o empate, a nota de Matemática será utilizada como segundo critério de desempate, e, por fim, a nota de Linguagens como terceiro critério. Para simplificar, vamos considerar que não haverá casos de dois alunos com médias idênticas e notas iguais em todos os critérios de desempate. Na Universidade Stark, cada candidato deve selecionar duas opções de curso. Segue abaixo as regras de seleção:

1. Primeira Opção: O candidato selecionado para sua primeira opção de curso não deverá ser colocado em nenhuma fila de espera.
2. Segunda Opção: O candidato selecionado para sua segunda opção deverá ser colocado na fila de espera da primeira opção, desde que haja vaga disponível na fila de espera.
3. Nenhuma Opção: Um candidato que não for selecionado para nenhuma de suas opções deverá ser colocado na fila de espera de ambos os cursos, respeitando o limite de candidatos por fila de espera.
4. Ambas as Opções: 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).

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 (**A nota de corte de cada curso é a menor nota média dos selecionados para o curso**).
- Lista de candidatos selecionados (nome, nota média, notas da Redação, Matemática e Linguagens), em ordem decrescente de nota média, considerando os critérios de desempate.
- Fila de espera (nome, nota média, notas da Redação, Matemática e Linguagens), em ordem decrescente de nota média, considerando os critérios de desempate.

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, nota média com duas casas decimais, notas da Redação, Matemática e Linguagens (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, notas médias com duas casas decimais e notas da Redação, Matemática e Linguagens dos candidatos 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 Redação Matemática Linguagens nomeSegundoClassificado notaMedia Redação Matemática Linguagens ... nomeUltimoClassificado notaMedia Redação Matemática Linguagens Fila de Espera nomePrimeiroClassificado notaMedia Redação Matemática Linguagens nomeSegundoClassificado notaMedia Redação Matemática Linguagens ... nomeUltimoClassificado notaMedia Redação Matemática Linguagens nomeCurso2 notaCorte Selecionados nomePrimeiroClassificado notaMedia Redação Matemática Linguagens nomeSegundoClassificado notaMedia Redação Matemática Linguagens ... nomeUltimoClassificado notaMedia Redação Matemática Linguagens Fila de Espera nomePrimeiroClassificado notaMedia Redação Matemática Linguagens nomeSegundoClassificado notaMedia Redação Matemática Linguagens ... nomeUltimoClassificado notaMedia Redação Matemática Linguagens	Matemática 733,33 Selecionados Mickey Mouse 766,67 800 700 800 Pato Donald 733,33 700 700 800 Fila de Espera Bob Esponja 700,00 600 700 800 Física 633,33 Selecionados Bob Esponja 700,00 600 700 800 Peppa Pig 633,33 700 700 500 Fila de Espera Química 783,33 Selecionados Deadpool 800,00 900 700 800 Peter Parker 783,33 700 850 800 Fila de Espera Peppa Pig 633,33 700 700 500 Estatística 566,67 Selecionados Tio Patinhas 666,67 700 700 600 Super Mario Bros 566,67 600 600 500 Fila de Espera

Estruturas de Dados e Algoritmos

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:** usar a estrutura de dados List nativa do C# para armazenar *a lista de candidatos selecionados de cada curso*.
- **Fila:** usar a estrutura de dados Fila Linear para armazenar a *fila de espera* de cada curso. O grupo deverá implementar a estrutura de dados Fila Linear. Caso necessário, podem ser incluídos métodos na classe Fila Linear, além dos métodos vistos nas aulas (Obs: Se a Fila já estiver cheia, não deve ser lançada exceção. Apenas não deve ser feita a inserção).
- **Algoritmo de ordenação:** o grupo deve implementar e usar no processamento algum algoritmo de ordenação eficiente.
- **Dicionário:** usar a estrutura de dados Dicionário nativa do C# para armazenar os *cursos*.

Entrega e Avaliação

Apenas um integrante de cada grupo deverá entregar no Canvas uma pasta compactada com o código fonte (arquivos .cs).

O código não deve ter nenhum comentário.

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, 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.

Data de entrega (via Canvas): 17/06 até às 17h

Entrevistas com grupos: 17/06, 19/06 e 20/06

Referências

Adaptação do material elaborado pelos professores Raquel Prates e Luiz Chaimowicz