

Linguagens de Programação

Prof. Miguel Elias Mitre Campista

<http://www.gta.ufrj.br/~miguel>

Parte II

Introdução à Programação em C++
(Continuação)

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Relembrando da Última Aula...

- Funções
- Classes de armazenamento
- Regras de escopo
- Funções sobrecarregadas
- Templates
- Mais exemplos de programação orientada a objetos...

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Arrays

- Estruturas de dados que contêm itens de dados relacionados do mesmo tipo
- Tamanho constante desde o momento em que são criados
 - Entidades "estáticas"
- Arrays de caracteres podem também representar strings
- Arrays podem ser representados como em C
 - Entretanto, podem também ser objetos vetores como implementado na STL (*Standard Template Library*)
 - Os vetores são mais seguros e versáteis

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Arrays

- Grupo consecutivo de posições da memória
 - Todas são do mesmo tipo
- Índice
 - Número da posição usado para indicar uma localização/elemento específico
 - Deve ser um inteiro positivo ou uma expressão do tipo inteiro
 - O primeiro elemento tem índice zero
 - Ex.: Suponha $a = 5$ e $b = 6$
 - $c[a + b] += 2$;
 - » Adiciona 2 ao elemento do array $c[11]$.

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Array

- Inicialização de um array em uma declaração com uma lista inicializadora
 - Lista inicializadora
 - Os itens encontram-se entre chaves $\{\}$
 - Os itens na lista são separados por vírgula
 - Ex.: `int n[] = { 10, 20, 30, 40, 50 };`
 - Pelo fato de o tamanho do array ser omitido na declaração, o compilador determina o tamanho do array com base no tamanho da lista inicializadora
 - Os valores do índice são 0, 1, 2, 3, 4
 - São inicializados nos valores 10, 20, 30, 40, 50, respectivamente.

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Array

- Inicialização de um array em uma declaração com uma lista inicializadora
 - Se houver menos inicializadores que elementos no array
 - Os elementos remanescentes são inicializados em zero
 - Ex.: `int n[10] = { 0 };`
 - » Inicializa explicitamente o primeiro elemento em zero
 - » Inicializa implicitamente os nove elementos restantes em zero
 - Se houver mais inicializadores que elementos no array
 - Ocorrerá erro de compilação

Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 2
 * Autor: Miguel Campista
 */
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    int n [] = {32, 27, 2, 5, 22, 90, 1, 10, 67, 99}; // Array n de 10 inteiros

    cout << "Elemento" << setw(13) << "valor" << endl;

    // Saída dos elementos do array
    for (int j = 0; j < 10; j++)
        cout << setw(7) << j << setw(13) << n [j] << endl;

    return 0;
}
```

Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 2
 * Autor: Miguel Campista
 */
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    int n [] = {32, 27, 2, 5, 22, 90, 1, 10, 67, 99}; // Array n de 10 inteiros

    cout << "Elemento" << setw(13) << "valor" << endl;

    // Saída dos elementos do array
    for (int j = 0; j < 10; j++)
        cout << setw(7) << j << setw(13) << n [j] << endl;

    return 0;
}
```

Lista inicializadora utilizada para inicializar o array. Compilador aloca memória conforme o número de elementos declarados

Exemplo Usando Array em C++

```
shell->$ g++ exemplo.cpp -o ex2
```

```
shell->$ ./ex2
```

Elemento	valor
0	32
1	27
2	2
3	5
4	22
5	90
6	1
7	10
8	67
9	99

```
shell->$ return 0;
```

Array

- Fornecer mais inicializadores em uma lista inicializadora de array do que o número de elementos existentes no array é um erro de compilação
- Esquecer de inicializar os elementos de um array cujos elementos deveriam ser inicializados é um erro de lógica

Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 3
 * Autor: Miguel Campista
 */
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    const int numElem = 10;
    int n [numElem]; // Array n de 10 inteiros

    // Cálculo dos elementos do array
    for (int i = 0; i < numElem; i++)
        n [i] = 2 + 2 * i;

    cout << "Elemento" << setw(13) << "valor" << endl;

    // Saída dos elementos do array
    for (int j = 0; j < 10; j++)
        cout << setw(7) << j << setw(13) << n [j] << endl;

    return 0;
}
```

Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 3
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
using namespace std;

int main() {
    const int numElemo = 10;
    int n [numElemo]; // Arr. n de 10 inteiros

    // Cálculo dos elementos do array
    for (int i = 0; i < numElemo; i++)
        n[i] = 2 + 2 * i;

    cout << "Elemento" << setw(13) << "valor" << endl;

    // Saída dos elementos do array
    for (int j = 0; j < 10; j++)
        cout << setw(7) << j << setw(13) << n [j] << endl;

    return 0;
}

```

Declaração do número de elementos do array utilizando uma variável **const**

Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 3
 * Autor: Miguel Campista
 */
shell-$ g++ exemplo.cpp -o ex3
shell-$ ./ex3
Elemento      valor
0             2
1             4
2             6
3             8
4            10
5            12
6            14
7            16
8            18
9            20
shell-$

```

Array

- Variáveis constantes
 - São declaradas usando-se o qualificador **const**
 - São também chamadas de constantes identificadas ou variáveis de leitura
 - Devem ser inicializadas com uma sentença constante quando são declaradas e não podem ser modificadas posteriormente
 - Podem ser colocadas em qualquer lugar em que se espera uma expressão constante
 - Usar variáveis constantes para especificar o tamanho do array torna os programas mais escaláveis e torna o programa mais fácil de modificar

Array

- Variáveis constantes
 - Não atribuir um valor a uma variável constante quando ela é declarada é um erro de compilação
- Atribuir um valor a uma variável constante em uma instrução executável é um erro de compilação

```
const int x;
```

```
const int x = 1;
x = 2;
```

Array

- Variáveis constantes
 - Não atribuir um valor a uma variável constante quando ela é declarada é um erro de compilação
- Atribuir um valor a uma variável constante em uma instrução executável é um erro de compilação

```
const int x; X Errol
```

```
const int x = 1;
x = 2; X Errol
```

Array

- Recomendação
 - Definir o tamanho de um array como uma variável constante torna os programas mais claros
 - Técnica que elimina os chamados "números mágicos"
 - Ex.: Mencionar repetidamente o tamanho 10 em código de processamento de array para um array de 10 elementos dá para o número 10 uma importância artificial que pode confundir o leitor quando o programa incluir outros números 10 que não estão relacionados com o tamanho do array

Array

- Emprego de arrays de caracteres para manipular strings
 - Os arrays podem ser de qualquer tipo, incluindo chars.
 - Strings de caracteres podem ser armazenados em arrays char
 - Podem ser inicializados por meio de um literal de string.
 - Ex: `char string1[] = "Hi";`
 - Equivalente a
 - `char string1[] = { 'H', 'i', '\0' };`
 - O array contém cada caractere mais um caractere especial de terminação de string denominado caractere nulo ('\0')

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Array

- Emprego de arrays de caracteres para manipular strings
 - Podem também ser inicializados com constantes de caractere individuais em uma lista inicializadora
 - Ex: `char string1[] = { 'f', 'i', 't', 's', 't', '\0' };`
 - Além disso, podem inserir uma string diretamente em um array de caracteres por meio do teclado, usando `cin`
 - Ex: `cin >> string1;`
 - `cin >>` talvez leia mais caracteres que o array pode armazenar

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Sétimo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 7
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

int main() {
    char string1 [20];
    char string2[] = ("string literal"); // reserva 15 caracteres

    // Lê a string fornecida pelo usuário para string1
    cout << "Entre com a string \"Hello there!\": ";
    cin >> string1;

    // Strings recebidas
    cout << "string1 eh: " << string1 << "string2 eh: " << string2 << endl;

    cout << "string1 com espacos entre caracteres eh:\n";
    for (int i = 0; string1[i] != '\0'; i++)
        cout << string1[i] << " ";

    cin >> string1; // Lê there
    cout << "string1 eh: " << string1 << endl;

    return 0;
}
```

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Sétimo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 7
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

int main() {
    char string1 [20];
    char string2[] = ("string literal"); // reserva 15 caracteres

    // Lê a string fornecida pelo usuário para string1
    cout << "Entre com a string \"Hello there!\": ";
    cin >> string1;

    // Strings recebidas
    cout << "string1 eh: " << string1 << "string2 eh: " << string2 << endl;

    cout << "string1 com espacos entre caracteres eh:\n";
    for (int i = 0; string1[i] != '\0'; i++)
        cout << string1[i] << " ";

    cin >> string1; // Lê there
    cout << "string1 eh: " << string1 << endl;

    return 0;
}
```

Armazena "string literal" como um array de caracteres

Inicialização de um array por meio de cin

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Sétimo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 7
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

int main() {
    char string1 [20];
    char string2[] = ("string literal"); // reserva 15 caracteres

    // Lê a string fornecida pelo usuário para string1
    cout << "Entre com a string \"Hello there!\": ";
    cin >> string1;

    // Strings recebidas
    cout << "string1 eh: " << string1 << "string2 eh: " << string2 << endl;

    cout << "string1 com espacos entre caracteres eh:\n";
    for (int i = 0; string1[i] != '\0'; i++)
        cout << string1[i] << " ";

    cin >> string1; // Lê there
    cout << "string1 eh: " << string1 << endl;

    return 0;
}
```

Loop até encontrar o caractere nulo

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Sétimo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 7
 * Autor: Miguel Campista
 */
#include <iostream>

int main() {
    char string1 [20];
    char string2[] = ("string literal"); // reserva 15 caracteres

    // Lê a string fornecida pelo usuário para string1
    cout << "Entre com a string \"Hello there!\": ";
    cin >> string1;

