

Curso Completo de Algoritmos e Lógica de Programação

educandoweb.com.br

Prof. Dr. Nelio Alves

Capítulo: Linguagem C++

Atenção

Este material foi elaborado para alunos do curso **Algoritmos e Lógica de Programação** do professor Nelio Alves.

A didática do conteúdo deste material está adaptada para quem já fez a parte do referido curso sobre construção de algoritmos na linguagem do VisualG.

Para mais informações:
educandoweb.com.br

Instalação das ferramentas

IDE : Code Blocks

<http://www.codeblocks.org/downloads>

- Download the binary release
- Escolher pacote de instalação (setup) com **Code Blocks + Mingw**

Primeiro programa em C++

VisualG	C++
Algoritmo "primeiro"	<code>#include <iostream></code>
Var	<code>using namespace std;</code>
Inicio	<code>int main()</code>
escreval("Ola mundo!")	<code>{</code>
Fimalgoritmo	<code> cout << "Ola mundo!" << endl;</code>
	<code> return 0;</code>
	<code>}</code>

TIPOS DE DADOS E VARIÁVEIS

Significado	Tipo VisualG	Tipo C++	Valor padrão	Observação
número inteiro	inteiro	int	não atribuído (lixo de memória)	int : -32767 a 32767 ou -2147483648 a 2147483647 long int : -2147483648 a 2147483647 long long int : -9223372036854775807 a 9223372036854775807
número de ponto flutuante	real	double	não atribuído (lixo de memória)	float : precisão simples double : precisão dupla
um único caractere	caractere	char	não atribuído (lixo de memória)	Valores literais devem ter aspas simples. Exemplo: 'F'
texto	caractere	string	não atribuído (lixo de memória)	Valores literais devem ter aspas duplas. Exemplo: "Maria"
valor lógico	logico	bool	não atribuído (lixo de memória)	Valores possíveis: false / true

Referência:

<https://pt.cppreference.com/w/cpp/language/types>

Exemplo: declaração e atribuição de variáveis

VisualG	C++
<p>Algoritmo "teste"</p> <p>Var</p> <pre> idade : inteiro salario, altura : real genero : caractere nome : caractere </pre> <p>Inicio</p> <pre> idade <- 20 salario <- 5800.5 altura <- 1.63 genero <- "F" nome <- "Maria Silva" escreval("IDADE = ", idade) escreval("SALARIO = ", salario:4:2) escreval("ALTURA = ", altura:4:2) escreval("GENERO = ", genero) escreval("NOME = ", nome) </pre> <p>Fimalgoritmo</p>	<pre> #include <iostream> #include <iomanip> #include <string> using namespace std; int main() { int idade; double salario, altura; char genero; string nome; idade = 20; salario = 5800.5; altura = 1.63; genero = 'F'; nome = "Maria Silva"; cout << fixed << setprecision(2); cout << "IDADE = " << idade << endl; cout << "SALARIO = " << salario << endl; cout << "ALTURA = " << altura << endl; cout << "GENERO = " << genero << endl; cout << "NOME = " << nome << endl; return 0; } </pre>

NOTA: A linguagem C++ aceita atribuição diretamente na declaração da variável, inclusive para texto. Por exemplo:

```

int idade = 20;
string nome = "Maria Silva";

```

OPERADORES EM C++

Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão ("mod")

Operadores comparativos

Operador	Significado
<	menor
>	maior
<=	menor ou igual
>=	maior ou igual
==	igual
!=	diferente

Operadores lógicos

Operador	Significado
&&	e
	ou
!	não

SAÍDA DE DADOS EM C++

Comando no VisualG	Comando em C++	Biblioteca / namespace
escreva / escreval	cout <<	<code>#include <iostream></code> <code>using namespace std;</code>

Formatação de ponto flutuante	Biblioteca
cout << fixed << setprecision(2)	<code>#include <iomanip></code> <code>using namespace std;</code>

Biblioteca universal (inclui tudo e dispensa bibliotecas individuais)
<code>#include <bits/stdc++.h></code>

SAÍDA DE DADOS EM C++

Exemplo VisualG	Exemplo C++	Resultado na tela
<pre>escreva("Bom dia") escreva("Boa noite")</pre>	<pre>cout << "Bom dia"; cout << "Boa noite";</pre>	Bom diaBoa noite
<pre>escreval("Bom dia") escreval("Boa noite")</pre>	<pre>cout << "Bom dia" << endl; cout << "Boa noite" << endl;</pre>	Bom dia Boa noite
<pre>x, y : inteiro x <- 10 y <- 20 escreval(x) escreval(y)</pre>	<pre>int x, y; x = 10; y = 20; cout << x << endl; cout << y << endl;</pre>	10 20
<pre>x : real x <- 2.3456 escreval(x:4:2)</pre>	<pre>double x; x = 2.3456; cout << fixed << setprecision(2) << x << endl;</pre>	2.35
<pre>idade : inteiro salario : real nome : caractere sexo : caractere idade <- 32 salario <- 4560.9 nome <- "Maria Silva" sexo <- "F" escreval("A funcionaria ", nome, ", sexo ", sexo, ", ganha ", salario:8:2, " e tem ", idade, " anos.")</pre>	<pre>int idade; double salario; string nome; char sexo; idade = 32; salario = 4560.9; nome = "Maria Silva"; sexo = 'F'; cout << fixed << setprecision(2); cout << "A funcionaria " << nome << ", sexo " << sexo << ", ganha " << salario << " e tem " << idade << " anos" << endl;</pre>	A funcionaria Maria Silva, sexo F, ganha 4560.90 e tem 32 anos

PROCESSAMENTO DE DADOS / CASTING EM C++

Exemplo VisualG	Exemplo C++	Resultado na tela
<pre>x, y : inteiro x <- 5 y <- 2 * x escreval(x) escreval(y)</pre>	<pre>int x, y; x = 5; y = 2 * x; cout << x << endl; cout << y << endl;</pre>	<pre>5 10</pre>
<pre>x : inteiro y : real x <- 5 y <- 2 * x escreval(x) escreval(y)</pre>	<pre>int x; double y; x = 5; y = 2 * x; cout << x << endl; cout << fixed << setprecision(1) << y << endl;</pre>	<pre>5 10.0</pre>
<pre>b1, b2, h, area : real b1 <- 6.0 b2 <- 8.0 h <- 5.0 area <- (b1 + b2) / 2.0 * h; escreval(area)</pre>	<pre>double b1, b2, h, area; b1 = 6.0; b2 = 8.0; h = 5.0; area = (b1 + b2) / 2.0 * h; cout << area << endl;</pre>	<pre>35</pre>
<pre>a, b, resultado : inteiro a <- 5 b <- 2 resultado <- a \ b escreval(resultado)</pre>	<pre>int a, b, resultado; a = 5; b = 2; resultado = a / b; cout << resultado << endl;</pre>	<pre>2</pre>
<pre>a : real b : inteiro a <- 5.0 b <- Int(a) escreval(b)</pre>	<pre>double a; int b; a = 5.0; b = (int) a; cout << b << endl;</pre>	<pre>5</pre>

ENTRADA DE DADOS EM C++

Comando no VisualG	Comando em C++	Biblioteca
leia	<code>cin >></code> <code>getline</code> (para ler texto até a quebra de linha)	<code>#include <iostream></code> <code>using namespace std;</code>

Comando para limpeza de buffer

```
cin.ignore(INT_MAX, '\n');
```

QUANDO USAR: quando você for ler um texto até a quebra de linha (`getline`), mas antes o seu programa já leu algum outro dado e deixou uma quebra de linha pendente. Por exemplo:

```
cout << "Digite sua idade: ";  
cin >> idade;  
cout << "Digite seu nome completo: ";  
cin.ignore(INT_MAX, '\n');    // ----- limpeza de buffer  
getline(cin, nome);
```


