

Tecnologia Em Analise e Desenv. de Sistemas

Estrutura de Dados

Aula 3 – Tipos Abstratos de Dados

Prof. Dr. Ricardo Luis Balieiro

Problemas com strings

- Entrada formatada:
 - **cin >> stringObject;**
 - o operador **>>** formata os dados que recebe através de seu fluxo de entrada; **pula o espaço em branco.**
- Entrada não formatada: **getline** função para um **string**
 - **getline(cin, s)**
 - não pula o espaço em branco.
 - delimitado pela nova linha.
 - lê uma linha inteira de caracteres em **s**.

Problemas com strings

- A função **ignore()** do **cin** serve para limpar qualquer '**\n**' do buffer que possa ter sido inserido em um **cin** anterior.
- Colocar o **cin.ignore()** antes de um **getline(...)**.

Visão geral

No desenvolvimento de software, existem dois aspectos a serem analisados:

- Dados que estes procedimentos irão manipular.
- Procedimentos (funções)

Conceitos básicos

Tipos de dados

- Tipos de dados em linguagens de programação é o conjunto de valores que uma determinada variável pode assumir.

Os tipos de dados são divididos em:

- Tipos primitivos de dados
- Tipos estruturados de dados.

Conceitos básicos

Tipos primitivos de dados:

- Os tipos primitivos de dados são os tipos básicos da linguagem de programação (int, char, float, bool, etc).

Tipos estruturados de dados.

- Os tipos estruturados de dados são construídos a partir dos tipos primitivos e são exemplo de estrutura de dados (vetores, struct, unions)

Tipos Abstratos de Dados (TAD)

Vantagens

- Reuso: Podemos modificar a implementação do TAD sem modificar o código que usa o TAD, e vice-versa.
- Manutenção: TAD pode ser reaproveitado em vários programas ou módulos.

Definindo estruturas

```
#include <iostream>
#include <locale.h> // setlocale
#include <stdlib.h>
#include <string>

using namespace std;

struct Pessoa
{
    string nome;
    int idade;
    float salario;
};
```

Acessando a estruturas



Cada campo (variável) da estrutura pode ser acessada usando o operador “.” (ponto).

Acessando a estruturas

```
#include <iostream>
#include <locale.h> // setlocale
#include <stdlib.h>
#include <string>

using namespace std;

struct Pessoa
{
    string nome;
    int idade;
    float salario;
};
```

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa aluno;

    aluno.nome = "Maria José";
    aluno.idade = 25;
    aluno.salario = 3000.00f;

    cout << "Nome: " << aluno.nome << endl;
    cout << "Idade: " << aluno.idade << endl;
    cout << "Salário: " << aluno.salario << endl;

    system("pause");
    return 0;
}
```

Exercícios

- Crie um programa que cadastre dados de veículo em forma de estrutura dados. Os dados são: modelo, ano de fabricação, número de portas. Em seguida exiba os dados cadastrados.

Exercícios

```
struct Veiculo  
{  
    string modelo;  
    int anoFabricacao;  
    int nroPortas;  
};
```

```
int main() {  
    setlocale(LC_ALL, "Portuguese");  
  
    Veiculo carro;  
  
    carro.modelo = "HB20";  
    carro.anoFabricacao = 2019;  
    carro.nroPortas = 4;  
  
    cout << endl << endl;  
    cout << "Marca: " << carro.modelo << endl;  
    cout << "Modelo: " << carro.anoFabricacao << endl;  
    cout << "Ano Fab.: " << carro.nroPortas << endl;  
  
    system("pause");
```

Usando CIN

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa aluno;

    cout << "Entre com o nome do aluno: ";
    getline(cin, aluno.nome);

    cout << "Entre com a idade: ";
    cin >> aluno.idade;

    cout << "Entre com o salário: ";
    cin >> aluno.salario;

    cout << endl << endl;
    cout << "Nome: " << aluno.nome << endl;
    cout << "Idade: " << aluno.idade << endl;
    cout << "Salário: " << aluno.salario << endl << endl;

    system("pause");
    return 0;
}
```

Exercícios

- Altere o programa de cadastro de dados do veículo para que as entradas dos dados sejam feitas via CIN.

