



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ
Campus Teresina - Central

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO PIAUÍ

CAMPUS TERESINA-CENTRAL

DIRETORIA DE ENSINO

Estrutura de Dados – Listas - Aula 2 Struct e Classes em C++

Professora: Elanne Cristina O. dos Santos

elannecristina.santos@gmail.com

elannecristina.santos@ifpi.edu.br

Definição de tipo - Struct

```
struct aluno {  
    int mat;  
    float nota;  
    char nome[30];  
};  
typedef struct aluno Aluno;
```

```
typedef struct aluno{  
    int matricula;  
    float nota;  
    char nome[30];  
}Aluno;
```

Exemplo

```
struct aluno {
int mat;
float nota;
char nome[30];
};

typedef struct aluno Aluno;

main() {
    Aluno vet_aluno[20];

    int i = 0;

    while (i < 20) {
        printf("Digite a mat:");
        scanf("%d",&vet_aluno[i].mat);
        printf("Digite a nota:");
        scanf("%f",&vet_aluno[i].nota);
        printf("Digite o nome:");
        scanf("%s",vet_aluno[i].nome);
        i = i + 1;

    }

}
```

Exemplo 2 – usando gets ou fgets na leitura de string ou getline(cin,string)

```
int i = 0;
while (i<=2) {
    printf("Digite matricula:");
    scanf("%d",&vet[i].mat);
    printf("Digite a nota:");
    scanf("%f",&vet[i].nota);
    fflush(stdin);
    printf("Digite o nome:");
    //fgets(vet[i].nome,30,stdin);
    gets(vet[i].nome);
    i++;
}
```

OBS: para usar os recursos de entrada e saída da biblioteca *iostream* em C++, é preciso incluir o comando *using namespace std;*

Exemplo 3 – usando *cout* e *cin*

```
for(int i=0;i<3;i++){  
    cout<<"Digite a matricula:";  
    fflush(stdin);  
    cin>>v[i].mat;  
    cout<<"Digite a nome:";  
    fflush(stdin);  
    cin>>v[i].nome;  
    cout<<"Digite a nota:";  
    fflush(stdin);  
    cin>>v[i].nota;  
    cout<<"\n\n";  
}
```

Classes

```
#include <string.h>
#include <iostream>

using namespace std;

class Aluno{
    private:
        int mat;
        string nome;
    public:
        Aluno(int m, string n){
            mat = m;
            nome = n;
        }
};
```

Objetos

```
main(){  
    //Está chamando um construtor da classe  
    // que recebe os parâmetros especificados.  
    Aluno a1(1,"carlos");  
    //criar uma nova alocação de memória e  
    //o endereço desta alocação será atribuído  
    //obviamente para um ponteiro e não diretamente o tipo.  
    Aluno *a2= new Aluno(2,"Jose");  
}
```

Atributos e métodos

```
class Aluno{
    private:
        int mat;
        string nome;
    public:
        Aluno(int m, string n){
            mat = m;
            nome = n;
        }
        string getNome();
};

string Aluno::getNome(){
    return nome;
}

main(){
    Aluno a3(1, "Joao");
    cout<<"Mat: "<<a1.getNome();
    cout<<"Nome: "<<a1.getNome();
}
```


Mais exemplos...

```
#include <stdlib.h>
#include <string>
#include <iostream>
using namespace std;

class Carro {
public:
    string modelo;
    string marca;

    Carro(){}
    Carro(char x[23], string y) {
        modelo=x;
        marca = y;
    }
};
```

Mais exemplos...

```
-  
main(){  
    Carro c1("uno", "fiat");  
    cout<<c1.modelo<<endl;  
    cout<<c1.marca<<endl;  
    Carro *c2= new Carro("palio", "fiat");  
    cout<<c2->modelo<<endl;  
    cout<<c2->marca;  
}
```

Mais exemplos...

```
class Carro{
    private:
        string modelo;
        string marca;
    public:
        Carro(string m1,string m2){
            modelo = m1;
            marca = m2;
        }

        string getModelo();
};

string Carro::getModelo(){
    return modelo;
}
```

Structs & Classes em C++

```
struct Node2 {  
    int id;  
    Node2(){  
        id=0;  
    }  
    int setNode2(int i){  
        id = i;  
    }  
    int getNode2(){  
        return id;  
    }  
};
```

É possível definir os métodos dentro de uma estrutura em C++?

Structs e Classes em C++

❑ **Struct NÃO PERMITE HERANÇA.**

❑ **USANDO REFERENCIA TYPE ou VALUE TYPE:**

- **Reference Type:** Na variável será armazenado apenas um endereço da memória aonde está o objeto, e não a variável por completa.
 - *Ex.: o tipo String também é reference Type*
- **Value Type:** Os objetos Value type se comportam da forma oposta aos Reference Type, um objeto Value Type armazena o objeto inteiro nele, e com isso quando esse valor é atribuído a outra variável a primeira variável possui um objeto e a segunda possui um segundo objeto

Structs e Classes em C++

- Quais serão os valores ao final do programa? Por que?

```
main(){  
    struct Node2 y1;  
    struct Node2 y2;  
    y1.setNode2(3);  
    y2 = y1;  
    y2.setNode2(4);  
    cout<<"(VALUE)y1.getNode2():"<<y1.getNode2()<<endl;  
    cout<<"(VALUE)y2.getNode2():"<<y2.getNode2()<<endl;  
    cout<<"-----"<<endl;}
```

Struct e Classes em C++

- Quais serão os valores ao final do programa? Por que?

```
main(){
```

```
    struct Node2 *x=new Node2();
```

```
    struct Node2 *x2=new Node2();
```

```
    x2 = x;
```

```
    cout<<"x2->getNode2():"<<x2->getNode2()<<endl;
```

```
    x2->setNode2(4);
```

```
    cout<<"x2->getNode2():"<<x2->getNode2()<<endl;
```

```
    cout<<"x->getNode2():"<<x->getNode2()<<endl;
```

```
    cout<<"-----"<<endl;
```

```
}
```

Struct e Classes em C++

```
class Node3 {  
    int id;  
    public:  
    Node3(){  
        id=2;  
    }  
    int setNode3(int i){  
        id = i;  
    }  
    int getNode3(){  
        return id;  
    }  
};
```


Struct e Classes em C++

- Quais serão os valores ao final do programa? Por que?

```
main(){  
    Node3 a1;  
    a1.setNode3(4);  
    Node3 b1;  
    b1=a1;  
    b1.setNode3(5);  
    cout<<"b1.getNode3() :"<<b1.getNode3()<<endl;  
    cout<<"a1.getNode3():"<<a1.getNode3()<<endl;  
}
```

Structs e Classes em C++

- Quais serão os valores ao final do programa? Por que?

```
main(){  
    Node3 *a=new Node3();  
    Node3 *b=new Node3();  
    b=a;  
    a->setNode3(5);  
    cout<<"b.getNode3():"<<b->getNode3()<<endl;  
    cout<<"a.getNode3():"<<a->getNode3()<<endl;  
}
```

Obrigada pela atenção!!!
Boa semana de estudos pra vcs!!