

Programação de Computadores - CI-208

Prof. Murilo da Silva - DINF/UFPR

Aula 07

Escrevendo várias vezes a mesma coisa na tela:

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7
8     cout<<"Sexta-feira eh feriado!" << endl;
9     cout<<"Sexta-feira eh feriado!" << endl;
10    cout<<"Sexta-feira eh feriado!" << endl;
11    cout<<"Sexta-feira eh feriado!" << endl;
12    cout<<"Sexta-feira eh feriado!" << endl;
13    cout<<"Sexta-feira eh feriado!" << endl;
14    cout<<"Sexta-feira eh feriado!" << endl;
15    cout<<"Sexta-feira eh feriado!" << endl;
16    cout<<"Sexta-feira eh feriado!" << endl;
17    cout<<"Sexta-feira eh feriado!" << endl;
18
19    return 0;
20 }
```

Escrevendo várias vezes a mesma coisa na tela:

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int contador;
8
9     contador = 0;
10
11     while (contador < 10) {
12
13         cout<<"Sexta-feira eh feriado!" << endl;
14         contador = contador + 1;
15     }
16
17     return 0;
18 }
```

Padrão repetitivo

O que você quer repetir?

```
cout<<"Sexta-feira eh feriado!" << endl;
```

```
while (expressão)
```

```
cout<<"Sexta-feira eh feriado!" << endl;
```

Você tem que reconhecer o **Padrão Repetitivo**

Controle da repetição

Como controlar a repetição?

Inicialização:

contador começa com valor **0** (zero)
[No início, não começou a contagem e nenhuma mensagem exibida ainda]

Incremento:

a cada repetição, conta mais **1**
[Conta a mensagem exibida]

```
contador = 0;
```

```
while (expressão)
```

```
cout<<"Sexta-feira eh feriado!" << endl;
```

```
contador = contador + 1;
```

Bloco de repetição

O incremento **TAMBÉM** deve fazer parte da repetição:
bloco de comandos da repetição

Os comandos entre { e } criam um único
bloco de comandos.

bloco da repetição:
conjunto de comandos
que se repetem

```
contador = 0;
```

```
while (expressão)
```

```
{
```

```
cout<<"Sexta-feira eh feriado!" << endl;
```

```
contador = contador + 1;
```

```
}
```

Condição de Parada

A repetição continua enquanto o que for verdadeiro ($\neq 0$) ?

Condição de parada:

repete-se a mensagem enquanto o contador variar de 0 a 9 (inclusive)

Terminada a repetição:

contador \rightarrow número de repetições \rightarrow 10

```
contador = 0;
```

```
while ( contador < 10 )  
{
```

```
    cout << "Sexta-feira eh feriado!" << endl;
```

```
    contador = contador + 1;
```

```
}
```

```
// O que é o valor de contador neste ponto?
```

Escrevendo várias vezes a mesma coisa na tela:

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int contador;
8
9     contador = 0;
10
11     while (contador < 10) {
12
13         cout<<"Sexta-feira eh feriado!" << endl;
14         contador = contador + 1;
15     }
16
17     return 0;
18 }
```

Escrever os números de 1 a 10

```
/* Programa 'escreve1a10' */  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout << 1 << endl;  
    cout << 2 << endl;  
    cout << 3 << endl;  
    cout << 4 << endl;  
    cout << 5 << endl;  
    cout << 6 << endl;  
    cout << 7 << endl;  
    cout << 8 << endl;  
    cout << 9 << endl;  
    cout << 10 << endl;  
    return 0;  
}
```

Existe um padrão
repetitivo, mas o valor
muda em cada repetição

→ Então isto é uma
variável

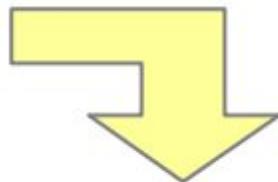
```
/* Programa 'escreve1a10' */  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int contador;  
  
    /* inicialização */  
    while ( /* expressão */ ) {  
        cout << contador << endl;  
  
        /* incremento */  
    }  
  
    return 0;  
}
```

Escrever os números de 1 a 10

```
/* Programa 'escreve1a10' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int contador;  
  
    contador = 1; // inicialização  
    while ( contador <= 10 )  
    {  
        cout << contador << endl;  
        contador = contador + 1; // incremento  
    }  
    return 0;  
}
```

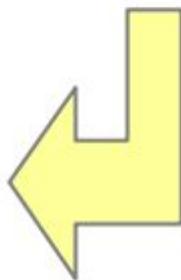
Escrever os inteiros pares no intervalo 1..100