ENTRADA DE DADOS EM C++

Exemplo VisualG	Exemplo C++
<p>Algoritmo "teste_entrada"</p> <p>Var</p> <p> salario1, salario2 : real nome1, nome2 : caractere idade : inteiro sexo : caractere</p> <p>Inicio</p> <p> escreva("Nome da primeira pessoa: ") leia(nome1) escreva("Salario da primeira pessoa: ") leia(salario1)</p> <p> escreva("Nome da segunda pessoa: ") leia(nome2) escreva("Salario da segunda pessoa: ") leia(salario2)</p> <p> escreva("Digite uma idade: ") leia(idade) escreva("Digite um sexo (F/M): ") leia(sexo)</p> <p> escreval("Nome 1: ", nome1) escreval("Salario 1: ", salario1:4:2) escreval("Nome 2: ", nome2) escreval("Salario 2: ", salario2:4:2) escreval("Idade: ", idade) escreval("Sexo: ", sexo)</p> <p>Fimalgoritmo</p>	<pre>#include <iostream> #include <iomanip> #include <string> #include <climits> using namespace std; int main() { double salario1, salario2; string nome1, nome2; int idade; char sexo; cout << "Nome da primeira pessoa: "; getline(cin, nome1); cout << "Salario da primeira pessoa: "; cin >> salario1; cout << "Nome da segunda pessoa: "; cin.ignore(INT_MAX, '\n'); // ----- limpeza de buffer getline(cin, nome2); cout << "Salario da segunda pessoa: "; cin >> salario2; cout << "Digite uma idade: "; cin >> idade; cout << "Digite um sexo (F/M): "; cin >> sexo; cout << fixed << setprecision(2); cout << "Nome 1: " << nome1 << endl; cout << "Salario 1: " << salario1 << endl; cout << "Nome 2: " << nome2 << endl; cout << "Salario 2: " << salario2 << endl; cout << "Idade: " << idade << endl; cout << "Sexo: " << sexo << endl; return 0; }</pre>

COMO EXECUTAR O DEBUGGER NO CODE BLOCKS

ATENÇÃO: o debug NÃO FUNCIONA para arquivos isolados. Seu programa deve estar dentro de um PROJETO.

PASSOS PARA CONFIGURAR O DEBUGGER:

- Settings -> Compiler -> Toolchain Executables
 - Debugger: GDB/CDB debugger: default
- Settings -> Debugger -> GDB/CDB debugger -> Default
 - Executable path: C:\Program Files (x86)\CodeBlocks\MinGW\bin\gdb32.exe

COMANDOS DO DEBUGGER:

- Habilitar/desabilitar breakpoint: **F5**
- Iniciar o debug: **F8**
- Rodar um passo: **F7**
- Parar o debug: **SHIFT+F8**
- Mostrar variáveis: **Debug -> Debugging windows -> Watches**

ESTRUTURA CONDICIONAL EM C++

Simple	Composta	Encadeamento
<pre>if (condição) { comando1 comando2 }</pre>	<pre>if (condição) { comando1 comando2 } else { comando3 comando4 }</pre>	<pre>if (condição1) { comando1 comando2 } else if (condição2) { comando3 comando4 } else { comando5 comando6 }</pre>

Exemplo VisualG	Exemplo C++
<p>Algoritmo "teste_condicional"</p> <p>Var</p> <p> hora : inteiro</p> <p>Início</p> <p> escreva("Digite uma hora do dia: ")</p> <p> leia(hora)</p> <p> se hora < 12 então</p> <p> escreval("Bom dia!")</p> <p> senão</p> <p> escreval("Boa tarde!")</p> <p> fimse</p> <p>Fimalgoritmo</p>	<pre>#include <iostream> using namespace std; int main() { int hora; cout << "Digite uma hora do dia: "; cin >> hora; if (hora < 12) { cout << "Bom dia!" << endl; } else { cout << "Boa tarde!" << endl; } return 0; }</pre>

ESTRUTURA ENQUANTO EM C++

Sintaxe	Regra
<pre>while (condição) { comando1 comando2 }</pre>	V: executa e volta F: pula fora

Exemplo VisualG	Exemplo C++
<pre>Algoritmo "teste_enquanto" Var x, soma : inteiro Inicio soma <- 0 escreva("Digite o primeiro numero: ") leia(x) enquanto x <> 0 faca soma <- soma + x escreva("Digite outro numero: ") leia(x) fimenquanto escreval("SOMA = ", soma) Fimalgoritmo</pre>	<pre>#include <iostream> using namespace std; int main() { int x, soma; soma = 0; cout << "Digite o primeiro numero: "; cin >> x; while (x != 0) { soma = soma + x; cout << "Digite outro numero: "; cin >> x; } cout << "SOMA = " << soma; return 0; }</pre>

ESTRUTURA PARA EM C++

Sintaxe / regra

Executa somente
na primeira vez

V: executa e volta
F: pula fora

Executa toda vez depois
de voltar

```
for ( início ; condição ; incremento) {
    comando1
    comando2
}
```

Exemplo VisualG

Algoritmo "teste_para"

Var

N, i, x, soma : inteiro

Inicio

escreva("Quantos numeros serao digitados? ")
leia(N)

soma <- 0
para i de 1 ate N faca
 escreva("Digite um numero: ")
 leia(x)
 soma <- soma + x
fimpara

escreval("SOMA = ", soma)

