

# Linguagem C: while e do-while

## Introdução à linguagem C(++)

Profs. Anilton, Lasaro e Paulo Coelho

Faculdade de Computação  
Universidade Federal de Uberlândia



# Organização

Introdução

`while`

Exemplos

Exercícios



# Introdução I

- ▶ Suponha de você deseja fazer um programa para calcular ler duas notas, calcular e imprimir a média de dez alunos da disciplina. A maneira menos prática de fazer isso seria:

```
1 ...
2 float nota1, nota2, media;
3 cout << "Entre nota 1 e nota 2 do aluno 1: " << endl;
4 cin << nota1 << nota2;
5 media = (nota1 + nota2) / 2;
6 cout << "A media das notas eh " << media << endl;
7 cout << "Entre nota 1 e nota 2 do aluno 2: " << endl;
8 cin << nota1 << nota2;
9 media = (nota1 + nota2) / 2;
10 cout << "A media das notas eh " << media << endl;
11 cout << "Entre nota 1 e nota 2 do aluno 3: " << endl;
12 cin << nota1 << nota2;
13 media = (nota1 + nota2) / 2;
14 cout << "A media das notas eh " << media << endl;
15 cout << "Entre nota 1 e nota 2 do aluno 4: " << endl;
16 cin << nota1 << nota2;
17 media = (nota1 + nota2) / 2;
```

# Introdução II

```
18 cout << "A media das notas eh " << media << endl;
19 cout << "Entre nota 1 e nota 2 do aluno 5: " << endl;
20 cin << nota1 << nota2;
21 media = (nota1 + nota2) / 2;
22 cout << "A media das notas eh " << media << endl;
23 cout << "Entre nota 1 e nota 2 do aluno 6: " << endl;
24 cin << nota1 << nota2;
25 media = (nota1 + nota2) / 2;
26 cout << "A media das notas eh " << media << endl;
27 cout << "Entre nota 1 e nota 2 do aluno 7: " << endl;
28 cin << nota1 << nota2;
29 media = (nota1 + nota2) / 2;
30 cout << "A media das notas eh " << media << endl;
31 cout << "Entre nota 1 e nota 2 do aluno 8: " << endl;
32 cin << nota1 << nota2;
33 media = (nota1 + nota2) / 2;
34 cout << "A media das notas eh " << media << endl;
35 cout << "Entre nota 1 e nota 2 do aluno 9: " << endl;
36 cin << nota1 << nota2;
37 media = (nota1 + nota2) / 2;
38 cout << "A media das notas eh " << media << endl;
39 cout << "Entre nota 1 e nota 2 do aluno 10: " << endl;
```

# Introdução III

```
40 cin << nota1 << nota2;  
41 media = (nota1 + nota2) / 2;  
42 cout << "A media das notas eh " << media << endl;  
43 ...
```

- ▶ propenso a erros
- ▶ difícil de estender
- ▶ retrabalho
- ▶ A linguagem C fornece estruturas de repetições, as quais permitem repetir um determinado conjunto de comandos.

## while I

- ▶ O comando `while` permite repetir um bloco de comandos.
- ▶ Sua forma geral é muito simples:

```
1 while (<condicao>)  
2 {  
3 // bloco de comandos  
4 // a ser repetido  
5 }
```

- ▶ O bloco de comandos entre as chaves será repetido **enquanto** a condição dentro dos parênteses for verdadeira.

## while II

- ▶ De volta ao nosso exemplo, podemos reescrevê-lo de maneira mais inteligente:

```
1 ...
2 float nota1, nota2, media;
3 int i = 1; // valor inicial do identificador do aluno
4
5 while (i <= 10)
6 {
7 cout << "Entre nota 1 e nota 2 do aluno " << i << ": " <<
    endl;
8 cin << nota1 << nota2;
9 media = (nota1 + nota2) / 2;
10 cout << "A media das notas eh " << media << endl;
11 i++; // aumentamos o valor de i no final de cada calculo da
    media
12 }
13 ...
```

# do-while |

- ▶ Uma variação do comando `while`
- ▶ `do-while` (faça-enquanto ou repita-enquanto)
- ▶ Sua forma geral é dada por:

```
1 do
2 {
3  \\ bloco de comandos
4  \\ a ser repetido
5 }
6 while (<condicao>);
```

## do-while II

### ► Reescrevendo o exemplo:

```
1 ...
2 float nota1, nota2, media;
3 int i = 1; // valor inicial do identificador do aluno
4
5 do
6 {
7 cout << "Entre nota 1 e nota 2 do aluno " << i << ": " <<
      endl;
8 cin << nota1 << nota2;
9 media = (nota1 + nota2) / 2;
10 cout << "A media das notas eh " << media << endl;
11 i++; // aumentamos o valor de i no final de cada calculo da
      media
12 }
13 while (i <= 10);
14 ...
```

### ► Qual a diferença entre while e do-while?

# Exemplos I

▶ Some todos os números pares entre 1 e 999.

▶ `int` some = 2 + 4 + 6 + 8 ...

▶ Ou, utilizando o `while`, teríamos:

```
1 ...
2 int n = 2, // primeiro par maior do que 1
3     soma = 0; // soma inicialmente zerada
4 while (n < 999)
5 {
6     soma = soma + n;
7     n += 2; //Observe o uso de +=
8 }
9 cout << "O valor da soma eh " << soma << endl;
10 ...
```

# Exemplos II

- ▶ Maior dentre 10 números inteiros lidos. Utilizando o `do-while`:

```
1 ...
2 int i = 0, // contador da qtde de numeros lidos
3   maior, n;
4 do
5 {
6   cout << "Entre um numero: ";
7   cin >> n;
8   if (i == 0) // se for o primeiro numero lido
9     {           // ele serah o menor
10      maior = n;
11    }
12   else        // a partir do segundo
13     {
14       if(n > maior) // atualizo o maior
15         {
16           maior = n;
17         }
18     }
19   i++;
```

# Exemplos III

```
20 }  
21 while (i < 10);  
22 ...
```

# Fatorial

Faça uma função que recebe um número inteiro positivo e retorna o fatorial deste número. A função principal deve ler o número do qual se deseja calcular o fatorial e imprimir o resultado.



# Primo

Faça uma função que recebe um número inteiro positivo e retorna `true` se o número for primo ou `false`, caso contrário. A função principal deve ler o número e imprimir o resultado.



# Primo II

Modifique o programa anterior para imprimir todos os números primos abaixo de dois milhões.



# Espelhar

Faça um programa que leia um número inteiro positivo e imprima esse número de trás pra frente. A impressão deve ser feita pela função auxiliar `inverteNumero`.