    // Strings recebidas
    cout << "string1 eh: " << string1 << "string2 eh: " << string2 << endl;

    cout << "string1 com espacos entre caracteres eh:\n";
    for (int i = 0; string1[i] != '\0'; i++)
        cout << string1[i] << " ";

    cin >> string1; // Lê there
    cout << "string1 eh: " << string1 << endl;

    return 0;
}
```

```
shell-> g++ exemplo.cpp -o ex7
shell-> ./ex7
Entre com a string "Hello there": hello there
string1 eh: hello
string2 eh: string literal
string1 com espacos entre caracteres eh:
h e l l o
string1 eh: there
shell->
```

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Array

- Arrays locais `static` e arrays locais automáticos
 - Uma variável local `static` em uma função
 - Existe durante a execução do programa
 - Mas é visível apenas no corpo da função
 - Um array local `static`
 - Existe durante a execução do programa
 - É inicializado quando sua declaração é encontrada pela primeira vez
 - Todos os elementos são inicializados em zero, se não forem inicializados explicitamente
 - » Isso não ocorre com os arrays locais automáticos

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Oitavo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 8
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

// Protótipo de funções
void staticArrayInit();
void automaticArrayInit();

int main() {
    cout << " ** Primeira chamada para cada funcao:\n\n";
    staticArrayInit();
    cout << endl;
    automaticArrayInit();

    cout << "\n\n ** Segunda chamada para cada funcao:\n\n";
    staticArrayInit();
    cout << endl;
    automaticArrayInit();
    cout << "\n\n" << endl;

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Oitavo Exemplo Usando Array em C++

```
void staticArrayInit() {
    //static int array1 [3]; // como 0 na primeira vez que é chamada
    // Saída do array
    for (int i = 0; i < 3; i++)
        cout << "array1[" << i << "] = " << array1[i] << " ";
    cout << "\nValores no array estatico:\n";
    for (int i = 0; i < 3; i++)
        cout << "array1[" << i << "] = " << (array1[i] + 5) << " ";
    cout << "\nValores no array estatico ao sair da funcao:\n";
}

void automaticArrayInit() {
    // automaticamente inicializado com 0 na primeira vez que é chamada
    int array2 [3] = {1, 2, 3};
    for (int i = 0; i < 3; i++)
        cout << "array2[" << i << "] = " << array2[i] << " ";
    cout << "\nValores atribuidos ao array automatico:\n";
    for (int i = 0; i < 3; i++)
        cout << "array1[" << i << "] = " << (array2[i] + 5) << " ";
    cout << "\nValores no array automatico ao sair da funcao:\n";
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Oitavo Exemplo Usando Array em C++

```
void staticArrayInit() {
    // Inicializa elementos como 0 na primeira vez que é chamada
    static int array1 [3];
}

// C:\Documents and Settings\Campista\Meus documentos\vaugur\undo7\ouf.exe
array1[0] = 0 array1[1] = 0 array1[2] = 0
Valores no array estatico
array1[0] = 5 array1[1] = 5 array1[2] = 5
Valores no array estatico ao sair da funcao
array2[0] = 1 array2[1] = 2 array2[2] = 3
Valores atribuidos ao array automatico
array1[0] = 6 array1[1] = 7 array1[2] = 8
Valores no array automatico ao sair da funcao

** Segunda chamada para cada funcao:
array1[0] = 5 array1[1] = 5 array1[2] = 5
Valores no array estatico
array1[0] = 10 array1[1] = 10 array1[2] = 10
Valores no array estatico ao sair da funcao
array2[0] = 1 array2[1] = 2 array2[2] = 3
Valores atribuidos ao array automatico
array1[0] = 6 array1[1] = 7 array1[2] = 8
Valores no array automatico ao sair da funcao

cout << "\nValores atribuidos ao array automatico:\n";
for (int i = 0; i < 3; i++)
    cout << "array1[" << i << "] = " << (array2[i] + 5) << " ";
cout << "\nValores no array automatico ao sair da funcao:\n";
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Para passar um argumento array a uma função
 - Especifique o nome do array sem os colchetes
 - O array `hourlyTemperatures` é declarado como
 - `int hourlyTemperatures[24];`
 - A chamada de função
 - `modifyArray(hourlyTemperatures, 24);`
 - passa o array `hourlyTemperatures` e o seu tamanho à função `modifyArray`
 - O tamanho do array é normalmente passado como outro argumento, de modo que a função possa processar o número específico de elementos no array

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Os arrays são passados por referência
 - A chamada de função na verdade passa o endereço inicial do array
 - Portanto, a função sabe em que posição o array se encontra na memória
 - O chamador concede à função chamadora acesso direto aos dados do chamador
 - A função chamada pode manipular esses dados

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Passar arrays por referência é mais eficiente
 - Arrays passados por valor requerem uma cópia de cada elemento
 - Arrays grandes passados com frequência demandaria tempo e exigiria espaço de armazenamento considerável para as cópias dos elementos do array

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Elementos individuais do array passados por valor
 - Fragmentos de dados simples
 - São chamados escalares ou quantidades escalares
 - Para passar um elemento a uma função
 - Use o elemento do array com o seu índice como um argumento
- Funções que consideram os arrays como argumentos
 - A lista de parâmetros da função deve especificar um parâmetro de array
 - Ex: `void modArray(int b[], int arraySize);`

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Funções que consideram os arrays como argumentos
 - O parâmetro de array pode incluir o tamanho do array
 - O compilador o ignora, pois só se preocupa com o endereço do primeiro elemento

```
void funcao( int a[10], int num );
```

```
void funcao( int a[], int num );
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Os protótipos de função podem incluir nomes de parâmetro
 - Mas o compilador vai ignorá-los
 - Os nomes de parâmetro devem ser deixados fora dos protótipos de função

```
void funcao( int a[], int num );
```

```
void funcao( int [], int );
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Nono Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 9
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>

using namespace std;

void modifyArray(int [], int); // Protótipo de função
void modifyElement(int); // Protótipo de função

int main() {
    const int arraySize = 5;
    int a [arraySize] = {0, 1, 2, 3, 4};

    cout << " ** Passagem de array por referencia\n";
    cout << "Array original:\n";
    for (int i = 0; i < arraySize; i++)
        cout << setw(3) << a[i];
    cout << endl;

    modifyArray(a, arraySize);

    cout << "Array modificado:\n";
    for (int i = 0; i < arraySize; i++)
        cout << setw(3) << a[i];
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Nono Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 9
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>

using namespace std;

void modifyArray(int [], int); // Protótipo de função
void modifyElement(int); // Protótipo de função

int main() {
    const int arraySize = 5;
    int a [arraySize] = {0, 1, 2, 3, 4};

    cout << " ** Passagem de array por referencia\n";
    cout << "Array original:\n";
    for (int i = 0; i < arraySize; i++)
        cout << setw(3) << a[i];
    cout << endl;

    modifyArray(a, arraySize);

    cout << "Array modificado:\n";
    for (int i = 0; i < arraySize; i++)
        cout << setw(3) << a[i];
}
```

Função recebe um array como argumento

Passagem do array

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Nono Exemplo Usando Array em C++

```
cout << "\n\n ** Passagem de elemento por valor\n\n";
cout << "Elemento original:\n";
cout << setw(3) << a[3] << endl;

modifyElement(a[3]);

cout << "Elemento modificado:\n";
cout << setw(3) << a[3] << endl;

return 0;
}

void modifyArray(int b[], int numberElements) {
    // Multiplica cada elemento do array por dois
    for (int i = 0; i < numberElements; i++)
        b[i] *= 2;
}

void modifyElement(int e) {
    cout << "Valor do elemento em modifyElement: " << e * 2 << endl;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Nono Exemplo Usando Array em C++

```
cout << "\n\n ** Passagem de elemento por valor\n\n";
cout << "Elemento original:\n";
cout << setw(3) << a[3] << endl;
modifyElement(a[3]);
cout << "Elemento modificado:\n";
cout << setw(3) << a[3] << endl;

return 0;
}

void modifyArray(int b[], int numberElements) {
    // Multiplica cada elemento do array por dois
    for (int i = 0; i < numberElements; i++)
        b[i] *= 2;
}

void modifyElement(int e) {
    cout << "Valor do elemento em modifyElement: " << (e * 2) << endl;
}
```

Passagem de elemento

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Nono Exemplo Usando Array em C++

```
cout << "\n\n ** Passagem de elemento por valor\n\n";

shell$ g++ exemplo.cpp -o ex9
shell$ ./ex9
** Passagem de array por referencia
Array original:
0 1 2 3 4
Array modificado:
0 2 4 6 8

** Passagem de elemento por valor
Elemento original:
6
Valor do elemento em modifyElement: 12
Elemento modificado:
6
shell$
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Parâmetros de array `const`
 - Qualificador `const`
 - Evita que valores do array sejam alterados no chamador por códigos na função chamada
 - Os elementos no array são constantes no corpo da função
 - Permite que o programador evite alterações acidentais nos dados

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Passagem de Array para Função

- Parâmetros de array `const`
 - Qualificador `const`
 - Evita que valores do array sejam alterados no chamador por códigos na função chamada
 - Os elementos no array são constantes no corpo da função
 - Permite que o programador evite alterações acidentais nos dados



Como os arrays são passados por referência, é comum utilizar o qualificador `const` para evitar alterações

Décimo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 10
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

void tryToModify(const int i[]):

int main() {
    int a[] = {10, 20, 30};

    tryToModify(a);

    cout << a[0] << " " << a[1] << " " << a[2] << endl;

    return 0;
}

void tryToModify(const int a[]) {
    a[0] /= 2;
    a[1] /= 2;
    a[2] /= 2;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 10
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

void tryToModify(const int a[]);

int main() {
    int a[] = {10, 20, 30};
    tryToModify(a);

    cout << a[0] << ' ' << a[1] << ' ' << a[2] << endl;

    return 0;
}

void tryToModify(const int a[]) {
    a[0] /= 2;
    a[1] /= 2;
    a[2] /= 2;
}

```

Use do const evita que a função altere o array

O array só é const dentro da função

O array não pode ser modificado dentro do corpo da função

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 10
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

shell> g++ exemplo.cpp -o ex10
Erro!
shell>

    tryToModify(a);

    cout << a[0] << ' ' << a[1] << ' ' << a[2] << endl;

    return 0;
}

void tryToModify(const int a[]) {
    a[0] /= 2;
    a[1] /= 2;
    a[2] /= 2;
}

