

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Sistemas de Informação - noite

Laboratório de Algoritmos e Técnicas de Programação

Profs. Soraia Lúcia da Silva / Faber Henrique

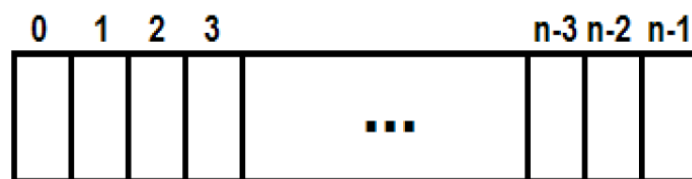
AULA 8 (2/2018) – ARRAYS E ARRAYS BIDIMENSIONAIS

1. ARRAYS

- LEITURA

– DEFINIÇÃO:

· Conjunto de variáveis que pode ser endereçada por um índice



- Posições válidas de 0 a (n-1)
- Cuidado: Frequentemente, os programadores erram e acessam posições negativas ou maiores que (n-1)

- DECLARAÇÃO:

```
tipo [ ] variável;
```

– DEFINIÇÃO DO TAMANHO:

```
int[ ] vet1 = new int[10];
```

ou

```
int[ ] vet2;  
vet2 = new int[10];
```

ou

```
int[ ] vet3 = {5, 6, 20, 40, 2, 34, 87, 3, 1, 4};
```

Observação 1: O valor inicial das posições de vet1 e vet2 será zero e de vet3 será os valores passados entre { }.

– ACESSAR O CONTEÚDO DE UMA POSIÇÃO DO VETOR:

```
int[ ] vet = new int [3];
```

```
//Quantas posições o nosso arrays possui?  
//Quais são as posições que podem ser acessadas?
```

```
vet[0] = 5;  
vet[1] = 9;  
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];  
vet[1]--;
```

```

//O nosso array pode ser indexado a partir de variáveis.

int posicao = 0;
vet[posicao] = 3;
for(int i = 0; i < 3; i++){
    ler: vet[i]; //Em C# vet[i] =int.parse(Console.ReadLine( ));
}

for(int i = 0; i < 3; i++){
    escrever: vet[i];
}

```

- DECLARAÇÃO MÚLTIPLA:

```
double[ ] v1, v2, v3;
```

- ERRO INDEX OUT OF RANGE EXCEPTION:

- Acesso a posições maiores que (n-1)

```

int n = 10;
int vet1[ ] = new int[n];
...
for(int i = n; i >= 0; i--){
    escreva(vet1[i]);
}

```

- Acesso a posições negativas

```

int n = 10;
int vet1[ ] = new int[n];
...
for(int i = n - 1; i >= 0; i--){
    vet1[i-1] = vet1[i];
}

```

- TAMANHO DO ARRAY:

```

public static void Main(String[] args)
{
    int[] vet = new int[10];
    for (int i = 0; i < vet.Length; i++)
    {
        Escrever(vet[i]);
    }
}

```

- EXEMPLO

```

class AulaArray
{
    //Faca um metodo que leia os elementos de um array de tamanho n e mostre o
    maior elemento do array

    public static void Exemplo01() {
        int n, maior;
        int[] vet;
    }
}

```

```
Escrever("Entre com o tamanho do array:");
Ler(n);

vet = new int[n];

for (int i = 0; i < n; i++){
    Escrever("Entre com um elemento:");
    vet[i] = int.Parse(Console.ReadLine());
}
maior = vet[0];
for (int i = 1; i < n; i++){

    if (vet[i] > maior){
        maior = vet[i];
        Escrever("Atualizando maior...");
    }
}
Escrever("Maior elemento: " + maior);
}
```

1. EXERCÍCIOS - Arrays

- 1) Faça um programa que leia um conjunto de 100 idades e armazene-as em um vetor. Depois mostre as idades que são maiores que a média de idades do grupo. Observe que o resultado esperado (idades maiores que a média do grupo) depende do cálculo da média que só obteremos após ler todos os dados.
- 2) Faça um programa que preencha um vetor de 12 posições e em seguida leia também dois valores X e Y quaisquer correspondentes a duas posições no vetor. Ao final seu programa deverá escrever a soma dos valores encontrados entre as respectivas posições X e Y. Você deve verificar se estes elementos fazem parte dos índices do vetor.
- 3) Faça um programa que leia N números (N será fornecido pelo usuário) e coloque em um vetor. Depois mostre o menor número lido e em quais posições ele aparece no vetor. Exemplo: {3, 5, 2, 7, 4, 2, 9, 5}, o menor número lido é o 2 e aparece nas posições 2 e 5.