

Desafio

Técnicas de Programação 2019.2

Programação Estruturada 2019.2

Resumo

O desafio tem como principal objetivo testar a sua capacidade de programar sob pressão. O desafio deve ser realizado em DUPLA, utilizando o conceito de [pair programming](#), ou seja, um computador por dupla e um revezamento a cada **20 minutos**. Você pode usar todos os recursos disponíveis e até trocar ideias com outras duplas mas seu código deve ser único, sem cópias.

O projeto deve ser entregue até o final da aula, pois vale alguns pontos extras para primeira prova.

- 2 pontos extras para as 3 primeiras duplas.
- 1,5 pontos extras para as 5 próximas duplas.
- 0,5 ponto extra para os demais que entregarem no tempo.

A funcionalidade e a organização do código são requisitos essenciais na avaliação do seu programa.

“Programs must be written for people to read, and only incidentally for machines to execute.”

— Harold Abelson, *Structure and Interpretation of Computer Programs*

Have fun!

Desafio

O desafio é criar um sistema completo para gerenciar suas tarefas diárias. Para fazer isso você terá que criar ao menos duas classes específicas. São elas:

1. A classe Tarefa para representar uma tarefa individual, que deve seguir o protótipo abaixo:

```
class Tarefa{  
    private:  
        string descricao;  
        string prioridade;  
        bool status;  
    public:  
        bool setDescricao(string d);  
        bool setPrioridade(string p);  
        void setStatus(bool b);  
  
        string getDescricao();  
        string getPrioridade();  
        string getStatus(); //Retorna "Finalizada" ou "Incompleta"  
  
        void obterTarefa();  
};
```

A sua implementação deve ter todos os métodos e atributos listados acima, mas fique a vontade para adicionar novas funcionalidades.

2. Uma classe Lista para gerenciar uma lista indefinida de tarefas, como o protótipo abaixo:

```
class Lista{  
    private:  
        vector<Tarefa> lista;  
    public:  
        void adicionarTarefa(Tarefa t);  
        void removerTarefa(string d); // remove UMA tarefa  
        baseado na descrição  
        void mudarStatus(string d); // inverte o status de uma  
        tarefa baseado na descrição  
        void mostrarTarefas();  
        int size();  
  
        void mostrarTarefasPorDescricao();  
        void mostrarTarefasPorPrioridade();  
        void mostrarTarefasPorStatus();  
};
```

Para testar suas classes crie um programa que contenha, ao menos, o seguinte menu:

```
cout << "Informe a sua opção: ";  
  
    cout << "1 - Adicionar tarefa\n \  
        2 - Mostrar tarefas\n \  
        3 - Mostrar tarefas por descrição\n \  
        4 - Mostrar tarefas por prioridade\n \  
        5 - Mostrar tarefas por status\n \  
        6 - Alterar status da tarefa\n \  
        7 - Remover tarefa\n \  
        8 - Sair\n";
```