```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Estudo de Caso: Classe GradeBook

- Classe GradeBook
 - Representa um livro de notas que armazena e analisa as notas
 - Agora pode armazenar notas em um array
- Membros de dados static
 - Variáveis das quais os objetos de uma classe não têm uma cópia separada
 - Uma única cópia é compartilhada por todos os objetos da classe
 - Podem ser acessados mesmo sem objetos da classe
 - Use o nome da classe seguido do operador binário de resolução de escopo e o nome dos membros de dados static

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```

/*
 * Aula 7 -- Exemplo 11
 * Arquivo: GradeBookCap7Ex11.h
 * Autor: Miguel Campista
 */
#include <string>
#include <iomanip>

using namespace std;

// Definição da classe GradeBook
class GradeBook {
public:
    // Constante - Número de alunos que fizeram o teste
    const static int students = 10;
    // Construtor inicializa courseName com a string-argumento
    GradeBook(string, const int i);
    // Função que configura o nome do curso
    void setCourseName(string);
    // Função que devê o nome do curso
    string getCourseName();
    // Função para dar entrada nos conceitos dos alunos
    double getAverage();
    // Função para exibir os conceitos
    void displayGrades();
    // Função para retornar a menor nota dos alunos
    int getMinimum();
    // Função para retornar a maior nota dos alunos
    int getMaximum();
};

```

Décimo Primeiro Exemplo Usando Array em C++

```

/*
 * Aula 7 -- Exemplo 11
 * Arquivo: GradeBookCap7Ex11.h
 * Autor: Miguel Campista
 */
#include <string>
#include <iomanip>

using namespace std;

// Definição de classe GradeBook
class GradeBook {
public:
    // Constante - Número de alunos que fizeram o teste
    const static int students = 10;
    // Construtor inicializa courseName com a string-argumento
    GradeBook(string, const int i);
    // Função que configura o nome do curso
    void setCourseName(string);
    // Função que obtém o nome do curso
    string getCourseName();
    // Função para dar entrada nos conceitos dos alunos
    double getAverage();
    // Função para exibir os conceitos
    void displayGrades();
    // Função para retornar a menor nota dos alunos
    int getMinimum();
    // Função para retornar a maior nota dos alunos
    int getMaximum();
};

```

students é uma variável da classe static

Décimo Primeiro Exemplo Usando Array em C++

```

// Função para realizar funções nas notas
void processGrades();
// Função para exibir e distribuição de notas de turma
void displayBarChart();
void displayMessage();

private:
    string courseName;
    int grades[students];
};

```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

array grades para armazenar as notas

```
// Função para realizar funções nas notas
void processGrades();
// Função para exibir a distribuição de notas da turma
void displayBarChart();
void displayMessage();

private:
    string courseName;
    int grades[students];
};
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 11
 * Arquivo: GradeBookCap7Ex11.cpp
 * Autor: Miguel Campista
 */

#include <iostream>
#include <string>
#include "GradeBookCap7Ex11.h"

// Construtor inicializa courseName com o string-argumento
GradeBook::GradeBook(string name, const int gradesArray[]) {
    setCourseName(name); // Chama a função set para inicialização

    for (int i = 0; i < students; i++)
        grades[i] = gradesArray[i];
}

// Função para retornar a menor nota dos alunos
int GradeBook::getMinimum() {
    int minimum = 100;
    for (int i = 0; i < students; i++) {
        if (grades[i] < minimum)
            minimum = grades[i];
    }
    return minimum;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

Copia elementos de gradesArray para o atributo grades

```
/*
 * Aula 7 -- Exemplo 11
 * Arquivo: GradeBookCap7Ex11.cpp
 * Autor: Miguel Campista
 */

#include <iostream>
#include <string>
#include "GradeBookCap7Ex11.h"

// Construtor inicializa courseName com o string-argumento
GradeBook::GradeBook(string name, const int gradesArray[]) {
    setCourseName(name); // Chama a função set para inicialização

    for (int i = 0; i < students; i++)
        grades[i] = gradesArray[i];
}

// Função para retornar a menor nota dos alunos
int GradeBook::getMinimum() {
    int minimum = 100;
    for (int i = 0; i < students; i++) {
        if (grades[i] < minimum)
            minimum = grades[i];
    }
    return minimum;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum() {
    int maximum = 0;
    for (int i = 0; i < students; i++) {
        if (grades[i] > maximum)
            maximum = grades[i];
    }
    return maximum;
}

// Função que calcula a média das notas da turma
double GradeBook::getAverage() {
    int total = 0;
    for (int i = 0; i < students; i++)
        total += grades[i];
    return static_cast<double>(total)/students;
}

// Função que configura o nome do curso
void GradeBook::setCourseName(string name) {
    if (name.length() <= 25) {
        courseName = name;
    } else {
        courseName = name.substr(0, 25);
        cout << "Warning: Nome \'" << name <<
            "\' excede o limite máximo de 25 caracteres..." << endl <<
            "Nome limitado aos primeiros 25 caracteres!" << courseName <<
            endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

Loop em grades para o encontrar a nota máxima

```
// Função para retornar a menor nota dos alunos
int GradeBook::getMinimum() {
    int minimum = 100;
    for (int i = 0; i < students; i++) {
        if (grades[i] < minimum)
            minimum = grades[i];
    }
    return minimum;
}

// Função que calcula a média das notas da turma
double GradeBook::getAverage() {
    int total = 0;
    for (int i = 0; i < students; i++)
        total += grades[i];
    return static_cast<double>(total)/students;
}

// Função que configura o nome do curso
void GradeBook::setCourseName(string name) {
    if (name.length() <= 25) {
        courseName = name;
    } else {
        courseName = name.substr(0, 25);
        cout << "Warning: Nome \'" << name <<
            "\' excede o limite máximo de 25 caracteres..." << endl <<
            "Nome limitado aos primeiros 25 caracteres!" << courseName <<
            endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
// Função que obtém o nome do curso
string GradeBook::getCourseName() {
    return courseName;
}

void GradeBook::displayMessage() {
    cout << "Bem-vindo ao seu primeiro programa com classes em C++" << endl;
    cout << getCourseName() << "!" << endl;
}

// Função para realizar funções nas notas
void GradeBook::processGrades() {
    displayGrades();

    cout << "\nMédia da turma eh: " << setprecision(2) << fixed <<
        getAverage() << endl;
    cout << "Menor nota foi: " << getMinimum() << endl;
    cout << "Maior nota foi: " << getMaximum() << endl;
    cout << endl;
    displayBarChart();
}

// Função para exibir os conceitos
void GradeBook::displayGrades() {
    for (int i = 0; i < students; i++) {
        cout << "Student " << setw(2) << i + 1 << " " <<
            setw(3) << grades[i] << endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
// Função para exibir a distribuição de notas de turma
void GradeBook::displayBarChart() {
    cout << "Distribuição de notas: " << endl;

    const int frequencySize = 11;
    int frequency [frequencySize] = {};

    for (int i = 0; i < students; i++)
        frequency [grades [i]/10]++;

    for (int count = 0; count < frequencySize; count++) {
        if (count == 0)
            cout << " 0-9:";
        else if (count == 10)
            cout << " 100:";
        else
            cout << count * 10 << "-" << (count * 10) + 9 << ":";

        for (int stars = 0; stars < frequency [count]; stars++)
            cout << " *";

        cout << endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
// Função para exibir a distribuição de notas de turma
void GradeBook::displayBarChart() {
    cout << "Distribuição de notas: " << endl;

    const int frequencySize = 11;
    int frequency [frequencySize] = {};

    for (int i = 0; i < students; i++)
        frequency [grades [i]/10]++;

    for (int count = 0; count < frequencySize; count++)
        if (count == 0)
            cout << " 0-9:";
        else if (count == 10)
            cout << " 100:";
        else
            cout << count * 10 << "-" << (count * 10) + 9 << ":";

        for (int stars = 0; stars < frequency [count]; stars++)
            cout << " *";

        cout << endl;
    }
}
```

Loop em grades para o encontrar a frequência

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 11
 * Arquivo principal
 * Autor: Miguel Campista
 */

#include <iostream>
#include <string>
#include "GradeBookCap7Ex11.h" // Inclui a definição da classe

using namespace std;

int main() {
    // Array de notas dos alunos
    int gradesArray [GradeBook::students] = {87, 68, 94, 100, 85, 78, 85, 91, 76, 87};

    // Cria dois objetos GradeBook
    GradeBook gradeBook("Linguagens de Programação", gradesArray);

    // Exibe o valor inicial de courseName
    gradeBook.displayMessage();
    gradeBook.processGrades();

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 11
 * Arquivo principal
 * Autor: Miguel Campista
 */

#include <iostream>
#include <string>
#include "GradeBookCap7Ex11.h" // Inclui a definição da classe

using namespace std;

int main() {
    // Array de notas dos alunos
    int gradesArray [GradeBook::students] = {87, 68, 94, 100, 85, 78, 85, 91, 76, 87};

    // Cria dois objetos GradeBook
    GradeBook gradeBook("Linguagens de Programação", gradesArray);

    // Exibe o valor inicial de courseName
    gradeBook.displayMessage();
    gradeBook.processGrades();

    return 0;
}
```

Usa students declarado como static na classe

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Primeiro Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 11
 * Arquivo principal
 * Autor: Miguel Campista
 */

#include <iostream>
#include <string>
#include "GradeBookCap7Ex11.h" // Inclui a definição da classe

using namespace std;

int main() {
    // Exibe o valor inicial de courseName
    gradeBook.displayMessage();
    gradeBook.processGrades();

