

Módulo 1

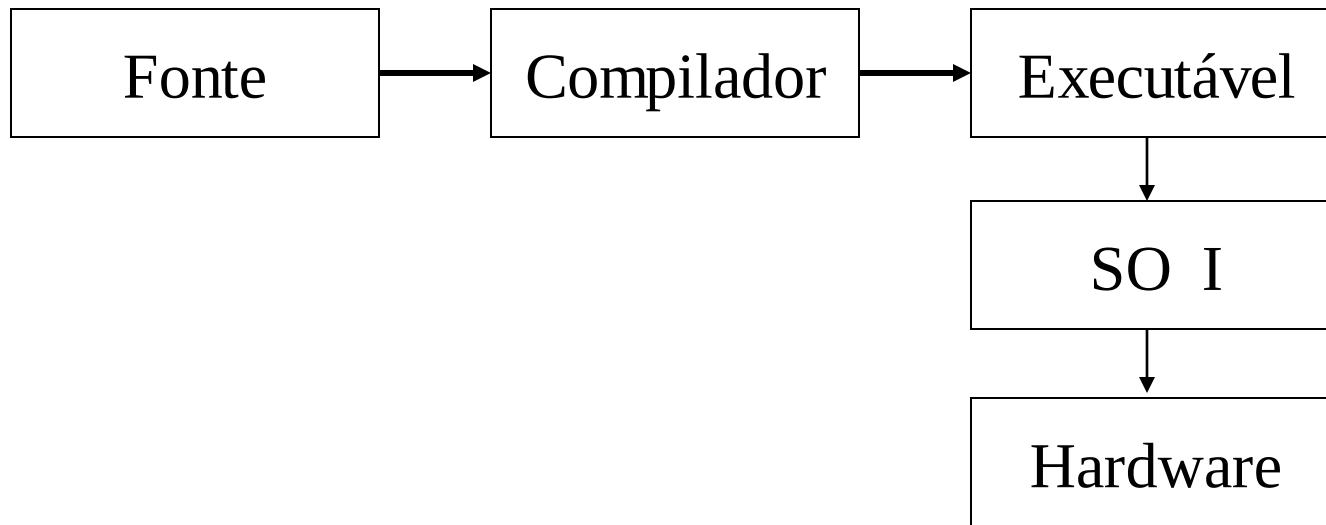
Introdução

AEDS I

C++

(Rone Ilídio)

Introdução – Linguagens de Programação



C++ - Características

- Evolução do C, ou seja, possui a mesma sintaxe
- É uma linguagem orientada a objetos
- Possui uma biblioteca mais vasta que o C
- Fácil aprendizado

Primeiro programa em C++

```
#include <iostream.h>
```

```
void main(){
```

```
    cout << "Primeiro Programa!";
```

```
}
```

Alterando o primeiro programa

```
#include <stdio.h>
#include <iostream.h>
int main()
{
    cout << "Primeiro Programa!"<<
        "\nProgramador Rone Ilidio";
    getch();
}
```

Variáveis

- Variável é um espaço de memória reservado para armazenar um certo tipo de dado e tendo um nome (identificador) para referenciar o conteúdo.
- Em C++ as variáveis podem ser declaradas em qualquer lugar do programa
- Cuidado → C++ é case sensitive

Variáveis

```
#include <iostream.h>
```

```
#include <stdio.h>
```

```
void main(){
```

```
    int num;
```

```
    num = 18;
```

```
    cout << "O numero associado a variavel num e:"
```

```
    << num;
```

```
    getchar();
```

```
}
```

Tipos de variáveis

Tipo	Bit	Escala	Exemplo
char	8	-128 a 127	‘a’, ‘A’, ‘1’, ‘@’
int	16	-32768 a 32767	3, -13, 0
float	32	3.4E-38 a 3.4E+38	-3.14
double	62	1.7E-308 a 1.7E+308	
void	0	nenhum valor	nenhum valor

Tipos das variáveis - modificadores

- Os modificadores alteram os tipos predefinidos.
São eles: long, short e unsigned

Tipo	Bit	Escala	Exemplo
unsigned	16	0 a 65535	1, 15, 1000
short	16	-32768 a 32767	3, -13, 0
long	32	-2147483648 a 2147483647	
unsigned long	32	0 a 4294967295	
long double	80	3.4E-4932 a 1.1E+4932	

Inicializando Variáveis

```
#include <iostream.h>
#include <stdio.h>
void main(){
    int evento = 5;
    char corrida = 'C';
    float tempo = 27.25;
    cout << "O tempo vitorioso na eliminatoria " << corrida <<
        " da competicao " << evento << " foi " << tempo << ".";
    getch();
}
```

