



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa



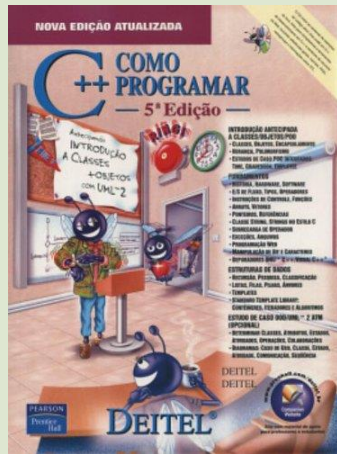
CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico

Projeto Olímpico de Programação

[C++] Apresentação

Livros

C Como Programar



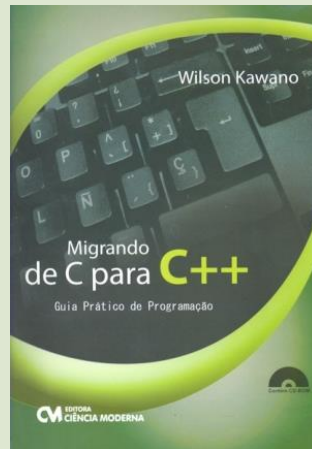
Deitel

Prentice Hall BR

2016

8576050560

Migrando de C para C++



Wilson

Ciência Moderna

2010

8573939478

Princípios e Práticas de Programação com C++



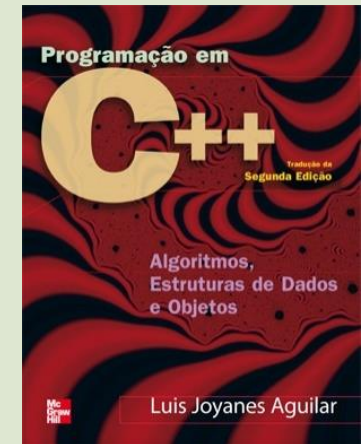
Stroustrup

Bookman

2011

8577809587

Programação em C++



Joyanes

McGraw Hill

2008

8586804819

Um pouco da História

- Iniciada nos anos 80, lançada em 1985;
- É uma linguagem de programação que mantém **compatibilidade** com a linguagem C;
- Dispõe de todos os recursos para programar OO;
- Possui uma grande quantidade de funcionalidades, sendo a mais importante a biblioteca de programação genérica **STL** (*Standard Template Library*).

Algumas Aplicações Desenvolvidas em C++

- Apple OS X (SO);
- AutoCAD;
- Máquina de busca do Google;
- MySQL (banco de dados);
- Pacote Office;
- Photoshop;
- Windows (SO);
- ...

Documentação

Site Oficial:

- <http://www.open-std.org/JTC1/SC22/WG21>

Ótima referência:

- <http://en.cppreference.com/w/cpp>

Ótima referência (Bibliotecas):

- <http://www.cplusplus.com/reference>

Conceitos Básicos

Conceitos Básicos

- **Tipos de Dados:**
 - Os mesmos definidos na linguagem C;
 - Tipo string (conjunto de caracteres).
- **Comando de Atribuição:**
 - O mesmo definido na linguagem C.
- **Operadores aritméticos, lógicos e relacionais:**
 - Os mesmos definidos na linguagem C.



Comandos Básicos

Bibliotecas

Herança de C (ISSO/IEC 14882):

- As bibliotecas da linguagem C podem ser utilizadas em C++, acrescentando a letra “c” na frente do nome e retirando o “.h”.

Alguns Exemplos:

- cctype, cstring, cstdlib, ctemp e cstdio.

No código:

```
#include <cstdio>
```

Exemplo: Compatibilidade com C

```
#include <stdio>

int main() {
    int numero;

    printf("digite um numero: ");
    scanf("%d", &numero);

    printf("Numero = %d\n", numero);

    return 0;
}
```

Bibliotecas

Documentação:

- <http://en.cppreference.com/w/cpp/header>

Alguns Exemplos:

- string, array, vector, deque, list, set, map, stack, queue, algorithm, iterator e iostream.

No código:

```
#include <iostream>
```

Biblioteca: iostream

- Apresenta funções para manipulação de fluxo de dados padrão do sistema:
 - Entrada padrão;
 - Saída padrão;
 - Saída de erros padrão.

cout

- Console Output;
- Equivale ao comando printf, ou seja, exibe caracteres.