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Busca em Array

- Os arrays podem armazenar grande quantidade de dados
 - Talvez seja necessário determinar se existe um certo valor-chave no array
- Busca linear
 - Compara cada elemento de um array com uma chave de pesquisa

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Busca Linear em Array

- Caso os elementos do array não estejam organizados segundo alguma classificação
 - Para encontrar um elemento, é igualmente provável que o valor seja encontrado tanto na primeira quanto na última posição
 - Em média, o programa deve comparar a chave de pesquisa com metade dos elementos no array
 - Para determinar se o valor não se encontra no array
 - O programa deve comparar a chave de pesquisa com todos os elementos no array
- Busca linear é uma solução eficiente em arrays menores ou arrays não classificados

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Segundo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 12
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

int linearSearch(const int [], int, int);

int main() {
    const int arraySize = 100;
    int a [arraySize];
    int searchKey;

    for (int i = 0; i < arraySize; i++)
        a [i] = 2 * i;

    cout << "Entre com o elemento a ser buscado: ";
    cin >> searchKey;

    int element = linearSearch(a, searchKey, arraySize);

    if (element != -1)
        cout << "Elemento encontrado na posicao: " << element << endl;
    else
        cout << "Elemento nao encontrado." << endl;

    return 0;
}
```

Décimo Segundo Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 12
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

int linearSearch(const int [], int, int);

int main() {
    const int arraySize = 100;
    int a [arraySize];
    int searchKey;

    for (int i = 0; i < arraySize; i++)
        a [i] = 2 * i;

    cout << "Entre com o elemento a ser buscado: ";
    cin >> searchKey;

    int element = linearSearch(a, searchKey, arraySize);

    if (element != -1)
        cout << "Elemento encontrado na posicao: " << element << endl;
    else
        cout << "Elemento nao encontrado." << endl;

    return 0;
}
```

A função recebe o array, o valor chave e o tamanho do array como argumentos

A função retorna a a posição do elemento encontrado ou -1 se não encontrar

Décimo Segundo Exemplo Usando Array em C++

```
int linearSearch(const int array[], int key, int sizeOfArray) {
    for (int i = 0; i < sizeOfArray; i++) {
        if (array[i] == key)
            return i;
    }
    return -1;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Segundo Exemplo Usando Array em C++

```
int linearSearch(const int array[], int key, int sizeOfArray) {
    for (int i = 0; i < sizeOfArray; i++) {
        if (array[i] == key)
            return i;
    }
    return -1;
}
```

Pesquisa em todo o array

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Segundo Exemplo Usando Array em C++

```
int linearSearch(const int array[], int key, int sizeOfArray) {
    for (int i = 0; i < sizeOfArray; i++) {
        if (array[i] == key)
            return i;
    }
    return -1;
}
```

```
shell->$ g++ exemplo.cpp -o ex12
shell->./ex1
Entre com o elemento a ser buscado: 4
Elemento encontrado na posicao: 2
shell->$
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Classificação de Arrays por Inserção

- Classificando dados
 - Ordenamento de elementos de um array
 - Uma das aplicações mais importantes da computação
 - **Praticamente toda organização tem de classificar algum tipo de dado**

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Classificação de Arrays por Inserção

- Classificação por inserção
 - Simples, mas ineficaz
 - A primeira iteração pega o segundo elemento
 - **Se for menor que o primeiro elemento, troca-o por este**
 - A segunda iteração examina o terceiro elemento
 - **Insera-o na posição correta com relação aos dois primeiros elementos**
 - ...
 - Na enésima iteração desse algoritmo, os primeiros i elementos no array original serão classificados

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Terceiro Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 13
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>

using namespace std;

int *insertionSort(int [], int);

int main() {
    const int arraySize = 10;
    int unsorted [arraySize] = {34, 56, 4, 10, 77, 51, 99, 30, 5, 52};
    int *sorted;

    cout << "Unsorted array:" << endl;
    for (int i = 0; i < arraySize; i++)
        cout << setw(4) << unsorted [i];
    cout << endl;

    sorted = insertionSort(unsorted, arraySize);
    cout << "Unsorted array:" << endl;
    for (int i = 0; i < arraySize; i++)
        cout << setw(4) << sorted [i];
    cout << endl;

    return 0;
}
```

Décimo Terceiro Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 13
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>

using namespace std;

int *insertionSort(int [], int);

int main() {
    const int arraySize = 10;
    int unsorted [arraySize] = {34, 56, 4, 10, 77, 51, 99, 30, 5, 52};
    int *sorted;

    cout << "Unsorted array:" << endl;
    for (int i = 0; i < arraySize; i++)
        cout << setw(4) << unsorted [i];
    cout << endl;

    sorted = insertionSort(unsorted, arraySize);
    cout << "Unsorted array:" << endl;
    for (int i = 0; i < arraySize; i++)
        cout << setw(4) << sorted [i];
    cout << endl;

    return 0;
}
```

Função que retorna a referência para o array ordenado

Décimo Terceiro Exemplo Usando Array em C++

```
int *insertionSort(int array [], int sizeOfArray) {
    int insert;
    for (int next = 1; next < sizeOfArray; next++) {
        insert = array [next]; // Armazena o valor do elemento atual
        int moveItem = next; // Inicializa a localização para colocar o elemento

        while ((moveItem > 0) && (array [moveItem - 1] > insert)) {
            array [moveItem] = array [moveItem - 1];
            moveItem--;
        }
        array [moveItem] = insert;
    }
    return array;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Terceiro Exemplo Usando Array em C++

```
int *insertionSort(int array [], int sizeOfArray) {
    int insert;
    for (int next = 1; next < sizeOfArray; next++) {
        insert = array [next]; // Armazena o valor do elemento atual
        int moveItem = next; // Inicializa a localização para colocar o elemento

        while ((moveItem > 0) && (array [moveItem - 1] > insert)) {
            array [moveItem] = array [moveItem - 1];
            moveItem--;
        }
        array [moveItem] = insert;
    }
    return array;
}
```

Para cada elemento do array, a localização segundo a ordenação deve ser encontrada e o elemento deve ser posicionado nessa posição

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Terceiro Exemplo Usando Array em C++

```
shell-$ g++ exemplo.cpp -o ex13
shell-$ ./ex13
Unsorted array:
34 56 4 10 77 51 93 30 5 52

Sorted array:
4 5 10 30 34 51 52 56 77 93
shell-$
```

```
array [moveItem] = insert;
}
return array;
}
```

Décimo Terceiro Exemplo Usando Array em C++

```
shell-$ g++ exemplo.cpp -o ex13
shell-$ ./ex13
Unsorted array:
34 56 4 10 77 51 93 30 5 52

Sorted array:
4 5 10 30 34 51 52 56 77 93
shell-$
```

```
array [moveItem] = insert;
}
return array;
}
```

Foi necessário a criação de um array ordenado?

Décimo Terceiro Exemplo Usando Array em C++

```
/* Aula 7 - Exemplo 13
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>
```

```
using namespace std;
```

```
int insertionSort(int [], int);
```

```
int main() {
    const int arraySize = 10;
    int unsorted [arraySize] = {34, 56, 4, 10, 77, 51, 93, 30, 5, 52};
    int *sorted;
```

```
    cout << "Unsorted array:" << endl;
```

```
    for (int i = 0; i < arraySize; i++)
```

```
        cout << setw(4) << unsorted [i];
```

```
    cout << endl;
```

```
    sorted = insertionSort(unsorted, arraySize);
```

```
    cout << "Unsorted array:" << endl;
```

```
    for (int i = 0; i < arraySize; i++)
```

```
        cout << setw(4) << sorted [i];
```

```
    cout << endl;
```

```
    return 0;
}
```

Aqui poderia ser o próprio array unsorted porque a passagem de parâmetro é por referência

Décimo Terceiro Exemplo Usando Array em C++

```
/* Aula 7 - Exemplo 13
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>
```

```
using namespace std;
```

```
int insertionSort(int [], int);
```

```
int main() {
    const int arraySize = 10;
    int unsorted [arraySize] = {34, 56, 4, 10, 77, 51, 93, 30, 5, 52};
    int *sorted;
```

```
    cout << "Unsorted array:" << endl;
```

```
    for (int i = 0; i < arraySize; i++)
```

```
        cout << setw(4) << unsorted [i];
```

```
    cout << endl;
```

```
    sorted = insertionSort(unsorted, arraySize);
```

```
    cout << endl;
```

```
    return 0;
}
```

Aqui poderia ser o próprio array unsorted porque a passagem de parâmetro é por referência

Como ficaria usando templates?

Décimo Terceiro Exemplo Usando Array em C++

```
/* Aula 7 - Exemplo 13 de template-sort.h
 * Autor: Miguel Campista
 */
#include <iostream>
#include <omanip>
#include "template-sort.h"
```

```
using namespace std;
```

```
int main() {
    const int arraySize = 10;
    int unsorted [arraySize] = {34, 56, 4, 10, 77, 51, 93, 30, 5, 52};
    int *sorted;
```

```
    cout << "Unsorted array:" << endl;
```

```
    for (int i = 0; i < arraySize; i++)
```

```
        cout << setw(4) << unsorted [i];
```

```
    cout << endl;
```

```
    sorted = insertionSort(unsorted, arraySize);
```

```
    cout << "Unsorted array:" << endl;
```

```
    for (int i = 0; i < arraySize; i++)
```

```
        cout << setw(4) << sorted [i];
```

```
    cout << endl;
```

```
    return 0;
}
```

Décimo Terceiro Exemplo Usando Array em C++

```
template <class X>
X insertionSort(X array [], int sizeOfArray) {
    X insert;
    for (int next = 1; next < sizeOfArray; next++) {
        insert = array [next]; // Armazena o valor do elemento atual
        int moveItem = next; // Inicializa a localização para colocar o elemento

        while (moveItem > 0 && (array [moveItem - 1] > insert)) {
            array [moveItem] = array [moveItem - 1];
            moveItem--;
        }
        array [moveItem] = insert;
    }
    return array;
}
```

Arrays Multidimensionais

- Arrays multidimensionais com duas dimensões
 - Denominados arrays bidimensionais ou arrays 2-D.
 - Representam tabelas de valores com linhas e colunas.
 - Os elementos são referenciados com dois subscritos ($[x][y]$).
 - Em geral, um array com m linhas e n colunas é chamado de array m por n .
- Os arrays multidimensionais podem ter mais de duas dimensões

Arrays Multidimensionais

- Declarando e inicializando arrays bidimensionais
 - Declarando um array bidimensional b
 - `int b[2][2] = { { 1, 2 }, { 3, 4 } };`
 - 1 e 2 inicializam `b[0][0]` e `b[0][1]`
 - 3 e 4 inicializam `b[1][0]` e `b[1][1]`
 - `int b[2][2] = { { 1 }, { 3, 4 } };`
 - A linha 0 contém valores 1 e 0 (implicitamente inicializados em zero).
 - A linha 1 contém os valores 3 e 4

Arrays Multidimensionais

	Coluna 1	Coluna 2	Coluna 3	Coluna 4
Linha 1	a [0][0]	a [0][1]	a [0][2]	a [0][3]
Linha 2	a [1][0]	a [1][1]	a [1][2]	a [1][3]
Linha 3	a [2][0]	a [2][1]	a [2][2]	a [2][3]

Décimo Quarto Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 14
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

void printArray (const int i[3]);

int main() {
    int array1 [2][3] = {{1, 2, 3}, {4, 5, 6}};
    int array2 [2][3] = {1, 2, 3, 4, 5};
    int array3 [2][3] = {{1, 2}, {4}};

    cout << "Os valores do array1 por linha sao: " << endl;
    printArray(array1);

    cout << "Os valores do array2 por linha sao: " << endl;
    printArray(array2);

    cout << "Os valores do array3 por linha sao: " << endl;
    printArray(array3);

    return 0;
}
```

Décimo Quarto Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 14
 * Autor: Miguel Campista
 */
#include <iostream>

using namespace std;

void printArray (const int i[3]);

int main() {
    int array1 [2][3] = {{1, 2, 3}, {4, 5, 6}};
    int array2 [2][3] = {1, 2, 3, 4, 5};
    int array3 [2][3] = {{1, 2}, {4}};

    cout << "Os valores do array1 por linha sao: " << endl;
    printArray(array1);

    cout << "Os valores do array2 por linha sao: " << endl;
    printArray(array2);

    cout << "Os valores do array3 por linha sao: " << endl;
    printArray(array3);

    return 0;
}
```

Uso de inicializadores de array

Décimo Quarto Exemplo Usando Array em C++

```
void printArray(const int a [][3]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++)
            cout << a [i][j] << ' ';
        cout << endl;
    }
}
```

Décimo Quarto Exemplo Usando Array em C++

```
void printArray(const int a [][3]) {  
    for (int i = 0; i < 2; i++) {  
        for (int j = 0; j < 3; j++)  
            cout << a [i][j] << ' ';  
        cout << endl;  
    }  
}
```

Uso de loops aninhados para imprimir um array

Décimo Quarto Exemplo Usando Array em C++

```
void printArray(const int a [][3]) {  
    for (int i = 0; i < 2; i++) {  
        for (int j = 0; j < 3; j++)  
            cout << a [i][j] << ' ';  
        cout << endl;  
    }  
}
```

```
shell-> g++ exemplo.cpp -o ex14
```

```
shell-> ./ex14
```

Os valores do array1 por linha são:

```
1 2 3
```

```
4 5 6
```

Os valores do array2 por linha são:

```
1 2 3
```

```
4 5 0
```

Os valores do array3 por linha são:

```
1 2 0
```

```
4 0 0
```

```
shell->
```

Arrays Multidimensionais

- Parâmetros de array multidimensional
 - O tamanho da primeira dimensão não é necessário
 - É igual ao array unidimensional
 - O tamanho das dimensões subsequentes é necessário
 - O compilador tem que saber quantos elementos deve pular para mover-se para o segundo elemento na primeira dimensão
- Ex.: `void printArray(const int a[][3]);`
 - A função pulará 3 elementos da linha 0 para acessar os elementos da linha 1 (`a[1][x]`)

Arrays Multidimensionais

- Manipulações de array multidimensional
 - Comumente executadas com instruções for
 - Exemplo
 - Modificar todos os elementos em uma linha
 - `for (int col = 0; col < 4; col++)
 a [2][col] = 0;`
 - Exemplo
 - Total de todos os elementos
 - `total = 0;
for (row = 0; row < 3; row++)
 for (col = 0; col < 4; col++)
 total += a [row][col];`

Estudo de Caso: Classe GradeBook

- Classe GradeBook
 - Array unidimensional
 - Armazena as notas dos alunos em um único exame
 - Array bidimensional
 - Armazena várias notas de um único aluno e de vários alunos da classe como um todo.
 - Cada linha representa as notas de um único aluno
 - Cada coluna representa todas as notas tiradas pelos alunos em um determinado exame

Décimo Quinto Exemplo Usando Array em C++

```
/*  
 * Aula 7 -- Exemplo 18  
 * Arquivo: GradeBookDepEx18.h  
 * Autor: Miguel Campista  
 */  
#include <string>  
#include <iomanip>  
  
using namespace std;  
  
// Definição da classe GradeBook  
class GradeBook {  
  
public:  
    // Constante - Número de alunos que fizeram o teste  
    const static int estudantes = 10;  
    // Constante - Número de alunos que fizeram o teste  
    const static int teste = 3;  
    // Construtor inicializa oGradeBook com o string-argumento  
    GradeBook(string, const int [][teste]);  
    // Função que configura o nome do curso  
    void setCourseName(string);  
    // Função que obtém o nome do curso  
    string getCourseName();  
    // Função para obter entrada nos conectores dos alunos  
    double getAverage(const int [], const int);  
    // Função para mostrar os subtotais  
    void displayGrades();  
    // Função para retornar a menor nota dos alunos  
    int getMinimum();  
};
```

Décimo Quinto Exemplo Usando Array em C++

```

/*
 * Aula 7 -- Exemplo 15
 * Arquivo: GradeBookCap7Ex15.h
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include <omanip>

using namespace std;

// Definição da classe GradeBook
class GradeBook {
public:
    // Constante - Número de alunos que fizeram o teste
    const static int students = 10;
    // Constante - Número de testes que fizeram o teste
    const static int tests = 10;
    GradeBook(string, const int &[tests]);
    void setCourseName(string);
    // Função que recebe o nome do curso
    string getCourseName();
    // Função para dar entrada nos conceitos dos alunos
    double getAverage(const int i, const int);
    // Função para exibir os conceitos
    void displayGrades();
    // Função para retornar a menor nota dos alunos
    int getMinimum();
};
    
```

O construtor GradeBook recebe um array bidimensional

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum() {
    // Função para realizar funções nas notas
    void processGrades();
    // Função para exibir a distribuição de notas da turma
    void displayBarChart();
    void displayMessage();
}

private:
    string courseName;
    int grades [students][tests];
};
    
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum();
// Função para realizar funções nas notas
void processGrades();
// Função para exibir a distribuição de notas da turma
void displayBarChart();
void displayMessage();

private:
    string courseName;
    int grades [students][tests];
};
    
```

Declara um array bidimensional para grades

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

/*
 * Aula 7 -- Exemplo 15
 * Arquivo: GradeBookCap7Ex15.cpp
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include "GradeBookCap7Ex15.h"

// Construtor inicializa courseName com a string-argumento
GradeBook::GradeBook(string name, const int gradesArray[][tests]) {
    setCourseName(name); // Chama a função set para inicialização

    for (int i = 0; i < students; i++) {
        for (int test = 0; test < tests; test++)
            grades [i][test] = gradesArray[i][test];
    }

    // Função para retornar a menor nota dos alunos
    int GradeBook::getMinimum() {
        int minimum = 100;
        for (int i = 0; i < students; i++) {
            for (int test = 0; test < tests; test++) {
                if (grades [i][test] < minimum)
                    minimum = grades [i][test];
            }
        }
        return minimum;
    }
};
    
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum() {
    int maximum = 0;
    for (int i = 0; i < students; i++) {
        for (int test = 0; test < tests; test++)
            if (grades [i][test] > maximum)
                maximum = grades [i][test];
    }
    return maximum;
}

// Função que calcula a média das notas de turma
double GradeBook::getAverage(const int testOfGrades [], const int grades) {
    int total = 0;

    for (int i = 0; i < grades; i++)
        total += testOfGrades [i];

    return static_cast <double> (total)/grades;
}

// Função que configura o nome do curso
void GradeBook::setCourseName(string name) {
    if (name.length () <= 25) {
        courseName = name;
    } else {
        courseName = name.substr (0, 25);
        cout << "Warning: Nome 'V' << name <<
        "\n" excede o limite máximo de 25 caracteres..." << endl <<
        "Nome limitado aos primeiros 25 caracteres!" << courseName <<
        endl;
    }
};
    
