INSTITUTO FEDERAL
RIO GRANDE DO SUL

Estruturas de Dados II

Tipo de dados: Registros (structs)

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

INSTITUTO FEDERAL
RIO GRANDE DO SUL

TIPOS DE DADOS

TIPOS DE DADOS PRIMITIVOS

A cada variável está associado um Tipo de Dados. O tipo de dado define quais os valores que a variável pode conter. Se, por exemplo, dissermos que uma variável é do tipo Inteiro, não poderemos lá colocar um valor Real ou um Caractere.

Ao declararmos o tipo de dados de uma variável, estamos a definir, não só, o tipo de valores que esta pode conter, mas também quais as operações que com elas podemos realizar.

Exemplo de tipos primitivos:

- INT
- REAL
- CHAR
- BOOL

TIPOS DE DADOS

TIPOS DE DADOS COMPOSTOS HOMOGÊNEOS

O nome mais comum atribuído as variáveis compostas homogêneas é vetor ou matrizes.

Vetores, matrizes ou *arrays* correspondem a um tipo de dado utilizado para representar (armazenar e manipular) uma coleção de valores do mesmo tipo.

Uma outra definição para vetores ou matrizes corresponde a um conjunto de variáveis, do mesmo tipo, identificadas por um único nome, onde cada variável (posição) é referenciada por meio de número chamado de índice. Os colchetes são utilizados para conter o índice.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

TIPOS DE DADOS

TIPOS DE DADOS COMPOSTOS HOMOGÊNEOS

As variáveis compostas homogêneas são estruturas de dados que caracterizam-se por um conjunto de variáveis do mesmo tipo.

Elas pode ser unidimensionais ou multidimensionais.

Unidimensionais (*vetores*):

A variável composta homogênea unidimensional caracteriza-se por dados agrupados linearmente numa única direção, como uma linha reta.

Multidimensionais (matrizes):

A variável composta multidimensional caracteriza-se por dados agrupados em diferentes direções, como em um cubo;

INSTITUTO FEDERAL
RIO GRANDE DO SUL

TIPOS DE DADOS

TIPOS DE DADOS COMPOSTOS HETEROGÊNEOS:

As estruturas heterogêneas são conjuntos de dados formados por tipos de dados primitivos diferentes (campos do registro) em uma mesma estrutura.

REGISTROS

Estrutura de Dados – STRUCT

As estruturas de dados consistem em criar apenas um dado que contém vários membros, que nada mais são do que outras variáveis.

De uma forma mais simples, é como se uma variável tivesse outras variáveis dentro dela.

A vantagem em se usar estruturas de dados é que podemos agrupar de forma organizada vários tipos de dados diferentes, por exemplo, dentro de uma estrutura de dados podemos ter juntos tanto um tipo float, um inteiro, um char ou um double.

As variáveis que ficam dentro da estrutura de dados são chamadas de membros.

```
struct nome
{
        tipo1 variavel1;
        tipo2 variavel2;
        tipo3 variavel3;
        ...
};
```

REGISTROS

Exemplo de Criação:

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

typedef struct Cpessoa
{
        char nome[20];
        int idade;
};

int main(void)
{
        Cpessoa aluno[4]; <- Definição do Vetors de Registro (4 posições)

        strcpy(aluno[0].nome, "Celso de Melo");
        aluno[0].idade = 22;

        strcpy(aluno[1].nome, "Augusto Nunes");
        aluno[1].idade = 22;

        strcpy(aluno[2].nome, "Julia Cleria");
        aluno[2].idade = 23;

        strcpy(aluno[3].nome, "Lucas Barroso");
        aluno[3].idade = 24;

        for (int i = 0; i<4 ; i++)
        {
                printf("%s \t %d \n", aluno[i].nome, aluno[i].idade);
        }
        getchar();
}
```

Definição do Registro (struct)

Manipulação

Impressão

INSTITUTO FEDERAL
RIO GRANDE DO SUL

REGISTROS

Ponteiro de Struct

Um struct consiste em vários dados agrupados em apenas um. Para acessarmos cada um desses dados, usamos um ponto (.) para indicar que o nome seguinte é o nome do membro.

Exemplo:

```
aluno[1].idade
```

Um ponteiro guarda o endereço de memória que pode ser acessado diretamente.

```
paluno->idade
```

Registros

```
...
typedef struct Cpessoa
{
        char nome[20];
        int idade;
};

int main(void)
{
        Cpessoa aluno[4];
        Cpessoa *paluno = aluno;

        strcpy(paluno->nome, "Celso de Melo");
        paluno->idade = 22;
        paluno++;

        strcpy(paluno->nome, "Augusto Nunes");
        paluno->idade = 22;
        paluno++;

        strcpy(paluno->nome, "Julia Cleria");
        paluno->idade = 23;
        paluno++;

        strcpy(paluno->nome, "Lucas Barroso");
        paluno->idade = 24;
        paluno++;

        paluno = aluno;

        for (int i = 0; i<4 ; i++)
        {
                printf("%s \t %d \n", paluno->nome, paluno->idade);
                paluno++;
        }

        getchar();
}
```

Criação do Registro (struct)

Criação do ponteiro para a struct

Avança uma posição de memória

Volta para a Posição inicial do ponteiro

Uma possível representação visual para uma struct seria:

```
struct {
  int a;
  char b;
  float c;
  int v[5];
} x;
```

Se fizermos:

x.a = 10;

A caixinha de a receberá o valor 10!