Exemplo:

```
cout << "Eu adoro estudar C++\n";  
cout << "Eu adoro estudar C++" << endl;  
cout << "Eu "  
    << "adoro "  
    << "estudar "  
    << "C++";  
cout << endl;
```



cout: Base Numérica

- É possível definir a base numérica que será utilizada para exibição dos números, por exemplo: hex, oct ou dec.
- Uma vez definida a base, todas as exibições serão realizadas naquela base informada.

Exemplo:

```
int numero = 65;  
  
cout << "Hexa: " << hex << numero;  
cout << endl;  
cout << "Octal: " << oct << numero;  
cout << endl;  
cout << setbase(10);  
cout << "Decimal: " << dec << numero;
```



cout: Números Reais

- É possível exibir o número real usando a notação ponto fixo (`fixed`) ou notação científica (`scientific`);
- A precisão do número também pode ser definida.

Exemplo:

```
double numero = 1.256;  
  
cout << fixed << numero;  
cout << endl;  
cout << setprecision(2) << numero;  
cout << endl;  
cout << scientific << numero;  
cout << endl;  
cout.precision(6);  
cout << scientific << numero;
```



cout: Espaçamento

- É possível definir a quantidade mínima de caracteres.

Exemplo:

```
int numero = 65;

cout << setw(10) << numero;
cout << endl;
cout << numero << endl;
cout.width(10);
cout << numero << endl;
cout << numero << endl;
cout << setw(10) << setfill('*') << numero;
cout << endl << numero;
cout.width(10);
cout.fill('*');
cout << endl << numero;
cout << endl << numero;
```



cout: Alinhamento do Texto

- É possível determinar o alinhamento à direita (right) ou à esquerda (left);
- O alinhamento permanecerá definido até ser alterado.

Exemplo:

```
int numero = 65;

cout << '[';
cout << setw(10) << left << numero << ']';
cout << endl;
cout.width(10);
cout << numero << endl;
cout << setw(10) << right << numero;
```



cin

- Console Input;
- Equivale ao comando scanf, ou seja, realiza leitura de dados;
- Não é necessário especificar o tipo de dado que será lido. O comando ">>" (extração) resolve as conversões.

cin: exemplo

```
int numero;  
double real;  
char caractere;  
  
cout << "Informe um numero: ";  
cin >> numero;  
cout << "Numero lido: " << numero << endl;  
  
cout << "Informe um caractere: ";  
cin >> caractere;  
cout << "Caractere lido: " << caractere << endl;  
  
cout << "Informe um valor real: ";  
cin >> real;  
cout << "Real lido: " << fixed << setprecision(2) << real;
```

cin: exemplo



cin

- Os problemas com a leitura de conjunto de caracteres persistem em C++;
- O comando getline resolve esse problema.

Exemplo:

```
char frase[20];  
  
cout << "Informe uma frase: ";  
cin.getline(frase, 20);  
  
cout << "Frase lida: " << frase << endl;
```



Relação int e char

- A relação presente em C, também está presente em C++ !

```
int numero = 65;
char letra = 'A';

cout << numero << " - " << letra;
cout << endl;
cout << (char)numero << " - " << (int)letra;
cout << endl;

for (int i = 'A'; i <= 'Z'; ++i) {
    cout << (char)i << " ";
}

if (letra == 65) {
    cout << endl << "Eh um 'A'!";
}
```



Biblioteca: iostream

```
#include <iostream>

int main(){
    int numero;

    std::cout << "Informe um numero: ";
    std::cin >> numero;

    std::cout << "Numero: " << numero << std::endl;

    return 0;
}
```

```
#include <iostream>


using namespace std;

int main(){
    int numero;

    cout << "Informe um numero: ";
    cin >> numero;

    cout << "Numero: " << numero << endl;

    return 0;
}
```



Mesmo programa, usando namespace.

Estruturas de Decisão

Estruturas de Decisão

- Segue o mesmo padrão adotado em C.