```

Usa loops aninhados para copiar elementos de gradesArray para grades

Loop nas linhas e nas colunas de grades para encontrar a nota mínima

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum() {
    int maximum = 0;
    for (int i = 0; i < students; i++) {
        for (int test = 0; test < tests; test++) {
            if (grades [i][test] > maximum)
                maximum = grades [i][test];
        }
    }
    return maximum;
}

// Função que calcula a média das notas de turma
double GradeBook::getAverage(const int testOfGrades [], const int grades) {
    int total = 0;

    for (int i = 0; i < grades; i++)
        total += testOfGrades [i];

    return static_cast <double> (total)/grades;
}

// Função que configura o nome do curso
void GradeBook::setCourseName(string name) {
    if (name.length () <= 25) {
        courseName = name;
    } else {
        courseName = name.substr (0, 25);
        cout << "Warning: Nome 'V' << name <<
        "\n" excede o limite máximo de 25 caracteres..." << endl <<
        "Nome limitado aos primeiros 25 caracteres!" << courseName <<
        endl;
    }
};
    
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```
// Função para retornar a maior nota dos alunos
int GradeBook::getMaximum() {
    int maximum = 0;
    for (int i = 0; i < estudantes; i++) {
        for (int test = 0; test < testes; test++)
            if (grades[i][test] > maximum)
                maximum = grades[i][test];
    }
    return maximum;
}

// Função que calcula a média das notas de turma
double GradeBook::getAverage(const int setOfGrades [], const int grades) {
    int total = 0;
    for (int i = 0; i < grades; i++)
        total += setOfGrades[i];
    return static_cast<double>(total)/grades;
}

// Função que configura o nome do curso
void GradeBook::setCourseName(string name) {
    if (name.length() <= 25) {
        courseName = name;
    } else {
        courseName = name.substr(0, 25);
        cout << "Warning: Nome << name <<
            "\n excede o limite máximo de 25 caracteres..." << endl <<
            "Nome limitado aos primeiros 25 caracteres!" << courseName <<
            endl;
    }
}
```

Loop nas linhas e nas colunas de grades para encontrar a nota máxima

Loop para calcular a média de um dos testes

Décimo Quinto Exemplo Usando Array em C++

```
// Função que obtém o nome do curso
string GradeBook::getCourseName() {
    return courseName;
}

void GradeBook::displayMessage() {
    cout << "Bem-vindo ao seu primeiro programa com classes em "
        << getCourseName() << "!" << endl;
}

// Função para realizar funções nas notas
void GradeBook::processGrades() {
    displayGrades();

    cout << "Média nota foi: " << getMinimum()
        << "\nMaior nota foi: " << getMaximum() << endl;
    cout << endl;
    displayBarChart();
}

// Função para realizar os conceitos
void GradeBook::displayBarChart() {
    for (int i = 0; i < estudantes; i++) {
        cout << "Student " << setv(i) << i + 1;

        for (int test = 0; test < testes; test++) {
            cout << setv(i) << grades[i][test];
        }

        double average = getAverage(grades[i], testes);
        cout << setv(i) << setprecision(2) << fixed << average << endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```
// Função para exibir a distribuição de notas da turma
void GradeBook::displayBarChart() {
    cout << "Distribuição de notas: " << endl;

    const int frequencySize = 11;
    int frequency[frequencySize] = {};

    for (int i = 0; i < estudantes; i++)
        for (int test = 0; test < testes; test++)
            frequency[grades[i][test]/10]++;

    for (int count = 0; count < frequencySize; count++) {
        if (count == 0)
            cout << " 0-9";
        else if (count == 10)
            cout << " 100";
        else
            cout << count * 10 << "-" << (count * 10) + 9 << " ";

        for (int stars = 0; stars < frequency[count]; stars++)
            cout << "*";

        cout << endl;
    }
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```
// Função para exibir a distribuição de notas da turma
void GradeBook::displayBarChart() {
    cout << "Distribuição de notas: " << endl;

    const int frequencySize = 11;
    int frequency[frequencySize] = {};

    for (int i = 0; i < estudantes; i++) {
        for (int test = 0; test < testes; test++)
            frequency[grades[i][test]/10]++;
    }

    for (int count = 0; count < frequencySize; count++)
        if (count == 0)
            cout << " 0-9";
        else if (count == 10)
            cout << " 100";
        else
            cout << count * 10 << "-" << (count * 10) + 9 << " ";

        for (int stars = 0; stars < frequency[count]; stars++)
            cout << "*";

        cout << endl;
    }
}
```

Loop nas linhas e nas colunas de grades para calcular a distribuição das notas

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 18
 * Arquivo principal
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include "GradeBookCap7Ex15.h" // Inclui a definição de classe

using namespace std;

int main() {
    // Array de notas dos alunos
    int gradeArray[GradeBook::estudantes][GradeBook::testes] =
        {{87, 96, 70}, {89, 87, 90},
         {84, 100, 90}, {100, 81, 82},
         {83, 68, 85}, {78, 87, 69},
         {85, 79, 83}, {85, 94, 100},
         {76, 72, 84}, {87, 93, 73}};

    // Cria dois objetos GradeBook
    GradeBook gradeBook("Linguagens de Programação", gradeArray);

    // Exibe o valor inicial de courseName
    gradeBook.displayMessage();
    gradeBook.processGrades();

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```
/*
 * Aula 7 -- Exemplo 18
 * Arquivo principal
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include "GradeBookCap7Ex15.h" // Inclui a definição de classe

using namespace std;

int main() {
    int gradeArray[GradeBook::estudantes][GradeBook::testes] =
        {{87, 96, 70}, {89, 87, 90},
         {84, 100, 90}, {100, 81, 82},
         {83, 68, 85}, {78, 87, 69},
         {85, 79, 83}, {85, 94, 100},
         {76, 72, 84}, {87, 93, 73}};

    // Cria dois objetos GradeBook
    GradeBook gradeBook("Linguagens de Programação", gradeArray);

    // Exibe o valor inicial de courseName
    gradeBook.displayMessage();
    gradeBook.processGrades();

    return 0;
}
```

Declaração do array 3 x 10

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Quinto Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 15
 * Arquivo principal
 * Autor: Miguel Campista
 */
C:\Users\Miguel\Documents\UFRJ\disciplinas\linguagens\projetos\aula7-ex15.exe
Bem-vindo ao seu primeiro programa com classes em Linguagens de Programacao!
Student 1: 87 94 70 84.33
Student 2: 68 87 90 81.67
Student 3: 74 100 90 91.67
Student 4: 100 81 82 87.67
Student 5: 83 65 85 79.33
Student 6: 70 87 85 84.33
Student 7: 85 75 83 81.00
Student 8: 71 84 100 85.00
Student 9: 76 72 84 79.33
Student 10: 87 93 75 84.33
Menor nota foi: 65
Maior nota foi: 100
Distribuição de notas:
0-9:
10-19:
20-29:
30-39:
40-49:
50-59:
60-69:
70-79:
80-89:
90-99:
100:

```

Introdução ao Template vector da C++ Standard Library

- Arrays baseados em ponteiro ao estilo do C
 - Apresentam alta probabilidade de erros e várias deficiências
 - O C++ não verifica se os subscritos são colocados fora do intervalo do array
 - Dois arrays não podem ser comparados de modo significativo com operadores de igualdade ou relacionais
 - Um array não pode ser atribuído a outro que esteja usando os operadores de atribuição

```

int a[10], b[10];
if (a == b) {
    ...
}

```

X Erro!

```

int a[10], b[10];
int b[10] = a;

```

X Erro!

Introdução ao Template vector da C++ Standard Library

- Template de classe vector
 - Disponível para construção de aplicativos com o C++
 - Pode ser definido para armazenar qualquer tipo de dados
 - Especificado entre colchetes angulares em `vector<type>`
 - Todos os elementos em um vector são configurados em 0 por padrão
 - A função-membro `size` obtém o tamanho do array
 - Número de elementos como um valor do tipo `size_t`
 - Os objetos vector podem ser comparados por meio dos operadores de igualdade e relacionais
 - O operador de atribuição pode ser usado em vectors

Introdução ao Template vector da C++ Standard Library

- Elementos vector podem ser obtidos como um lvalue (valor à esquerda) não modificável ou um rvalue modificável
 - lvalue não modificável
 - Expressão que identifica um objeto na memória, mas não pode ser usada para modificar esse objeto
 - lvalue modificável
 - Expressão que identifica um objeto na memória, mas pode ser usada para modificar o objeto

Introdução ao Template vector da C++ Standard Library

- Função `at` de vector
 - Oferece acesso a elementos individuais
 - Verifica limites
 - Lança uma exceção quando um índice especificado é inválido
 - O acesso com colchetes não executa a verificação de limites

Décimo Sexto Exemplo Usando Array em C++

```

/*
 * Aula 7 - Exemplo 16
 * Autor: Miguel Campista
 */
#include <iostream>
#include <iomanip>
#include <vector>

using namespace std;

void outputVector(const vector<int> &v) // Exibe o vector
void inputVector(vector<int> &v) // Exibe o vector

int main() {
    vector<int> inteiros1(7); // vector de inteiros de 7 elementos
    vector<int> inteiros2(10); // vector de inteiros de 10 elementos

    // Imprime o tamanho de inteiros1 e conteúdo
    cout << "Tamanho do vector inteiros1 eh " << inteiros1.size()
         << "\nvector depois da inicializacao:" << endl;
    outputVector(inteiros1);

    // Imprime o tamanho de inteiros2 e conteúdo
    cout << "Tamanho do vector inteiros2 eh " << inteiros2.size()
         << "\nvector depois da inicializacao:" << endl;
    outputVector(inteiros2);

    // Insere e imprime inteiros1 e inteiros2
    cout << "\nEntre 17 inteiros:" << endl;
    inputVector(inteiros1);
    inputVector(inteiros2);
}

```

Décimo Sexto Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 16
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include <vector>

using namespace std;
```

```
void outputVector(const vector<int> &v); // Exibe o vetor
void inputVector(vector<int> &v); // Exibe o vetor
```

```
int main() {
    vector<int> inteiros1(7); // vetor de inteiros de 7 elementos
    vector<int> inteiros2(10); // vetor de inteiros de 10 elementos
```

```
    // Imprime o tamanho de inteiros1 e conteúdo
    cout << "Tamanho do vetor inteiros1 eh " << inteiros1.size()
         << "\nvetor depois da inicializacao" << endl;
    outputVector(inteiros1);
```

```
    // Imprime o tamanho de inteiros2 e conteúdo
    cout << "Tamanho do vetor inteiros2 eh " << inteiros2.size()
         << "\nvetor depois da inicializacao" << endl;
    outputVector(inteiros2);
```

```
    // Terece e imprime inteiros1 e inteiros2
    cout << "\nEntre 17 inteiros:" << endl;
    inputVector(inteiros1);
    inputVector(inteiros2);
```

Uso do const evita que o array recebido seja alterado

Vectors que armazenam ints

Décimo Sexto Exemplo Usando Array em C++