Variáveis - String

Não existe o tipo string, por isso é utilizado um “vetor” de char da seguinte forma:

```
#include <iostream.h>
```

```
#include <stdio.h>
```

```
void main(){
```

```
    char t[30]="ufsj";
```

```
    cout << "O valor da variavel t eh:" << t;
```

```
    getchar();
```

```
}
```

Constantes

- São variáveis que não podem ter seu valor alterado
- Antes de sua declaração é colocada a palavra `const`
- Sempre dever ser inicializada
- Ex:

```
const int x = 10;
```

Constantes

```
#include <iostream.h>
#include <stdio.h>
void main(){
    const float pi = 3.14;
    float raio = 10;
    float a = pi * (raio * raio);
    cout << "A área do círculo de raio 10 é:" << a;
    getchar();
}
```

Operadores

- Atribuição: =
- Aritméticos:
 - Soma: +
 - Subtração: -
 - Multiplicação: *
 - Divisão: /
 - Modulo: %

Precedência

- $*$, $/$ e $\%$: mesma precedência, a maior
- $+$ e $-$: mesma precedência
- $=$: menor de todas
- Em caso de empate resolve-se da esquerda para a direita
- O uso de parênteses altera a precedencia

Precedência

```
#include <iostream.h>    x = 4 + 3 * 8 / 2 - 1;
#include <stdio.h>        5   3   1   2   3
void main(){
    int x = 4 + 3 * 8 / 2 - 1;
    cout << "O resultado: " << x; //15
    getchar();
}
```

Entrada de dados do usuário

- O comando “cin”, associado ao operador de extração “>>”, implementam a entrada de dados do usuário
- Obs: cin da biblioteca `iostream.h`

Entrada de dados do usuário

```
#include <iostream.h>
#include <stdio.h>
void main(){
    cout << "Entre com sua idade: ";
    int idade;
    cin >> idade;
    cout << "voce tem " << idade * 365 << " dias de vida.";
    getchar();
}
```

Múltiplas entradas

- As múltiplas entradas são digitadas separadas por espaço em branco. O objeto cin entende um espaço em branco como término de uma entrada e o [ENTER] como finalizador geral.

Múltiplas entradas

```
#include <iostream.h>
#include <stdio.h>
void main(){
    cout << "Entre com as notas das 4 provas: ";
    float p1, p2, p3, p4;
    cin >> p1 >> p2 >> p3 >> p4;
    cout << "Sua media eh: " << (p1 +p2 + p3 +p4)/4;
    getchar() ;
}
```

Funções `getche()` e `getchar()` `getch()`

- Essas funções retornam a entrada de um caractere do teclado
- A diferença entre elas é que `getche()` mostra o caractere na tela, `getch()` e `getchar()` não mostra
- `getch()` e `getchar()` precisam de *enter* no final

Funções getch() e getch()

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
void main(){
    cout << "Escolha 1 ou 2: ";
    char p;
    p = getch();
    //p = getche();
    cout << "\nVoce escolheu: " << p;

    getch();
    getch();
}
```

Operadores de incremento e decremento

- Para incrementar uma variável X você pode utilizar: $X = X + 1$;
- Uma outra forma similar é utilizar o operador $++$ das seguintes formas:
 - $X++$;
 - $++X$;
- O mesmo raciocínio é válido para decremento: $X--$ e $--X$
- Obs: esses operadores têm precedência maior que os operadores $*$, $/$ e $\%$

Operadores de incremento e decremento

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main(){
    clrscr();
    int x = 1;
    cout << "\nx = " << x;
    cout << "\nx = " << ++x;
    cout << "\nx = " << x++;
    cout << "\nx = " << x;
    char *r;
    gets(r);
}
```

Operadores aritméticos e de atribuição

- Regra geral:
 $x \text{ op} = \text{exp} \rightarrow x = x \text{ op} (\text{exp})$
 - onde x é uma variável
 - op é um operador ($*$, $/$, $+$, $-$ ou $\%$)
 - exp é uma expressão matemática qualquer
- Exemplo: $i += 2;$ equivale a $i = i + 2;$
 $x *= Y + 1;$ equivale a $x = x * (Y + 1);$
 $t /= 2.5;$ equivale a $t = t / 2.5$
 $p \% = 6$ equivale a $p = p \% 6$
 $d -= 3$ equivale a $d = d - 3$