Exercícios

```
struct Veiculo
{
    string modelo;
    int anoFabricacao;
    int nroPortas;
};
```

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Veiculo carro;

    cout << "Entre com o modelo do veículo: ";
    getline(cin, carro.modelo);

    cout << "Entre com o ano de fabricação: ";
    cin >> carro.anoFabricacao;

    cout << "Entre com o número de portas: ";
    cin >> carro.nroPortas;

    cout << endl << endl;
    cout << "Marca: " << carro.modelo << endl;
    cout << "Modelo: " << carro.anoFabricacao << endl;
    cout << "Ano Fab.: " << carro.nroPortas << endl;

    system("pause");
    return 0;
}
```

Inicializando estruturas

```
struct Pessoa
{
    string nome;
    int idade;
    float salario;
};
```

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa aluno = { "João Pedro", 20, 1500.00 };

    cout << endl << endl;
    cout << "Nome: " << aluno.nome << endl;
    cout << "Idade: " << aluno.idade << endl;
    cout << "Salário: " << aluno.salario << endl << endl;

    system("pause");
    return 0;
}
```

Copiando estruturas

```
struct Pessoa
{
    string nome;
    int idade;
    float salario;
};
```

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa aluno1 = { "João Pedro", 20, 1000.00 };
    Pessoa aluno2;

    // COPIA A ESTRUTURA
    aluno2 = aluno1;

    cout << endl << endl;
    cout << "Nome: " << aluno2.nome << endl;
    cout << "Idade: " << aluno2.idade << endl;
    cout << "Salário: " << aluno2.salario << endl << endl;

    system("pause");
    return 0;
}
```

Passando estrutura para função

```
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa aluno;

    aluno.nome = "Maria José";
    aluno.idade = 25;
    aluno.salario = 3000.00f;

    exibirDados(aluno);

    system("pause");
    return 0;
}

void exibirDados(Pessoa paramAluno) {

    cout << endl << endl;
    cout << "Nome: " << paramAluno.nome << endl;
    cout << "Idade: " << paramAluno.idade << endl;
    cout << "Salário: " << paramAluno.salario << endl << endl;
}
```

Exercícios

- Altere o programa, de cadastro de dados do veículo, de forma que os dados sejam passados para uma função e sejam exibidos na tela.

Retornando estrutura de função

```
struct Pessoa
{
    string nome;
    int idade;
    float salario;
};

Pessoa obterDados();
int main() {
    setlocale(LC_ALL, "Portuguese");

    Pessoa alunaMaria;

    alunaMaria = obterDados();

    cout << endl << endl;
    cout << "Nome: " << alunaMaria.nome << endl;
    cout << "Idade: " << alunaMaria.idade << endl;
    cout << "Salário: " << alunaMaria.salario << endl << endl;

    system("pause");
    return 0;
}
```

```
Pessoa obterDados() {
    Pessoa aluno;

    aluno.nome = "Maria José";
    aluno.idade = 25;
    aluno.salario = 3000.00f;

    return aluno;
}
```

Exercícios

- Implemente um estrutura de ContaBancaria, com os seguintes dados do cliente: código do cliente, nome, número da conta e saldo. Em seguida crie as seguintes funções:
 - **obterDados(...)**: obtém os dados da conta bancária e os retorna para a função main().
 - **imprimirDados(...)**: recebe como parâmetro os dados da conta bancária e os imprime na tela.

Vetor de Estrutura

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

using namespace std;

struct Produto
{
    int CodProduto;
    string descricao;

};
```

Vetor de Estrutura

```
int main() {  
  
    Produto Prod[2];  
  
    cout << "Entre com código do produto 1: ";  
    cin >> Prod[0].CodProduto;  
  
    cout << "Entre com a descrição do produto 1: ";  
    getline(cin, Prod[0].descricao);  
  
    cout << "Entre com código do produto 2: ";  
    cin >> Prod[1].CodProduto;  
  
    cout << "Entre com a descrição do produto 2: ";  
    getline(cin, Prod[1].descricao);
```

Vetor de Estrutura

```
cout << endl << endl << "Exibindo informações." << endl;
cout << "===== Produto 1 =====" << endl;
cout << "Código: " << Prod[0].CodProduto << endl;
cout << "Descrição: " << Prod[0].descricao << endl << endl;

cout << "===== Produto 2 =====" << endl;
cout << "Código: " << Prod[1].CodProduto << endl;
cout << "Descrição: " << Prod[1].descricao << endl << endl;

system("pause");
return 0;
}
```

F I M

“Nossa história, depositada nas mãos de Deus, pode ser reescrita a qualquer momento”.

(Pe. Luís Erlin - 9 Meses com Maria)

Prof. Dr. Ricardo Luis Balieiro
ricbalieiro@gmail.com