

ELE078 - Programação Orientada a Objetos

Atividade Prática 01

Parte 01: Manipulando Programas Simples em C++:

Dica: usar as classes *fstream* , *vector* e *string*.

1.1. Escreva um programa que abra um arquivo e conte o número de espaços em branco do arquivo.

```
In [ ]: #include <string>
#include <fstream>
#include <iostream>
using namespace std;

...
```

1.2. Dado um vetor de números inteiros positivos aleatórios entrados via teclado (número negativo indica fim da entrada dos dados), faça um programa utilizando a classe template vector para comprimir o vetor suprimindo as repetições de números vizinhos através da contagem do número de repetições de cada um da seguinte forma:

- Vetor de entrada: 1 1 1 4 1 1 4 4 25 67 67 67 67 2 2
- Vetor de saída: 3 1 1 4 2 1 2 4 1 25 4 67 2 2

```
In [ ]: #include <iostream>
#include <vector>
#include <cstdio>
using namespace std;

...
```

Parte 02: Manipulando o Tempo:

Crie uma classe denominada **Tempo** que contenha 4 atributos inteiros chamados dia (variação de 0 a um valor máximo não limitado) hora (variação de 0 a 23), min (0 a 59) e seg (0 a 59). Separe a definição da classe em um `.h` e a implementação das funções em um `.cpp`.

- (a) Crie uma função membro da classe para checar a consistência dos atributos e modificá-los caso não sejam consistentes. Por exemplo, caso *dia* = 45, *hora* = 30, *min* = 56 e *seg* = 65, estes valores devem ser transformados em *dia* = 46, *hora* = 6, *min* = 57 e *seg* = 5. Nas funções a seguir a consistência do objeto Tempo resultante deve ser sempre mantida (use esta função para isto)!
- (b) Crie um único construtor para esta classe, que possa ser usado para criar variáveis do tipo Tempo através de declarações como: *Tempo t*, *t1(72)*, *t2(90,3)*, *t3(4, 7,55)*, *t4(45, 30, 56, 65)*. O primeiro parâmetro é o número de dias, o segundo o número de horas, o terceiro o de minutos e o quarto o de segundos. Repare que no objeto *t4*, os parâmetros passados não configuram um objeto válido. Você deve usar a função de (a) para torná-lo válido.
- (c) Crie uma função membro para somar dois objetos do tipo Tempo. Use a função da letra (a) para manter a consistência do resultado;
- (d) Crie funções membro para leitura e impressão de dados via entrada padrão. (novamente, letra (a) ...);
- (e) Crie funções membro que permitam incrementar e decrementar 1 segundo a um objeto do tipo tempo. (novamente, letra (a) ...);

Dica: Você deve usar os recursos aprendidos até o momento para tornar o programa eficiente:

- encapsulamento
- inicialização e destruição de objetos com construtores e destrutor
- sobrecarga de construtores e funções membro
- argumentos default em construtores e funções membro

In []: