

Trabalho Prático 2 - Diário de Classe

Data de entrega: domingo, 28 Jul 2024, 23:59

Número máximo de arquivos: 1

Tipo de trabalho: Trabalho individual

Universidade Federal de Minas Gerais

Departamento de Ciência da Computação

Programação e Desenvolvimento de Software I

Trabalho Prático 2 - Diário de Classe

O Diário de Classe é uma ferramenta utilizada por professores de diversas instituições de ensino, como os da própria UFMG, para manter o controle sobre as informações dos alunos matriculados em suas disciplinas. Este trabalho prático consiste na implementação de um sistema baseado em funcionalidades que podem estar disponíveis em um diário de classe eletrônico, a ser realizada por meio de código escrito na linguagem C. Seu objetivo é treinar, por meio da aplicação práticas, os conceitos de criação de funções, manipulação de arquivos com extensão TXT e uso de registradores vistos em sala. **Leia este arquivo até o final para não perder informações importantes!**

Na nossa adaptação você deverá implementar as seguintes operações, que serão descritas de forma mais detalhada a seguir:

1. Exibir informações do professor
2. Exibir informações de um aluno
3. Inserir um aluno em uma turma
4. Lançar as notas de um aluno
5. Exibir informações de uma turma
6. Exibir a situação dos alunos de uma turma
7. Exportar os dados armazenados no sistema

Seu programa sempre será executado passando dois argumentos pela linha de comando. O primeiro será o arquivo TXT com as informações iniciais sobre o professor, suas turmas e os alunos já matriculados em cada turma. Já o segundo será também um arquivo TXT em que deverão ser escritos os dados do professor, turmas e alunos ao solicitar a opção de

exportação (7). Em outras palavras, o primeiro argumento é o arquivo de entrada e o segundo argumento é o arquivo de saída. Exemplo de execução:

```
./tp2 input.txt output.txt
```

O arquivo de entrada preencherá a maioria das informações do professor. Um professor possui um nome com até 100 caracteres, um número de registro que consiste em um inteiro $N \geq 1$ e **pelo menos** 1 turma, sendo que pode ter **no máximo** 2 turmas. Uma turma possui um código com até 10 caracteres, um nome com até 100 caracteres e uma lista de alunos, sendo que cada turma terá **pelo menos** 1 aluno e **no máximo 6 alunos**. Um aluno possui um nome com até 100 caracteres, um número de matrícula M que consiste em um inteiro $M \geq 1$ e as notas de suas avaliações, sendo que cada aluno realiza **exatamente** 3 avaliações. Segue o exemplo de um arquivo de entrada:

```
≡ input.txt
1  Albert Einstein
2  1801
3  2
4  ESTRUTURAS DE DADOS I
5  ED I
6  ALGORITMOS I
7  ALG I
8  4
9  Igor Vinicius
10 2401
11 Joaquim Silva
12 2402
13 Vitor Santos Silva Andrade
14 2403
15 Luciano Neves
16 2404
17 3
18 Andrezinho
19 2405
20 Matheus Pereira
21 2406
22 Joao Silva
23 2407
```

A primeira linha do arquivo de entrada representa o nome do professor e a segunda linha representa seu código de registro. A seguir, há um inteiro N ($1 \leq N \leq 2$) que representa a quantidade de turmas desse professor. As próximas $2 * N$ linhas representam, respectivamente, o nome da disciplina e o código da disciplina. Finalmente, para cada disciplina, haverá um inteiro M ($1 \leq M \leq 6$) que representa a quantidade de alunos matriculados na disciplina, sendo que **a ordem das disciplinas é a mesma ordem dos alunos matriculados**. No exemplo acima, os quatro primeiros alunos pertencem à disciplina de Estruturas de Dados I, enquanto os 3 últimos pertencem à disciplina Algoritmos I. Para cada disciplina, seguirão $2 * M$ linhas, em que a primeira linha representa o nome do aluno e a segunda o número de matrícula do aluno.