```
/*
 * Aula 7 - Exemplo 16
 * Autor: Miguel Campista
 */
#include <iostream>
#include <string>
#include <vector>

using namespace std;
```

```
void outputVector(const vector<int> &v); // Exibe o vetor
void inputVector(vector<int> &v); // Exibe o vetor
```

```
int main() {
    vector<int> inteiros1(7); // vetor de inteiros de 7 elementos
    vector<int> inteiros2(10); // vetor de inteiros de 10 elementos
```

```
    // Imprime o tamanho de inteiros1 e conteúdo
    cout << "Tamanho do vetor inteiros1 eh " << inteiros1.size()
         << "\nvetor depois da inicializacao" << endl;
    outputVector(inteiros1);
```

```
    // Imprime o tamanho de inteiros2 e conteúdo
    cout << "Tamanho do vetor inteiros2 eh " << inteiros2.size()
         << "\nvetor depois da inicializacao" << endl;
    outputVector(inteiros2);
```

```
    // Mostra e imprime inteiros1 e inteiros2
    cout << "\nEntre 17 inteiros:" << endl;
    inputVector(inteiros1);
    inputVector(inteiros2);
```

Função size retorna o tamanho dos vectors

Décimo Sexto Exemplo Usando Array em C++

```
cout << "\nDepois de inputVector, os vetores contêm:\n"
     << "inteiros1" << endl;
outputVector(inteiros1);
cout << "inteiros2" << endl;
outputVector(inteiros2);
```

```
// Use operador de diferença (!=) com objetos vector
cout << "\nValidando: inteiros1 != inteiros2" << endl;
```

```
if (inteiros1 != inteiros2)
    cout << "inteiros1 e inteiros2 nao sao iguais" << endl;
```

```
// Criando vector inteiros3 usando inteiros1 como um
// inicializador; imprime tamanho e conteúdo
vector<int> inteiros3(inteiros1); // copia construtor
cout << "\nTamanho do vetor inteiros3 eh " << inteiros3.size()
     << "\nvetor depois da inicializacao" << endl;
outputVector(inteiros3);
```

```
// Use operador de atribuição (=) com objetos vector
cout << "\nAtribuindo inteiros1 aos inteiros3" << endl;
inteiros1 = inteiros2; // inteiros1 é maior que inteiros2
cout << "inteiros1:" << endl;
outputVector(inteiros1);
cout << "inteiros3:" << endl;
outputVector(inteiros3);
```

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
cout << "\nDepois de inputVector, os vetores contêm:\n"
     << "inteiros1" << endl;
outputVector(inteiros1);
cout << "inteiros2" << endl;
outputVector(inteiros2);
```

```
// Use operador de diferença (!=) com objetos vector
cout << "\nValidando: inteiros1 != inteiros2" << endl;
```

```
if (inteiros1 != inteiros2)
    cout << "inteiros1 e inteiros2 nao sao iguais" << endl;
```

```
// Criando vector inteiros3 usando inteiros1 como um
// inicializador; imprime tamanho e conteúdo
vector<int> inteiros3(inteiros1); // copia construtor
cout << "\nTamanho do vetor inteiros3 eh " << inteiros3.size()
     << "\nvetor depois da inicializacao" << endl;
outputVector(inteiros3);
```

```
// Use operador de atribuição (=) com objetos vector
cout << "\nAtribuindo inteiros1 aos inteiros3" << endl;
inteiros1 = inteiros2; // inteiros1 é maior que inteiros2
cout << "inteiros1:" << endl;
outputVector(inteiros1);
cout << "inteiros3:" << endl;
outputVector(inteiros3);
```

Comparação dos vectors com "!="

Inicialização de um vector com outro

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
cout << "\nDepois de inputVector, os vetores contêm:\n"
     << "inteiros1" << endl;
outputVector(inteiros1);
cout << "inteiros2" << endl;
outputVector(inteiros2);
```

```
// Use operador de diferença (!=) com objetos vector
cout << "\nValidando: inteiros1 != inteiros2" << endl;
```

```
if (inteiros1 != inteiros2)
    cout << "inteiros1 e inteiros2 nao sao iguais" << endl;
```

```
// Criando vector inteiros3 usando inteiros1 como um
// inicializador; imprime tamanho e conteúdo
vector<int> inteiros3(inteiros1); // copia construtor
cout << "\nTamanho do vetor inteiros3 eh " << inteiros3.size()
     << "\nvetor depois da inicializacao" << endl;
outputVector(inteiros3);
```

```
// Use operador de atribuição (=) com objetos vector
cout << "\nAtribuindo inteiros1 aos inteiros3" << endl;
inteiros1 = inteiros2; // inteiros1 é maior que inteiros2
cout << "inteiros1:" << endl;
outputVector(inteiros1);
cout << "inteiros3:" << endl;
outputVector(inteiros3);
```

Linguagens de Programação – DEL-Pol/UFRJ

Atribuição dos valores de um vector para outro

Décimo Sexto Exemplo Usando Array em C++

```
// Use operador de equação (==) com objetos vector
cout << "\nValidando: inteiros1 == inteiros2" << endl;

if (inteiros1 == inteiros2)
    cout << "inteiros1 e inteiros2 sao iguais" << endl;
```

```
// Use colchetes para criar e/ou
cout << "inteiros[5] is " << inteiros[5];
```

```
// Use colchetes para criar e/ou
cout << "\nAtribuindo 1000 ao inteiros[5]" << endl;
inteiros[5] = 1000;
cout << "inteiros1:" << endl;
outputVector(inteiros1);
```

```
system("PAUSE");
// Tentativa de usar indice fora do intervalo
cout << "\nTentativa de atribuir 1000 ao inteiros.at(15)" << endl;
inteiros.at(15) = 1000; // ERROR: fora do intervalo

return 0;
}
```

Linguagens de Programação – DEL-Pol/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
// Use operador de equação (==) com objetos vector
cout << "Validando: integer1 == integer2" << endl;
if (integer1 == integer2)
    cout << "integer1 e integer2 são iguais" << endl;

cout << "integer1[5] is " << integer1[5];

// Use colchetes para atribuir um valor ao elemento de um array
cout << "Inserindo 1000 ao integer1[5]" << endl;
integer1[5] = 1000;
cout << "integer1" << endl;
outputVector( integer1 );

system("PAUSE");
// Tentativa de usar indice fora do intervalo
cout << "Tentativa de atribuir 1000 ao integer1.at(15)" << endl;
integer1.at(15) = 1000; // ERROR: fora do intervalo
return 0;
}
```

Comparação dos
vectors com
"=="

Exibindo um
elemento de um
vector

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
// Use operador de equação (==) com objetos vector
cout << "Validando: integer1 == integer2" << endl;
if (integer1 == integer2)
    cout << "integer1 e integer2 são iguais" << endl;

// Use colchetes para criar e/ou atualizar um elemento de um array
cout << "Inserindo 1000 em integer1[5]" << endl;
integer1[5] = 1000;
cout << "integer1" << endl;
outputVector( integer1 );

system("PAUSE");
// Tentativa de usar indice fora do intervalo
cout << "Tentativa de atribuir 1000 ao integer1.at(15)" << endl;
integer1.at(15) = 1000; // ERROR: fora do intervalo
return 0;
}
```

Atualizando o
valor

Tentativa de
atualizar um valor
fora do intervalo

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
void outputVector(const vector<int> &array) {
    size_t i; // declaração de variável de controle
    for (i = 0; i < array.size(); i++) {
        cout << setw(12) << array[i];
        if ((i + 1) % 4 == 0) // 4 número por linha de saída
            cout << endl;
        if (i % 4 != 0)
            cout << endl;
    }
}

void inputVector(vector<int> &array) {
    for (size_t i = 0; i < array.size(); i++)
        cin >> array[i];
}
```

Exibe os
elementos do
array

Inserção de
elementos com o
cin

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Sexto Exemplo Usando Array em C++

```
void outputVector(const vector<int> &array) {
    size_t i; // declaração de variável de controle
    for (i = 0; i < array.size(); i++) {
        cout << setw(12) << array[i];
        if ((i + 1) % 4 == 0) // 4 número por linha de saída
            cout << endl;
        if (i % 4 != 0)
            cout << endl;
    }
}

void inputVector(vector<int> &array) {
    for (size_t i = 0; i < array.size(); i++)
        cin >> array[i];
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

```
C:\Users\Miguel\Documents\UFRJ\disciplinas\linguagens\projeto\aula7-ex1.exe
Tamanho do vector integer1 eh 7
vector depois da inicializacao:
0 0 0 0 0 0 0
Tamanho do vector integer2 eh 10
vector depois da inicializacao:
0 0 0 0 0 0 0 0 0 0
Iniciando integer1:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Depois de inputVector, os vectors contem:
integer1: 1 2 3 4
integer2: 0 0 10 11
12 13 14 15
16 17
Validando: integer1 != integer2
integer1 e integer2 nao sao iguais
Tamanho do vector integer3 eh 7
vector depois da inicializacao:
1 2 3 4 5 6 7
Inserindo integer2 aos integer1:
integer1: 0 9 10 11
12 13 14 15
integer2: 0 9 10 11
12 13 14 15
Validando: integer1 == integer2
integer1 e integer2 sao iguais
integer1[5] is 10
Inserindo 1000 ao integer1[5]
integer1: 0 9 10 11
12 1000 14 15
16 17
```

Introdução à classe STL array do C++11

- classe STL array
 - Disponível a partir do C++11
 - Oferece métodos para interação com a estrutura de dados
 - Assim como a classe vector
 - Porém, a memória é alocada com tamanho fixo
 - Não é possível aumentar ou diminuir o tamanho da memória alocada para o array após a sua criação
 - Diferente da classe vector

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Sétimo Exemplo Usando Array em C++11