Fimalgoritmo

Exemplo C++

```
#include <iostream>
using namespace std;
int main() {
    int N, i, x, soma;
    cout << "Quantos numeros serao digitados? ";
    cin >> N;

    soma = 0;
    for (i = 1; i <= N; i++) {
        cout << "Digite um numero: ";
        cin >> x;
        soma = soma + x;
    }

    cout << "SOMA = " << soma << endl;

    return 0;
}
```

ESTRUTURA DO-WHILE EM C++ (variante da estrutura REPITA-ATÉ)

Sintaxe	Regra
<pre>do { comando 1 comando 2 } while (condição);</pre>	<p>V: volta F: pula fora</p>

Exemplo VisualG	Exemplo C++
<p>Algoritmo "exemplo_repita_ate"</p> <p>Var</p> <p> C, F : real</p> <p> resp : caractere</p> <p>Inicio</p> <p> repita</p> <p> escreva("Digite a temperatura em Celsius: ")</p> <p> leia(C)</p> <p> F <- 9.0 * C / 5.0 + 32.0</p> <p> escreval("Equivalente em Fahrenheit: ", F:6:1)</p> <p> escreva("Deseja repetir (s/n)? ")</p> <p> leia(resp)</p> <p> ate resp <> "s"</p> <p>Fimalgoritmo</p>	<pre>#include <bits/stdc++.h> using namespace std; int main() { double C, F; char resp; do { cout << "Digite a temperatura em Celsius: "; cin >> C; F = 9.0 * C / 5.0 + 32.0; cout << "Equivalente em Fahrenheit: " << F << endl; cout << "Deseja repetir (s/n)? "; cin >> resp; } while (resp == 's'); return 0; }</pre>

VETORES EM C++

Exemplo VisualG	Exemplo C++
<p>Algoritmo "teste_vetor"</p> <p>Var</p> <p>vet: vetor [0..9] de real N, i : inteiro</p> <p>Inicio</p> <p>escreva("Quantos numeros voce vai digitar? ") leia(N)</p> <p>para i de 0 ate N-1 faca escreva("Digite um numero: ") leia(vet[i]) fimpara</p> <p>escreval escreval("NUMEROS DIGITADOS:") para i de 0 ate N-1 faca escreval(vet[i]:8:1) fimpara</p> <p>Fimalgoritmo</p>	<pre>#include <iostream> #include <iomanip> using namespace std; int main() { int N, i; cout << "Quantos numeros voce vai digitar? "; cin >> N; double vet[N]; for (i = 0; i < N; i++) { cout << "Digite um numero: "; cin >> vet[i]; } cout << endl << "NUMEROS DIGITADOS:" << endl; cout << fixed << setprecision(1); for (i = 0; i < N; i++) { cout << vet[i] << endl; } return 0; }</pre>

MATRIZES EM C++

Exemplo VisualG	Exemplo C++
<p>Algoritmo "teste_matriz"</p> <p>Var</p> <p>mat: vetor [0..4, 0..4] de inteiro</p> <p>M, N, i, j : inteiro</p> <p>Inicio</p> <p> escreva("Quantas linhas vai ter a matriz? ")</p> <p> leia(M)</p> <p> escreva("Quantas colunas vai ter a matriz? ")</p> <p> leia(N)</p> <p> para i de 0 ate M-1 faca</p> <p> para j de 0 ate N-1 faca</p> <p> escreva("Elemento [", i, ",", j, "]: ")</p> <p> leia(mat[i, j])</p> <p> fimpara</p> <p> fimpara</p> <p> escreval</p> <p> escreval("MATRIZ DIGITADA:")</p> <p> para i de 0 ate M-1 faca</p> <p> para j de 0 ate N-1 faca</p> <p> escreva(mat[i, j])</p> <p> fimpara</p> <p> escreval</p> <p> fimpara</p> <p>Fimalgoritmo</p>	<pre>#include <iostream> using namespace std; int main() { int M, N, i, j; cout << "Quantas linhas vai ter a matriz? "; cin >> M; cout << "Quantas colunas vai ter a matriz? "; cin >> N; int mat[M][N]; for (i = 0; i < M; i++) { for (j = 0; j < N; j++) { cout << "Elemento [" << i << "," << j << "]: "; cin >> mat[i][j]; } } cout << endl << "MATRIZ DIGITADA:" << endl; for (i = 0; i < M; i++) { for (j = 0; j < N; j++) { cout << mat[i][j] << " "; } cout << endl; } return 0; }</pre>