```
/* Programa 'pares1a100' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << 2 << endl;  
    cout << 4 << endl;  
    cout << 6 << endl;  
    ....  
    cout << 100 << endl;  
    return 0;  
}
```



O padrão repetitivo é o mesmo,
mas vamos mudar
o nome da variável de contagem para **k**:

```
/* inicialização */  
while ( /* expressão */ ) {  
    cout << k << endl;  
    /* incremento */  
}
```



Escrever os inteiros pares no intervalo 1..100

Repete enquanto valor de **k** menor ou igual a 100:

```
int k;  
k = 2;  
while ( k <= 100 ) {  
    cout << k << endl;  
    k = k + 2;  
}
```

Programa completo:

```
/* Programa 'pares1a100' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int k;  
    k = 2;  
    while ( k <= 100 ) {  
        cout << k << endl;  
        k = k + 2;  
    }  
  
    return 0;  
}
```

Exercício

Escrever um programa que leia dois valores inteiros $n1$, $n2$, onde $n1 \leq n2$, e exiba na tela os valores inteiros no intervalo $[n1, n2]$.

Qual o valor inicial de k ?

Qual a condição de parada?

Escrever os números de **n1** a **n2**

```
/* Programa 'escreve_N1aN2' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int n1, n2, k;  
  
    cin >> n1 >> n2; // Leitura dos extremos do intervalo  
  
    k = n1;          // Inicialização de k ← extremo inferior  
  
    while ( k <= n2 ) // Condição de parada → k atinge extremo superior  
    {  
        cout << k << endl;  
        k = k + 1;    // Incremento de k  
    }  
  
    return 0;  
}
```

Escrever valores de uma série

1^2 2^2 3^2 $4^2 \dots$ 100^2

```
/* Programa 'seriePotencias' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << 1 * 1 << endl;  
    cout << 2 * 2 << endl;  
    cout << 3 * 3 << endl;  
    cout << 4 * 4 << endl;  
    ....  
    cout << 100 * 100 << endl;  
    return 0;  
}
```

padrão repetitivo

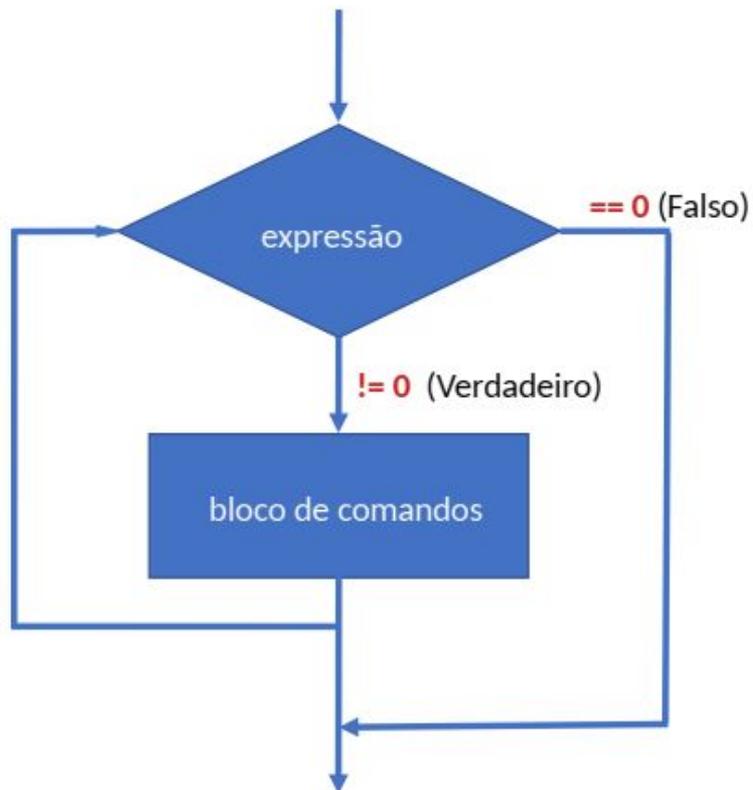
```
while ( ..... )  
{  
    cout << k * k << endl;  
    /* incremento */  
}
```

Escrever o programa completo.

Escrever valores de uma série

```
/* Programa 'seriePotencias' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int k;  
    k = 1;           // Inicialização de  $k \leftarrow 1$   
    while ( k <= 100 ) // Condição de parada  $\rightarrow k > 100$   
    {  
        cout << k * k << endl; // Imprime o quadrado de  $k$   
        k = k + 1;           // Avança para o próximo inteiro  
    }  
    return 0;  
}
```

Comando de repetição: **while**



Semântica

Enquanto o valor de **expressão** for **!= 0** (Verdadeiro), executa **bloco de comandos**.

Sintaxe em C++

```
while (expressão)  
    bloco de comandos;
```

```
{  
    comando 1;  
    comando 2;  
    .....  
}
```