Padrão de codificação:

Programador: Samuel Bony Alves

Linguagem: Java

Objetivo:

O desenvolvimento desse padrão de codificação, tem por objetivo permitir realizar uma contagem de LOC (Lines Of Code) de programas que possam vir a ser desenvolvidos futuramente.

1.Nomes significativos

1.1-Seja consistente quando for dar nome a funções, tipo, variáveis e constantes.

Exemplo:

int n; // recomendável não usar

int tamanho; // recomendável

1.2-Não utilizar identificadores que contêm dois ou mais underlines em uma linha.

Exemplo:

const int valor_de_x;

1.3-Não Utilizar identificadores começando como por underlines.

Exemplo:

const int _x; // recomendável não usar

1.4 -Variáveis com nome composto devem usar o padrão Camel Case .
Exemplo:
int Contador_particulas; // recomendável não usar

int ContadorParticulas; // recomendável

2. Organização de código:

2.1-Deixe pelo menos uma linha em branco entre cada definição de uma função.

Exemplo:

```
void Function1 ( void f) {
// faz alguma coisa
}


void Function2 (void f) {
// faz alguma coisa.
}
```

2.2-Evite linhas em branco desnecessárias.

Exemplo:

```
for ( int i = 0; i < 10; i++) {// recomendável não usar

if (ehprimo(i)) {


cout<<i;



}
```

2.3-Usar abertura de escopo ‘ { ‘ sempre precedido por um espaço e na linha da declaração no qual está o usando e terminar o escopo ‘ } ’ de forma que ele esteja alinhado com a declaração que o utiliza.

Exemplo:

```
for ( int i = 0; i < 10; i++) {
        if (i % 2 == 0) {
        // faz alguma coisa;
        }
}
```

2.4-Ao usar qualquer declaração de controle de fluxo , **for, if, else, while, dowhile,switchcase,** etc; dê espaço de uma tabulação.

Exemplo:

```
for ( int i = 0; i < 10; i++) {
        if (i % 2 == 0) {
        // faz alguma coisa;
        }
}
```

3. Comentários.

3.1-Cada arquivo deve conter um comentário com uma breve descrição do conteúdo do arquivo.

Exemplo:
// Este arquivo contem isso , isso aquilo, faz isso, etc
// …
3.2-Utilize // para os comentários.

3.3-Cada classe e função deve conter um comentário com uma descrição.

3.4-Após um comentário deixe uma linha em braco.

4. Ciclo de vida de objetos.

4.1-Declarar e inicializar variáveis perto de onde elas são utilizadas.

4.2-Se possível, inicializar variáveis na hora da declaração.

4.3-Sempre que for possível evite declarações de variáveis globais.

4.4-Declare cada variável em uma declaração separada.