PROGRAMAÇÃO ORIENTADA A OBJETOS

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

Prof. Bruno S. Faiçal
bsfaical@uel.br

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Existem tipos de dados chamados **primitivos**
  - Exemplo: char, int, float, double e void
- Estes elementos armazenam um único item de dado

- Tipos primitivos podem ser combinados dentro de uma estrutura para compor um novo tipo de dado
  - Em contrate com array e matriz, essas estruturas permitem serem compostas por tipos diferentes

- Uma das formas de fazer é utilizando o **struct**

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Sintaxe de definição:

```
struct nome{
    tipo1 var1;
    tipo2 var2;
    …
};
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Sintaxe declaração:

```
struct nome_tipo{
    tipo1 var1;
    tipo2 var2;
    …
};

int main(){
    nome_tipo estrutura_1;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Exemplo 1:

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

struct aluno
{
    int matricula;
    char nome[100];
};

int main(){
    aluno bruno;
}
```

Definição do tipo

Declaração da estrutura

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- As estruturas definidas podem ser **inicializadas**

- Atribuição dos valores devem obedecer a ordem da definição

- Os valores devem ser corretos ao tipo

Sintaxe:

```
nome_tipo estrutura_1 = {10, "Bruno"};
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Exemplo 2:

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

struct aluno
{
    int matricula;
    char nome[100];
};

int main(){
    aluno bruno = {202043, "Bruno S Faiçal"};
}
```

Definição do tipo

Inicializando a estrutura

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Os dados internos de uma estrutura são chamados **membros**

- Os membros devem ser acessados de maneira explicita pelo **nome da variável da estrutura** e o **nome do membro**
  - Os nomes devem estar conectados por . (“ponto”)

  Exemplo: ***estrutura_1.var1***

- Essa sintaxe é usada para exibir e também para inserir/alterar o valor

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Sintaxe acessar aos dados:

```
struct nome_tipo{
    tipo1 var1;
    tipo2 var2;
    …
};

int main(){
    nome_tipo estrutura_1;
    estrutura_1.var1 = 0;
    …
    cout << estrutura_1.var2;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Exemplo 3:

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

struct aluno
{
    int matricula;
    char nome[100];
};

int main(){
    aluno bruno = {202043, "Bruno S Faiçal"};

    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

Acessando os membros

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Nos exemplos anteriores é utilizado um array do tipo char para armazenar o nome do aluno

- Assim, vemos que é possível utilizar outros tipos como membros de novas estruturas
  - Array e Matrizes

- Para o tipo char, a exibição é tratada pelo cout, mas para outros tipos é necessário indicar o índice do elemento que queremos exibir

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Exemplo 4:

```
#include <iostream>
using namespace std;
struct aluno
{
    int matricula;
    char nome[100];
    float nota[4];
};
int main(){
    aluno bruno = {202043, "Bruno S Faiçal"};
    cout << bruno.matricula << ": " << bruno.nome << endl;
    bruno.nota[0] = 7.5;
    bruno.nota[1] = 8.5;
    bruno.nota[2] = 6.0;
    bruno.nota[3] = 7.4;
    for(int i=0; i<4; i++){
        cout << bruno.nota[i] << endl;
    }
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Exemplo 5:

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

struct aluno
{
    int matricula;
    string nome;
};

int main(){
    aluno bruno = {202043};
    cout << "Insira o seu nome: ";
    getline(cin, bruno.nome);
    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Exemplo 5:

Adição do pacote string.h

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

struct aluno
{
    int matricula;
    string nome;
};

int main(){
    aluno bruno = {202043};
    cout << "Insira o seu nome: ";
    getline(cin, bruno.nome);
    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Exemplo 5:

Adição do pacote string.h

Tipo de dado atualizado

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

struct aluno
{
    int matricula;
    string nome;
};

int main(){
    aluno bruno = {202043};
    cout << "Insira o seu nome: ";
    getline(cin, bruno.nome);
    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Exemplo 5:

Adição do pacote string.h

Tipo de dado atualizado

Inicialização parcial

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

struct aluno
{
    int matricula;
    string nome;
};

int main(){
    aluno bruno = {202043};
    cout << "Insira o seu nome: ";
    getline(cin, bruno.nome);
    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

- Exemplo 5:

Adição do pacote string.h

Tipo de dado atualizado

Inicialização parcial

Uso da função getline para obter a linha completa

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

struct aluno
{
    int matricula;
    string nome;
};

int main(){
    aluno bruno = {202043};
    cout << "Insira o seu nome: ";
    getline(cin, bruno.nome);
    cout << bruno.matricula << ": " << bruno.nome << endl;
}
```

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Assim como outras estruturas, é possível **aninhar estructs**
  - Ou seja, utilizar um tipo definido dentro de outro tipo definido

- Assim, é possível manter os dados organizados por sua semântica e reaproveitar o código

- É necessário observar que o acesso aos membros também obedecerão ao aninhamento existente

TIPOS DEFINIDOS PELO PROGRAMADOR EM C++

STRUCT

- Exemplo 6:

```
#include <iostream>
using namespace std;
struct notas{
    float nota[4];
};
struct aluno
{
    int matricula;
    char nome[100];
    notas notas_provas;
};
int main(){
    aluno bruno = {202043, "Bruno S Faiçal"};
    cout << bruno.matricula << ": " << bruno.nome << endl;
    bruno.notas_provas.nota[0] = 7.5;
    bruno.notas_provas.nota[1] = 8.5;
    bruno.notas_provas.nota[2] = 6.0;
    bruno.notas_provas.nota[3] = 7.4;

    for(int i=0; i<4; i++){
        cout << bruno.notas_provas.nota[i] << endl;
    }
}
```

Prof. Bruno S. Faiçal
bsfaical@uel.br