Após abrir o arquivo de entrada, ler e armazenar os dados, seu programa deve imprimir o seguinte menu:

- 1 - Informacoes do Professor
- 2 - Informacoes do Aluno
- 3 - Inserir Aluno
- 4 - Lancar Notas
- 5 - Informacoes da Turma
- 6 - Situacao dos Alunos
- 7 - Exportar Dados

Ele deve então ler e tratar a opção fornecida pelo usuário. Caso seja escolhida uma das operações entre a 1 e a 6, seu programa deve **aguardar uma nova instrução** após finalizar a operação atual. O programa encerra-se **somente** após o usuário escolher a operação 7 e ela terminar de executar.

Operação Escolhida	Entrada Fornecida	Saída Esperada
1	nenhuma.	SAÍDA PADRÃO (Terminal):

		<p>Professor xxxxxx</p> <p>Registro numero xxxxxx</p> <p>Para cada turma:</p> <p>Turma COD - NOME, x alunos (x é a quantidade de alunos)</p>
2	<p>ENTRADA PADRÃO (Terminal):</p> <p>matrícula do aluno desejado</p> <p>código da turma do aluno</p>	<p>SAÍDA PADRÃO:</p> <p>Aluno: xxxxxxxx</p> <p>Matricula: xxxxxxxx</p> <p>Prova 1: x1 / Prova 2: x2 / Prova 3: x3</p> <p>Nota Final: xg - Conceito X</p> <p>(ver cálculo de nota final e do conceito abaixo)</p>
3	<p>ENTRADA PADRÃO:</p> <p>nome do aluno</p> <p>matrícula do aluno</p> <p>código da turma em que ele entrará</p>	<p>nenhuma, o aluno deve ser inserido na turma correta com os dados informados.</p>
4	<p>ENTRADA PADRÃO:</p> <p>matrícula do aluno</p> <p>código da turma em que ele está</p> <p>nota da prova 1</p> <p>nota da prova 2</p> <p>nota da prova 3</p>	<p>nenhuma, o aluno deve ter as notas das respectivas avaliações registradas corretamente.</p>
5	ENTRADA PADRÃO:	SAÍDA PADRÃO:

	código da turma desejada	Informacoes da turma COD - NOME x alunos Para cada aluno na turma: Aluno: xxxxxxx Matricula: xxxxxxx
6	ENTRADA PADRÃO: código da turma desejada	SAÍDA PADRÃO: Situacao na Turma COD - NOME Para cada aluno: Aluno: xxxxxxxx Matricula: xxxxxxx Nota Final: x - Conceito X Situacao: xxxxx (ver as situações abaixo)
7	nenhuma.	Ver saída esperada abaixo. Deve ser escrita no arquivo de saída, definido pelo argumento de execução do programa.

Modelo de Saída Esperada para a Operação 7 (Exportar Dados)

<p>DADOS EXPORTADOS</p> <p>Professor xxxxxx - Registro xxxxxx</p> <p>Para cada turma</p> <p>Turma COD - NOME</p>
--

Para cada aluno na turma:

Aluno: xxxxxxxx

Matricula: xxxxxxxx

Nota Final: x - Conceito X - SITUACAO

(caso haja mais de uma turma, dê um espaço entre a nota final do último aluno da primeira turma e as informações da próxima turma)

Informações Importantes:

- seu programa deve, **obrigatoriamente**, implementar pelo menos 7 funções, uma para tratar cada operação escolhida. Você é livre para implementar quaisquer outras funções adicionais que julgar necessárias;
- a utilização de registradores (estruturas) no programa será levada em conta (positivamente) na avaliação dos trabalhos;
- todos os alunos, inclusive os inseridos manualmente, iniciam com 0 em todas as avaliações;
- a nota de cada avaliação será um valor **inteiro**, bem como a nota final deverá ser **um inteiro**;
- a nota final é dada pela **média** das avaliações. Cada avaliação receberá uma nota que N, tal que $0 \leq N \leq 100$;
- o conceito de um aluno é representado por um caractere e seguirá a disposição adotada pela UFMG:

CONCEITO	FAIXA DE NOTAS	SITUAÇÃO
A	90 a 100 pontos	Aprovado
B	80 a 89 pontos	Aprovado
C	70 a 79 pontos	Aprovado
D	60 a 69 pontos	Aprovado
E	40 a 59 pontos	Exame Especial
F	abaixo de 40 pontos	Reprovado

Exemplo de entradas e saídas esperadas para uma execução do programa. Considere como entrada a imagem do arquivo input.txt exibida mais acima.

