

4

Tipos abstratos e Struct em C/C++

Definindo um Struct (REGISTROS)

É uma estrutura do tipo registro e permite o armazenamento de vários tipos de dados (datatypes). Considere a estrutura DISCIPLINAS, com os campos, NOME e MEDIA para armazenar N registros, com o número de LINHAS variando de 0 até N - 1. Podemos criar um STRUCT de quatro maneiras diferentes:

1ª FORMA: typedef vetorizando por dentro

```
int n = 5; // tamanho
typedef struct disci med;

struct disci
{
    double media [ n ];
};

med tab;

tab.media[ 0 ]= 9.2;

tab.media[ 1 ]=10;
```

2ª FORMA: Simples, vetorizando por dentro

```
int n = 5; // tamanho
struct disciplinas
{
    double media [ n ];
};

struct disciplinas tab;

tab.media[0]= 9.2;

tab.media[1]=10;
```

3ª FORMA: Simples, vetorizando por fora

```
int n = 5; // tamanho
struct disci
{
    double media ;
};

struct disci tab[ n ];

tab[0].media=9.4;

tab[1].media=9.4;
```

4ª FORMA: typedef vetorizando por fora

```
int n = 5; // tamanho
typedef struct disci med;

struct disci
{
    double media;
};

med tab [ n ];

tab[0].media=9.4;

tab[1].media=9.4;
```

Programa 1 : código fonte com menu

```
#include "iostream"
#include "cstdlib"
#include "string.h"
using namespace std;
int const n=5;      int linha=-1;

struct disciplinas
{
    string nome[n];
    double media[n];
};

struct disciplinas tbmedias;

string ler_nome() {string nome;
cout<<"\nNome:"; cin>>nome;
return nome; }

double ler_media(){ double media;
cout<<"\nMedia:"; cin>>media;
return media; }

void novoregistro() { linha++;
cin.ignore();
string nom_temp = ler_nome();
tbmedias.nome [linha]= nom_temp;
tbmedias.media [linha]= ler_media();}

void exibir()
{ system("clear");
  for(int i=0;i<=linha;i++)
  {cout<<endl<<tbmedias.nome[i];
   cout<<"-"<<tbmedias.media[i] << endl;
  } system("sleep 3");
}

int main()
{
    int tecla=0;

    while (tecla != 3)
    { system("clear");
      cout<< "\n1 Ler\n2 Exibir\n3 Sair\nitem:";
      cin >> tecla;

      switch(tecla)
      {
          case 1: novoregistro(); break;
          case 2: exibir(); break;
          case 3: exit(0); break;
      }
    }
    return 0;}
```

Programa 2 : código fonte com menu

```
#include "iostream"
#include "cstdlib"
#include "string.h"
using namespace std;

int const n=5; int linha =-1;

struct disciplinas {
    string nome [n];
    double nota1 [n];
    double nota2 [n];
    double media [n];    };

struct disciplinas tbmedias;

string ler_nome (){    string nome;
    cout<<"\nnome:"; getline(cin,nome);
    return nome;    }

double ler_nota1(){    double m;
    cout<<"\nNota1:"; cin>>m;
    return m;}

double ler_nota2(){
    double m;
    cout<<"\nNota2:";
    cin>>m;
    return m;}

void novoregistro () {    linha++;    cin.ignore();
    string nom_temp = ler_nome();
    tbmedias.nome[linha]= nom_temp;
    tbmedias.nota1[linha]= ler_nota1();
    tbmedias.nota2[linha]= ler_nota2();}

void processar ( ) { system("clear");
    for (int i=0; i<=linha;i++)
        { tbmedias.media[i]=(tbmedias.nota1[i] +
        tbmedias.nota2[i])/2;    }
    cout << "processando...." << endl;
    system("sleep 2");    }

void exhibir () { system("clear");
    for ( int i=0; i<= linha; i++)
        {    cout << tbmedias.nome[i];
            cout<< " - " << tbmedias.nota1[i];
            cout<< " - " << tbmedias.nota2[i];
            cout<< " - " << tbmedias.media[i] << endl;}
    system("sleep 3");    }

int main () {    int tecla = 0;
    while (tecla != 4 ){
        system("clear");
        cout<<"\n1 ler\n2 Processar\n3 Exibir\n4 Sair\nItem:";
        cin >> tecla;
        switch (tecla)    {
            case 1: novoregistro();break;
            case 2: processar (); break;
            case 3: exhibir (); break;
            case 4: exit(0);break;
        }    }
    return 0;    }
```

TAREFA / AVALIAÇÃO

Faça o código fonte dos programas A, B da página 26 do livro Estudo dirigido de Algoritmos. a) Os dados de entrada e saída deverão ser armazenados em um struct com várias colunas; b) Deverá conter menu com switch case, funções para leitura dos dados de entrada e as respectivas funções para calcular os dados de saída; c) Tanto os dados de entrada quando os dados de saída deverão ser armazenados dentro de structs.

Nota: Entrega individual do código fonte juntamente com os prints de execução de cada programa dentro de apenas um arquivo no ms-word ou PDF.

Confira meus vídeos de ajuda para Struct:

Busque o Canal do Youtube: Eliseu Lemes C++

Acesse a playlist: Algoritmos e Linguagem de Programação em C++

Assista os vídeos: Aula1 Programação estruturada parte1

Aula2 Programação estruturada parte2

Aula3 Programação estruturada c++ parte3