

# Fundamentos de Programação

## Linguagem C++: arrays - aula I

Bruno Emerson Gurgel Gomes<sup>1</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia do RN (IFRN)

2012

## 1 Introdução

## 2 Vetores

- Inicializando vetores
- Imprimindo os elementos de um vetor
- Exemplos

- Os tipos primitivos (*int*, *bool*, *char*, *float*, *double*) não são suficientes para representar ou armazenar todas as possíveis informações em um programa
- Uma variável simples armazena apenas 1 valor de um certo tipo;
  - E se quisermos *guardar*, para usar no programa, um conjunto de valores relacionados ?
  - exemplos: as notas de uma turma, uma lista de nomes de pessoas, um conjunto de números que devem estar em uma certa ordem, etc.

- Um *array* é uma variável que armazena, em sequência, um **conjunto** de informações
  - de um mesmo tipo de dados
  - de tamanho finito e fixo (definido na declaração)
- Os *arrays* podem ser chamados de
  - “vetor”: um índice, correspondente a um conjunto de valores em uma linha
  - “matriz”: mais de um índice, correspondente a uma matriz de valores (linhas e colunas)

- A declaração de um vetor tem a forma:  
 $\langle \textit{tipo} \rangle \textit{nome}[n]$ ; onde  $n$  é o número de elementos do vetor
- Exemplo: **int notas [10]**
  - declara um vetor de inteiros (*int*)
  - o nome “notas” deve ser utilizado para acessar o vetor
  - pode guardar até 10 elementos
- Um *elemento* do vetor é acessado através do seu nome e da posição que ele ocupa (índice)
  - Primeiro elemento: índice 0
  - Último elemento: índice  $n - 1$ , sendo  $n$  o tamanho do vetor
  - No vetor **notas** o último elemento é acessado pelo índice 9 ( $a[9]$ )
- Pode ser utilizada uma expressão aritmética para se obter um elemento:
  - $\textit{numeros}[1+3]$  retorna o elemento da posição 4

- Os elementos do vetor devem ser fornecidos antes da sua utilização
- A declaração de um vetor na forma “<tipo> <nome>[n]” reserva apenas o espaço de memória correspondente ao tamanho do vetor
- Inicialização pode ser:
  - Durante a declaração do vetor
  - Depois da declaração, em tempo de execução

## Exemplos de inicialização na declaração

```
int v2[5] = {3, 4, 6, 12, 16};  
short v3[] = {20, 30};
```

## Exemplo de inicialização na execução com atribuição

```
int v1[10];  
for (int i = 0; i < 10; i++) {  
    v1[i] = i+1;  
}
```

## Exemplo de inicialização na execução com leitura

```
int v1[10];  
for (int i = 0; i < 10; i++) {  
    cin >> v1[i];  
}
```

- Vetores declarados, mas não inicializados
  - Os elementos do vetor possuem valores indeterminados (qualquer valor que esteja na memória)
- Vetores inicializados parcialmente na declaração
  - Aos elementos restantes é atribuído valores-padrão (0, *false*, *null*, etc.)
  - exemplo: `int v2[5] = {2, 3, -10}`. Os outros 2 elementos serão 0
- **Atenção:** A atribuição de um vetor a outro não é permitida em C++.



# Imprimindo os elementos de um vetor

- Para imprimir todos os elementos de vetor:
  - Use uma estrutura de repetição para percorrer todo o vetor
  - Por ser mais compacta, normalmente usa-se o “for”

## Imprimindo os elementos de um vetor de tamanho 50

```
for (int i = 0; i < 50; i++) {  
    cout << v1[i] << endl;  
}
```

- 1 Dado um vetor com 10 números reais, faça um algoritmo que percorra o vetor e imprima o seu maior elemento.
- 2 Utilizando vetor, imprimir os  $n$  primeiros termos da série de *Fibonacci*:  
1 2 3 5 8 ...  $n$

# Exemplo: números de dias transcorridos a partir de uma data (dentro de um mesmo ano)

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

int main() {
    short dmes[12] =
        {31, 28, 31, 30,
         31, 30, 31, 31,
         30, 31, 30, 31};

    short dia, mes, ano;

    cout <<
        "Digite uma
         data(dd/mm/aaaa)";
    char ch;
    cin >> dia >> ch >>
    mes >> ch >> ano;

    if (ano % 4 == 0 && ano % 100
        || ano % 400 == 0) {
        dmes[1] = 29;
    }

    int total = dia;

    for (short i = 0; i < mes-1; i++) {
        total += dmes[i];
    }

    cout <<
        "Total de dias transcorridos: "
        << total << endl;

    return 0;
}
```

- 1 Defina dois vetores com 5 elementos cada. Faça um algoritmo que calcule e imprima a soma deles. Ou seja, a soma de cada elemento de um vetor pelo elemento correspondente do outro vetor.
- 2 Leia a nota de 38 alunos de uma turma e calcule a média das notas, a maior e a menor nota.