UNIVERSIDADE da MADEIRA

ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO

Ano Letivo 2017/2018

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

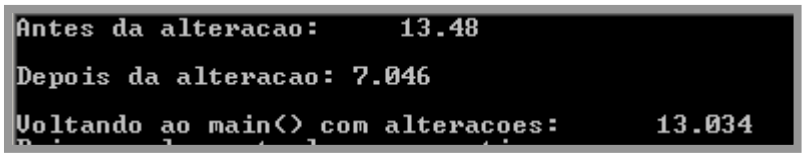
1 º Ano/2 º Semestre

Unidade Curricular: Programação Orientada a Objectos

Docente: Frederica Gonçalves

COLECTÂNEA DE EXERCÍCIOS - VI

1- Reutilizando o código das aulas, sobre o tema das funções – membro *static*, e passagem de argumentos por referência (utilização de ponteiros), crie um **main()** onde atribuirá critérios de **50%** a cada nota dos testes. Crie também, uma função **void muda_nota(aluno *a)**, que irá fazer a alteração da **nota_final**(), com passagem de argumentos por referência. O output deverá ser o seguinte:




```
#include <iostream>
#include <ctime>
using namespace std ;

class aluno {
public:
       static float crit_t1; //vars. public
       static float crit_t2; //satic
       char nome[40] ;
       int numero;
       float nota();
       float nota_final(float i,float j,float l,float m){
             return (i*j)+(l*m);}
};
float aluno::crit_t1;
float aluno::crit_t2;
...
…
float aluno::nota() {
      float n;
      n = rand () % 1501;
      return n/100 + 5;
}
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

…

…

```
void muda_nota(aluno *a){
      aluno::crit_t1 = 0.6;
      aluno::crit_t2 = 0.4;
      cout <<"\nDepois da alteracao: ";
      cout << a->nota_final(a->nota(), a->crit_t1, a->nota(), a->crit_t2)<<"\n";
      }

main(){
        srand((unsigned) time (NULL));
        aluno a1, *p;
        p = &a1;
        aluno::crit_t1 = 0.5;
        aluno::crit_t2 = 0.5;
        cout << "Antes da alteracao:\t";
        cout << a1.nota_final(a1.nota(), a1.crit_t1, a1.nota(), a1.crit_t2)<<"\n";
        muda_nota(p);
        cout << "\nVoltando ao main() com alteracoes:\t";
        cout << a1.nota_final(a1.nota(), a1.crit_t1, a1.nota(), a1.crit_t2)<<"\n";
        system("pause");
        }
```

…

…

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

2 - Considere a seguinte classe que permite guardar os dados de alunos de uma determinada escola (nome e escola a que pertence):

```
class aluno{
public:
static char escola[25];
char nome [20];


aluno(){
    strcpy(nome,"Defina nome");
    cout<<endl<<"Criado um objecto aluno."<<endl;
    }
~aluno(){
    cout<<endl<<"Destruido um objecto aluno"<<endl;
    }
}
```

a) **Elabore a função void setnome()**, que permita definir o nome do aluno, inserido pelo utilizador.

```
void setnome(){
        char nom[20];
        cout<<endl<<"Insira nome do aluno:";
        gets(nom);
        strcpy(nome,nom);
}
```

b) **Elabore a função void getnome()**, que permita apresentar, no monitor, o nome de um objecto aluno.

```
void getnome(){
        cout<<endl;
        cout<<"Nome do aluno: "<<nome;
        cout<<endl;
 }
```

c) **Elabore a função void setescola()**, que permita definir o nome da escola dos alunos, inserido pelo utilizador.

```
void setescola(){
                char nom[20];
                cout<<endl<<"Insira nome da escola:";
                gets(nom);
                strcpy(escola,nom);
  }
```

d) **Elabore a função void getescola()**, que permita apresentar, no monitor, o nome da escola de um objecto aluno.

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

```
void getescola(){
            cout<<endl;
            cout<<"Nome da escola: "<<escola;
            cout<<endl;
}
```

e) **Elabore uma função** para apresentar os dados de um aluno: nome do aluno e nome da escola. Esta função deverá receber como parâmetro um objecto aluno. Utilize o seguinte protótipo: void **ApresentarDadosAluno(aluno a).**

```
void ApresentarDadosAluno(aluno a){
            a.getescola();
            a.getnome();
}
```

f) **Elabore uma função** para definir os dados de um aluno: nome do aluno e nome da escola. Esta função deverá receber como parâmetro **um apontador** para um objecto aluno. Utilize o seguinte protótipo: **void DefinirDadosAlunoApontador(aluno *a).**

```
void DefinirDadosAlunoApontador(aluno *a){
      a->setescola();
      a->setnome();
}
```

g) **Elabore uma função** para apresentar os dados de um aluno: nome do aluno e nome da escola Esta função deverá receber como parâmetro **um apontador** para um objecto aluno. Utilize o seguinte protótipo: **void ApresentarDadosAluno(aluno *a).**

```
void ApresentarDadosAlunoApontador(aluno *a) {
            a->getescola();
            a->getnome();
}
```

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu