Estácio

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ESTRUTURA DE DADOS

AULA 1

RICARDO EIJI KONDO, Me.

TÓPICOS

UNIDADE I - Agregados Homogêneos e Heterogêneos

1.1 - Revisão dos Agregados Homogêneos

1.2 - Agregados Heterogêneos

1.2.1 - Fundamentos

1.2.2 - Aplicação em C/C++: Structs

1.2.2.1 - Operador ponto

CONCEITO DE ESTRUTURAS DE DADOS

- Representar as relações lógicas entre os dados de uma forma coerente para que possam ser processadas e registradas pelo computador.
- Criar programas com uma representação dos dados mais relevante para um problema real, de forma mais clara e limpa. Criação de procedimentos mais simples e eficiente melhorando o tempo de execução de um programa, melhora a leitura do código fonte, facilidade de manutenção e diminuição de custos.

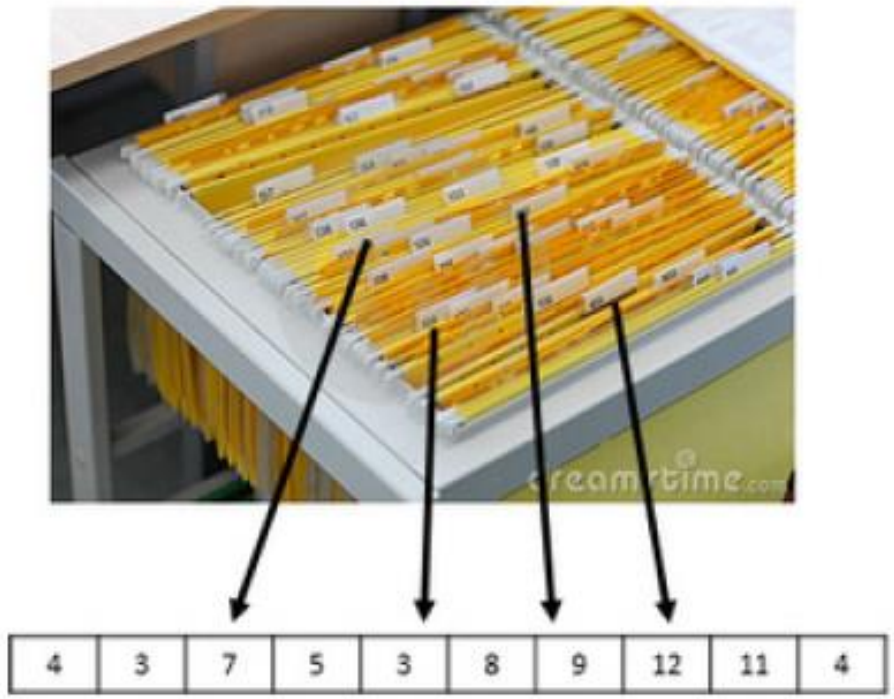
> De acordo com Mizrahi (2006), uma estrutura de dados pode ser definida como sendo uma coleção de variáveis, podendo ser tipos iguais ou diferentes, reunidas sob um único nome. Vários autores denominam estrutura de dados como sendo registros.

CONCEITO DE ESTRUTURAS DE DADOS

- Estruturas de dados podem ser divididas em duas formas: **homogênea** (vetores e matrizes) e **heterogênea** (registros).
  - **Homogêneas**: dados de um único tipo, como por exemplo, string ou inteiros. São empregadas em situações onde as informações podem ser organizadas em um único tipo de dados, normalmente utilizando vetores e matrizes
  - **Heterogêneas**: representação dos dados composta por diferentes tipos simultaneamente, tais como, string, inteiros, datas, etc.

