

Trabalho Prático 1

Objetivos

Consiste em rever conceitos básicos de programação bem como explorar os conceitos de Ponteiros + Alocação de Memória + Tipos Abstratos de Dados (TADs).

Descrição

Você deverá implementar um tipo abstrato de dados **MatrizQuadrada** para representar matrizes quadradas. Seu tipo abstrato deverá armazenar os elementos da matriz e a sua dimensão (ou ordem) **n**. Considere que as suas matrizes tem uma dimensão determinada em tempo de execução (alocação dinâmica). Seu TAD deve possuir construtor (leitura de **n** e alocação dinâmica da matriz), destrutor e procedimentos (ou funções) para:

1. ler uma matriz (leitura dos dados) (**leMatriz**);
2. somar duas matrizes (**soma**);
3. produto de duas matrizes (**produto**);
4. criar uma matriz identidade (**criaMatrizIdentidade**);
5. criar a transposta de uma matriz (**criaTransposta**);
6. comparar duas matrizes (**compara**);
7. imprimir uma matriz (**imprimeMatriz**).

O seu programa deve estar modularizado (.h e .cpp) e além disso, deve conter um **main.cpp**, que utilize o seu TAD. O seu programa deverá ter uma estrutura similar (não necessariamente exata) a apresentada abaixo. Isso significa que você pode fazer ajustes que considerar pertinentes.

```
1  /* ... Declaracao e implementacao do TAD MatrizQuadrada ... */
2  void main(void) {
3      MatrizQuadrada M1,M2,M3,I,R,T;
4      leMatriz(M1); /* Le uma matriz nxn*/
5      criaMatrizIdentidade(I,3); /* Cria uma matriz identidade de tamanho n */
6      produto(M1,I,R); /* R = M1*I */
7      if (compara(A,R)){
8          cout << "Matrizes iguais!";
9      }
10     else{
11         cout << "Matrizes diferentes!";
12     }
13     imprimeMatriz(M1);
14     imprimeMatriz(R);
15     criaTransposta(R,T) /* T = transposta de R*/
16     leMatriz(M2); /* Le uma matriz nxn */
17     soma(R,M2,M3); /* M3 = R+M2 */
18     imprimeMatriz(M2);
19     imprimeMatriz(M3);
20 }
```

Desenvolvimento e Entrega

O código fonte do programa deve ser desenvolvido em C++, estar bem indentado e comentado. A entrega deve ser efetuada conforme agendado no PVANet Moodle. Para isso, você deve criar um projeto contendo os arquivos `.h`, `.cpp`, e `main.cpp` criados. Envie, através do PVANet Moodle, uma pasta compactada (.rar ou .zip) contendo o projeto. A pasta compactada deve conter informações do aluno (ex.: julio_reis-tp1.zip). Para correção, serão considerados os seguintes critérios:

1. Documentação (**1pt**).
 - (a) Detalhamento do código.
 - (b) Comentários, endentação.
2. Funcionamento correto (**2 pts**).
 - (a) Compila e executa, não apresenta *crash*, etc.
3. Aplicação correta dos conceitos (**2 pts**).
 - (a) Uso de ponteiros, gerenciamento de memória, etc.

Comentários Gerais

- Comece a fazer este trabalho logo: o prazo para terminá-lo está tão longe quanto jamais poderá estar! :)
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados serão penalizados (NOTA Zero).