./tp1 input.txt saida.txt

ENTRADA	SAÍDA PADRÃO	SAIDA.TXT
1	1 - Informacoes do Professor	DADOS EXPORTADOS
4	2 - Informacoes do Aluno	
2401	3 - Inserir Aluno	Professor Albert Einstein - Registro 1801
ED I	4 - Lancar Notas	
100 100 100	5 - Informacoes da Turma	Turma ED I - ESTRUTURAS DE DADOS I
2	6 - Situacao dos Alunos	
2401	7 - Exportar Dados	Aluno: Igor Vinicius
ED I	Professor Albert Einstein	Matricula: 2401
5	Registro numero 1801	Nota Final: 100 - Conceito A - Aprovado
ED I	Turma 1: ED I - ESTRUTURAS DE DADOS I, 4 alunos	Aluno: Joaquim Silva
3	Turma 2: ALG I - ALGORITMOS I, 3 alunos	Matricula: 2402
Novaldo	Aluno: Igor Vinicius	Nota Final: 21 - Conceito F - Reprovado
Alunaldo	Matricula: 2401	Aluno: Vitor Santos Silva Andrade
2409	Prova 1: 100 / Prova 2: 100 / Prova 3: 100	Matricula: 2403
ALG I	Nota Final: 100 - Conceito A	Nota Final: 60 - Conceito D - Aprovado
4	Informacoes da turma ED I - ESTRUTURAS DE DADOS I:	Aluno: Luciano Neves
2402	4 alunos	Matricula: 2404
ED I		

12 35 17	Aluno: Igor Vinicius	Nota Final: 74 - Conceito C - Aprovado
4	Matricula: 2401	
2403	Aluno: Joaquim Silva	
ED I	Matricula: 2402	Turma ALG I - ALGORITMOS I
60 55 65	Aluno: Vitor Santos Silva Andrade	Aluno: Andrezinho
4	Matricula: 2403	Matricula: 2405
2404	Aluno: Luciano Neves	Nota Final: 0 - Conceito F - Reprovado
ED I	Matricula: 2404	Aluno: Matheus Pereira
77 84 61	Informacoes da turma ALG I - ALGORITMOS I:	Matricula: 2406
7	4 alunos	Nota Final: 0 - Conceito F - Reprovado
	Aluno: Andrezinho	Aluno: Joao Silva
	Matricula: 2405	Matricula: 2407
	Aluno: Matheus Pereira	Nota Final: 0 - Conceito F - Reprovado
	Matricula: 2406	Aluno: Novaldo Alunaldo
	Aluno: Joao Silva	Matricula: 2409
	Matricula: 2407	Nota Final: 0 - Conceito F - Reprovado
	Aluno: Novaldo Alunaldo	
	Matricula: 2409	

Dica: verifique as funções strcmp(), strlen(), strcpy() para manipulação de strings e fflush() para lidar com entrada e saída. Você não irá necessariamente utilizar todas, mas podem ser úteis.

Avisos Importantes após o TP1:

- imprima na saída padrão ou no arquivo de saída **exatamente** o que está especificado no enunciado. Evite impressões do tipo "Digite sua escolha", visto que parte da correção é automática e ela pode identificar tais mensagens como sendo parte da sua resposta, resultando em avaliação incorreta;
- você **não precisa** tratar possíveis entradas incorretas, a não ser que esteja especificado no enunciado que isso deve ser feito. Se o enunciado diz que pode ser escolhida uma opção entre 1 e 7, não é necessário tratar o envio de valores abaixo de 1 ou acima de 7;
- nós apoiamos que os estudantes busquem deixar o programa mais interativo para o usuário e incentivamos que eles tentem adicionar mensagens informativas e tratamento de dados, mas **façam isso fora do ambiente de avaliação**, como um exercício extra, para que a avaliação não seja prejudicada.