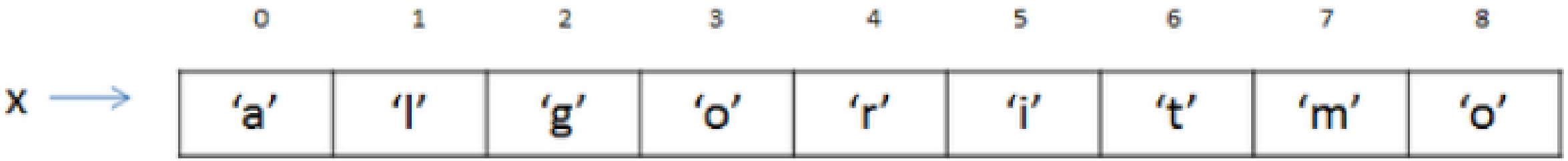
ESTRUTURAS HOMOGÊNEAS

- Variáveis compostas unidimensionais (vetor)
- Variáveis compostas multidimensionais (matriz)



ESTRUTURAS HOMOGÊNEAS

Um vetor, ou array, ou variável composta unidimensional é uma estrutura de dados contendo um conjunto de dados, todos do mesmo tipo, indexados por um valor.

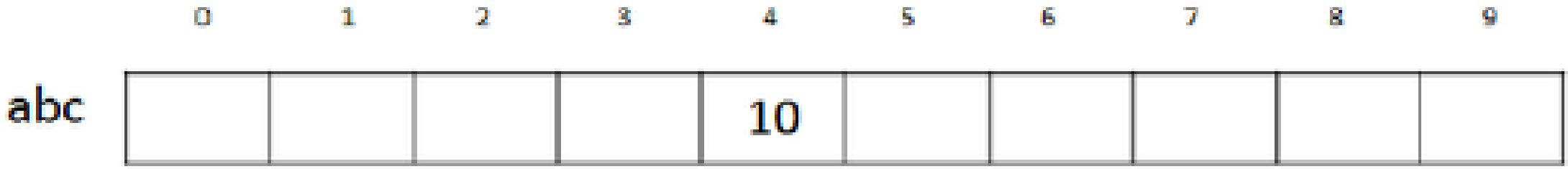


ESTRUTURAS HOMOGÊNEAS

- Índice

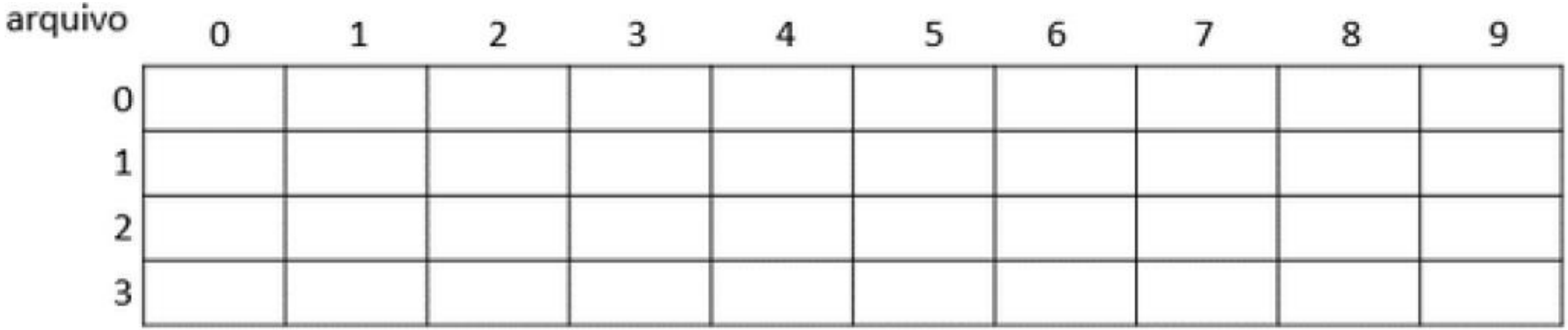
A posição de um elemento dentro de um vetor é feito por uma variável numérica chamada índice.

```
1  #include <iostream>
2  using namespace std;
3
4  int main (void){
5      int abc[10];
6
7      abc[4]=10;
8  }
```



ESTRUTURAS HOMOGÊNEAS

Esta estrutura possui vários nomes: estrutura composta homogênea multidimensional, matriz, arranjo bidimensional, array bidimensional ou vetor bidimensional.