```
#include <iostream>
#include <iomanip>
#include <array>

using namespace std;

int main () {
    array<int, 5> n;

    for (size_t i{0}; i < n.size (); i++) {
        n[i] = 0;
    }

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i{0}; i < n.size (); i++) {
        cout << setw (7) << i << setw(10) << n [i] << endl;
    }

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Sétimo Exemplo Usando Array em C++11

```
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14> g++ -Wall -std=c++11 aula7-ex17.cpp -o a
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14> ./a
Element      Value
0            0
1            0
2            0
3            0
4            0
```

```
using namespace std;

int main () {
    array<int, 5> n;

    for (size_t i{0}; i < n.size (); i++) {
        n[i] = 0;
    }

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i{0}; i < n.size (); i++) {
        cout << setw (7) << i << setw(10) << n [i] << endl;
    }

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Oitavo Exemplo Usando Array em C++11

```
#include <iostream>
#include <iomanip>
#include <array>

using namespace std;

int main () {
    array<int, 5> n {{32, 27, 64, 18, 95}};

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i{0}; i < n.size (); i++) {
        cout << setw (7) << i << setw(10) << n [i] << endl;
    }

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Oitavo Exemplo Usando Array em C++11

Inicialização agregada (múltiplos valores para inicialização de uma única estrutura) requer parênteses dentro de parênteses.

```
#include <iostream>
#include <iomanip>
#include <array>

using namespace std;

int main () {
    array<int, 5> n {{32, 27, 64, 18, 95}};

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i{0}; i < n.size (); i++) {
        cout << setw (7) << i << setw(10) << n [i] << endl;
    }

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Oitavo Exemplo Usando Array em C++11

```
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14> g++ -Wall -std=c++11 aula7-ex18.cpp -o a
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14> ./a
Element      Value
0            32
1            27
2            64
3            18
4            95
```

```
#include <iostream>
#include <iomanip>
#include <array>

using namespace std;

int main () {
    array<int, 5> n {{32, 27, 64, 18, 95}};

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i{0}; i < n.size (); i++) {
        cout << setw (7) << i << setw(10) << n [i] << endl;
    }

    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Geração de Números Aleatórios em C++11

- Função `rand` é considerada insegura
 - É possível conhecer a priori a sequência
- C++11 oferece:
 - Classes para geração de sequências pseudo-aleatórias
 - Classes para distribuição uniforme de números inteiros e outras distribuições

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Décimo Nono Exemplo Usando Array em C++11

```
#include <iostream>
#include <string>
#include <array>
#include <random>
#include <ctime>

using namespace std;

int main () {
    // Cria um objeto para geração de números pseudo-aleatórios
    default_random_engine engine (static_cast<unsigned int> (time (0)));
    // Cria um objeto para distribuição uniforme de inteiros
    uniform_int_distribution<unsigned int> randint (1, 6);

    array<unsigned int, 7> frequency {}; // Cria um array com 0s

    for (unsigned int i=0; i < 6000000; i++) {
        frequency [randint (engine)]++;
    }

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i=0; i < frequency.size (); i++) {
        cout << setw (7) << i << setw (10) << frequency [i] << endl;
    }

    return 0;
}
```

Décimo Nono Exemplo Usando Array em C++11

```
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14- g++ -Wall -std=c++11 aula7-ex19.cpp -o a
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14- ./a
Element Value
1 1001051
2 999236
3 1000288
4 999279
5 999248
6 1000798

int main () {
    // Cria um objeto para geração de números pseudo-aleatórios
    default_random_engine engine (static_cast<unsigned int> (time (0)));
    // Cria um objeto para distribuição uniforme de inteiros
    uniform_int_distribution<unsigned int> randint (1, 6);

    array<unsigned int, 7> frequency {}; // Cria um array com 0s

    for (unsigned int i=0; i < 6000000; i++) {
        frequency [randint (engine)]++;
    }

    cout << "Element" << setw (10) << "Value" << endl;

    for (size_t i=0; i < frequency.size (); i++) {
        cout << setw (7) << i << setw (10) << frequency [i] << endl;
    }

    return 0;
}
```

Range-based for no C++11

- Evita o uso de um contador para acessar os elementos do array...
 - Evita acesso a um elemento fora do intervalo
- Sintaxe:


```
//item recebe um elemento do array
for (tipo item : array)
//item recebe uma referência
for (tipo &item : array)
```
- Caso o índice seja necessário...
 - O range-based for não pode ser usado

Vigésimo Exemplo Usando Array em C++11

```
#include <iostream>
#include <array>

using namespace std;

int main () {
    array<int, 3> itens ({1, 2, 3});

    cout << "itens antes da modificação: ";
    for (int item : itens) {
        cout << item << endl;
    }

    cout << "multiplicação dos itens por 2..." << endl;
    for (int &itemRef : itens) {
        itemRef *= 2;
    }

    cout << "itens depois da modificação: ";
    for (int item : itens) {
        cout << item << endl;
    }

    return 0;
}
```

Vigésimo Exemplo Usando Array em C++11

```
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14- g++ -Wall -std=c++11 aula7-ex20.cpp -o a
~/disciplinas/linguagens/aulas/2017/programasC++11-C++14- ./a
itens antes da modificação: 1
2
3
multiplicação dos itens por 2...
itens depois da modificação: 2
4
6
```

```
cout << "itens antes da modificação: ";
for (int item : itens) {
    cout << item << endl;
}

cout << "multiplicação dos itens por 2..." << endl;
for (int &itemRef : itens) {
    itemRef *= 2;
}

cout << "itens depois da modificação: ";
for (int item : itens) {
    cout << item << endl;
}

return 0;
}
```

Vigésimo Exemplo Usando Array em C++11

E se fosse assim? Sem a referência...
O que seria impresso na tela?

```
int main () {
    array<int, 3> itens ({1, 2, 3});

    cout << "itens antes da modificação: ";
    for (int item : itens) {
        cout << item << endl;
    }

    cout << "multiplicação dos itens por 2..." << endl;
    for (int item : itens) {
        item = 2;
    }

    cout << "itens depois da modificação: ";
    for (int item : itens) {
        cout << item << endl;
    }

    return 0;
}
```

Vigésimo Exemplo Usando Array em C++11

```
1 #include <iostream>
2 #include <array>
3
4 int main() {
5     array<int, 3> itens({1, 2, 3});
6     cout << "itens antes da modificação: ";
7     for (int item : itens) {
8         cout << item << endl;
9     }
10    cout << "multiplicação dos itens por 2..." << endl;
11    for (int item : itens) {
12        item *= 2;
13    }
14    cout << "itens depois da modificação: ";
15    for (int item : itens) {
16        cout << item << endl;
17    }
18    return 0;
19 }
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Range-based for no C++11 usando auto

- Palavra-chave: **auto**
 - Requer que o compilador determine por inferência o tipo da variável
 - Baseado no valor usado para inicializar a variável

- Sintaxe:

```
for (auto item : array)
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Vigésimo Exemplo Usando Array em C++11

```
#include <iostream>
#include <array>

using namespace std;

int main() {
    array<int, 3> itens({1, 2, 3});
    cout << "itens antes da modificação: ";
    for (auto item : itens) {
        cout << item << endl;
    }
    cout << "multiplicação dos itens por 2..." << endl;
    for (auto &itemRef : itens) {
        itemRef *= 2;
    }
    cout << "itens depois da modificação: ";
    for (auto item : itens) {
        cout << item << endl;
    }
    return 0;
}
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Vigésimo Exemplo Usando Array em C++11

```
1 #include <iostream>
2 #include <array>
3
4 int main() {
5     array<int, 3> itens({1, 2, 3});
6     cout << "itens antes da modificação: ";
7     for (auto item : itens) {
8         cout << item << endl;
9     }
10    cout << "multiplicação dos itens por 2..." << endl;
11    for (auto &itemRef : itens) {
12        itemRef *= 2;
13    }
14    cout << "itens depois da modificação: ";
15    for (auto item : itens) {
16        cout << item << endl;
17    }
18    return 0;
19 }
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Exemplo: Ordenamento de Vetores

- Escreva um programa em C++ para ordenar uma sequência de inteiros utilizando o método do "insertion sort" e utilizando "vectors"

?

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Exemplo: Ordenamento de Vetores

```
1 /*
2  * Aula 7 - Exemplo 17 e/ vectors
3  * Autor: Miguel Campista
4  */
5 #include <iostream>
6 #include <vector>
7
8 using namespace std;
9
10 void insertionSort(vector<int> &v);
11
12 int main() {
13     const int arraySize = 10;
14     int a[] = {34, 56, 4, 10, 77, 51, 93, 30, 5, 82};
15
16     vector<int> unsorted(arraySize);
17     vector<int> sorted(arraySize);
18
19     for (int i = 0; i < arraySize; i++)
20         unsorted[i] = a[i];
21
22     cout << "Unsorted array:" << endl;
23
24     for (int i = 0; i < arraySize; i++)
25         cout << setv(4) << unsorted[i];
26     cout << endl;
27 }
```

Linguagens de Programação – DEL-Poli/UFRJ

Prof. Miguel Campista

Exemplo: Ordenamento de Vetores

```
insertionSort(unsorted);
sorted = unsorted;

cout << "\nSorted array:" << endl;

for (int i = 0; i < arraySize; i++)
    cout << setw(4) << sorted [i];
cout << endl;

return 0;

void insertionSort(vector<int> &array) {
    int insert;
    for (int next = 1; next < array.size(); next++) {
        insert = array [next]; // Armazena o valor do elemento atual
        int moveItem = next; // Inicializa o localizaçao para colocar o elemento

        while ((moveItem > 0) && (array [moveItem - 1] > insert)) {
            array [moveItem] = array [moveItem - 1];
            moveItem--;
        }
        array [moveItem] = insert;
    }
}
```

Leitura Recomendada

- Capítulos 7 do livro
 - Deitel, "C++ How to Program", 5th edition, Editora Prentice Hall, 2005