



PCS311

Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

Aula 2: Qualidade de Código

Escola Politécnica da Universidade de São Paulo

Qualidade do código

- O que é um *bom código*?
 - Correto
 - Faz o que se espera
 - Eficiente
 - Não desperdiça recursos (memória e processador)
 - Elegante
 - Simples, limpo, bonito e sem enfeites
 - *Testável*
 - Fácil de procurar erros

Qualidade do código

■ O que esse código faz?

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i,j,k;
6      int l[10];
7      cout << "Digite 10 numeros" << endl;
8      for (i=0;i<10;    i++)
9          cin >>l[i]; k= 0;
10         for(j =0;j < 10;j++) {
11             if (l[j]> 0) k +=l[j];}
12
13         cout << "Valor: " << k;
14         return 0;
15     }
```

EX09

Alguns problemas:

- Nome das variáveis
- Indentação
- Espaçamento
- Blocos não claros

Uma outra versão do código

EX10

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int entrada[10];
6      cout << "Digite 10 numeros" << endl;
7
8      for (int i = 0; i < 10; i++)
9          cin >> entrada[i];
10
11     int soma = 0;
12
13     for (int i = 0; i < 10; i++) {
14         if (entrada[i] > 0)
15             soma += entrada[i];
16     }
17
18     cout << "Valor: " << soma;
19     return 0;
20 }
```

Nome de variáveis

- Use nomes representativos
 - Cuidado com abreviações ou nomes longos
 - *Dica*: use “CTRL + ESPAÇO” para autocompletar
- Convenção: *CamelCase*
 - Variáveis e funções
 - Primeira palavra com letra minúscula e as demais com só a 1ª letra em maiúscula
 - *Exemplo*: i, peso, maiorNumero e pedidosAtrasados
 - Nomes de arquivos
 - Cada palavra com a 1ª letra maiúscula e as demais minúsculas
 - *Exemplo*: HelloWorld.cpp, Exemplo1.cpp, ListaLigada.cpp

Organização geral

- Declaração de variável no início do bloco
 - Bloco real (**if**, **for**, etc.) ou *lógico* (organização do código)
- Declare somente a variável controladora do **for** no **for**
 - *Exemplo:*

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



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



Espaçamento

- Use um espaço para separar uma palavra da seguinte
 - =, <=, **while**, **if**, **for** etc. são palavras
- Deixe um espaço depois, mas não antes, de cada sinal de pontuação
 - *Exemplo:* **int** i, j;
- Parênteses
 - Deixe um espaço antes, mas não depois, de abrir um parêntese
 - Deixe um espaço depois, mas não antes, de fechar um parêntese
 - *Exemplo:* **for** (**int** i = 0; i < 5; i++) j++;

Espaçamento

- Exceções

- `x[i]` e não `x [i]`
- `x++` e não `x ++`
- `"."` não é sinal de pontuação em C++!

Chaves (bloco)

- O { deve ficar na mesma linha do comando que define o bloco, com um espaço antes

- *Exemplo*

```
if (x > 5) {
```



```
if (x > 5){
```



```
if (x > 5)  
{
```



```
if (x > 5)  
{
```



- O } deve ficar em uma linha separada, mas alinhado com o comando que definiu o bloco

- *Exemplo*

```
if (x > 5) {  
    // ...  
}
```



Blocos

- O conteúdo de um bloco deve ficar tabulado (um “tab”, normalmente)
 - Deve ser possível diferenciar o que está dentro ou fora do bloco!
 - *Exemplo*

```
for (int i = 0; i < 5; i++) {  
    if (entrada[i] > 5) {  
        cout << entrada[i] << endl;  
    }  
}
```



```
for (int i = 0; i < 5; i++) {  
if (entrada[i] > 5) {  
cout << entrada[i] << endl;  
}}
```



Recomendações

- `if`, `else` e `else if` devem ficar alinhados

```
if (x == 0) {  
    //...  
} else if (x > 0) {  
    //...  
} else {  
    //...  
}
```



- As chaves podem ser omitidas se o bloco tiver apenas uma linha
 - Indente ou deixe na mesma linha o comando

```
if (x > 5)  
    i++;
```



```
if (x > 5) i++;
```



Outras recomendações

- Evite processamento desnecessário
- Evite excesso de otimização
 - *Código ilegível, mas eficiente!*
 - Dificuldade de manutenção
 - (a menos que otimização seja fundamental)
- **O fundamental é ser consistente**

Bibliografia

- FEOFILOFF, P. **Algoritmos em linguagem C**. Editora Campus, 2009. Apêndices A e D.