ESTRUTURAS HOMOGÊNEAS

- **Atribuir valores a matriz:**

M	0	1	2	3
0		7		12
1	10		5	
2		6		11
3	9		8	

```
 4  int main (void){
 5
 6      int m[4][4];
 7      int a, b;
 8
 9      m[1][2]=5;
10      m[2][1]=6;
11      m[0][1]=7;
12      a=3;
13      b=2;
14      m[a][b]=8;
15      m[a][b-2]=9;
16      m[a-2][b-2]=10;
17      m[b][a]=11;
18      m[b-2][a]=12;
19
20      return 0;
21  }
```

EXERCÍCIOS

- Fazer um programa utilizando vetores, que lê 5 valores e imprime o maior e o menor valor lido.
- Fazer um programa utilizando vetores, que lê 5 valores e imprime em ordem crescente.
- Escrever um algoritmo para armazenar valores inteiros em uma matriz (3,3). A seguir, calcular a média dos valores pares contidos na matriz e escrever seu conteúdo.

EXERCÍCIOS

- Fazer um programa utilizando um vetor com 10 posições “v[10]” , inicialize-o com os valores { 5, 10, 15, 16, 20, 21, 29, 10, 14, 15 }.
  - Mostre na tela em qual(is) posição(ões) está o número 15 – utilize loop de repetição para percorrer os índices;
  - Mostre na tela o valor e em quais posições existem números múltiplos de 5;
  - Mostre na tela a média aritmética dos valores ímpares;
  - Onde houver números ímpares, transforme-os em pares somando +1 e mostre na tela o novo vetor
- Fazer um programa utilizando uma matriz 3x5 “m[3][5]”, inicialize-o com os valores { {5, 10, 15, 20, 25},
  {7, 14, 21, 28, 15},
  {9, 15, 27, 36, 45} }.
  - Mesmas questões anteriores.

ESTRUTURAS HETEROGÊNEAS

As estruturas de dados, ou como são chamadas também de registros, são definidas através de **struct**. Em C++, utilizando struct é possível definir um novo tipo de dado composto por várias variáveis, denominadas membros da estrutura.

```
struct DADOS_ALUNO{
    int CodAluno;
    char Nome[100];
    int Turma;
};
```

ESTRUTURAS HETEROGÊNEAS

- Estrutura Local – pode ser utilizada somente por aquela função

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

void main(){
    struct DADOS_ALUNO{
           int CodAluno;
           char Nome[100];
           int Turma;
    };

    struct DADOS_ALUNO AlunoA;

    AlunoA.CodAluno = 10;
    strcpy(AlunoA.Nome, "Gabriela");
    AlunoA.Turma = 250;

    cout << "Código do Aluno: " << AlunoA.CodAluno << endl;
    cout << "Nome: " << AlunoA.Nome << endl;
    cout << "Turma: " <<  AlunoA.Turma << endl;

    system("pause > null");
}
```

ESTRUTURAS HETEROGÊNEAS

- Estrutura Global – pode ser utilizada por todas as funções na aplicação

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

struct DADOS_ALUNO{
    int CodAluno;
    char Nome[100];
    int Turma;
};
struct DADOS_ALUNO AlunoA;
```

```
void main(){
    AlunoA.CodAluno = 10;
    strcpy(AlunoA.Nome, "Gabriela");
    AlunoA.Turma = 250;

    cout << "Código do Aluno: " << AlunoA.CodAluno << endl;
    cout << "Nome: " << AlunoA.Nome << endl;
    cout << "Turma: " <<  AlunoA.Turma << endl;

    system("pause > null");
}
```

EXERCÍCIOS

1) Crie uma estrutura de dados chamado Carro, com os membros Cor, placa, QtdePessoas. Atribua valores “Azul”, “AAA-9999” e 5 para os membros. Imprima na tela os valores.
2) Para o mesmo exercício anterior, crie um vetor com 5 posições para a estrutura Carro. Receba do teclado em qual posição do vetor deverá ser escrito os valores. Imprima a lista de todas as posições do vetor.
3) Crie um programa que receba do teclado os valores para preencher as 5 posições do vetor Carro. Escolha através do teclado qual valor deverá ser mostrado na tela.