Estruturas de Repetição

Estruturas de Repetição

- Segue o mesmo padrão adotado em C;
- A novidade é que o “for” pode declarar variável com escopo local:

Exemplo:

```
for (int i = 0; i < 10; i++) {  
    }
```

Exemplo - 1

Tomadas (OBI2013, Nível Júnior, Fase 1)

- A Olimpíada Internacional de Informática (IOI, no original em inglês) é a mais prestigiada competição de programação para alunos de ensino médio; seus aproximadamente 300 competidores se reúnem em um país diferente todo ano para os dois dias de prova da competição. Naturalmente, os competidores usam o tempo livre para acessar a Internet, programar e jogar em seus notebooks, mas eles se depararam com um problema: o saguão do hotel só tem uma tomada.
- Felizmente, os quatro competidores da equipe brasileira da IOI trouxeram cada um uma régua de tomadas, permitindo assim ligar vários notebooks em uma tomada só; eles também podem ligar uma régua em outra para aumentar ainda mais o número de tomadas disponíveis. No entanto, como as réguas têm muitas tomadas, eles pediram para você escrever um programa que, dado o número de tomadas em cada régua, determina quantas tomadas podem ser disponibilizadas no saguão do hotel.

Exemplo - 1

```
#include <iostream>

using namespace std;

int main() {
    int t1,t2,t3,t4;

    cin >> t1 >> t2 >> t3 >> t4;

    cout << t1 + t2 + t3 + t4 - 3 << endl;

    return 0;
}
```

Exemplo - 2

Cédulas (URI Online Judge | 1018)

- Leia um valor inteiro. A seguir, calcule o menor número de notas possíveis (cédulas) no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2 e 1. A seguir mostre o valor lido e a relação de notas necessárias.

Exemplo de Entrada	Exemplo de Saída
576	576 5 nota(s) de R\$ 100,00 1 nota(s) de R\$ 50,00 1 nota(s) de R\$ 20,00 0 nota(s) de R\$ 10,00 1 nota(s) de R\$ 5,00 0 nota(s) de R\$ 2,00 1 nota(s) de R\$ 1,00

Exemplo - 2

```
#include <iostream>

using namespace std;

int main(){
    int valor;

    cin >> valor;

    cout << valor << endl;

    cout << valor/100 << " nota(s) de R$ 100,00" << endl;
    valor %= 100;
    cout << valor/50 << " nota(s) de R$ 50,00" << endl;
    valor %= 50;
    cout << valor/20 << " nota(s) de R$ 20,00" << endl;
    valor %= 20;
    cout << valor/10 << " nota(s) de R$ 10,00" << endl;
    valor %= 10;
    cout << valor/5 << " nota(s) de R$ 5,00" << endl;
    valor %= 5;
    cout << valor/2 << " nota(s) de R$ 2,00" << endl;
    valor %= 2;
    cout << valor << " nota(s) de R$ 1,00" << endl;

    return 0;
}
```

Vetores

Vetores

- Unidimensionais e bidimensionais;
- Seguem o mesmo padrão adotado em C.

Exemplo (URI 1018)

```
#include <iostream>

using namespace std;

int main(){
    int valor, cedulas[] = {100,50,20,10,5,2};

    cin >> valor;

    cout << valor << endl;

    for (int i = 0; i < 6; ++i){
        cout << valor/cedulas[i] << " nota(s) de R$ "
            << cedulas[i] << ",00" << endl;
        valor %= cedulas[i];
    }

    cout << valor << " nota(s) de R$ 1,00" << endl;

    return 0;
}
```

Alocação Dinâmica de Memória

Alocação Dinâmica de Memória

- É a alocação da memória em tempo de execução do software;
- É necessário usar um apontador para armazenar o endereço de memória que foi alocada, semelhante ao que acontece com C. Ou seja, nada mudou ! ;-)

Alocação Dinâmica de Memória

- Operador “new”:
 - Aloca um bloco de memória. Retorna o endereço do bloco alocado, caso contrário (memória indisponível) retorna NULL.

Exemplo:

```
int *pt, qtde;
```

```
cin >> qtde;
```

```
pt = new int[qtde];
```

```
int **pt, linha, coluna;
```

```
cin >> linha >> coluna;
```

```
pt = new int*[linha];
```

```
for (int i = 0; i < linha; ++i)  
    pt[i] = new int[coluna];
```

Alocação Dinâmica de Memória

- Operador “delete”:
 - Libera um bloco de memória alocada dinamicamente.

Exemplo:

```
int *num, *pt;
```

```
num = new int;
```

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

```
delete num;
```

```
delete[] pt;
```