



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO RIO GRANDE DO NORTE  
CAMPUS JOÃO CÂMARA

# PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS PADRÃO DE CODIFICAÇÃO

Nickerson Fonseca Ferreira  
[nickerson.ferreira@ifrn.edu.br](mailto:nickerson.ferreira@ifrn.edu.br)

# Convenção de código Java

2

- A Sun, atualmente Oracle, criou uma convenção para padronizar a codificação com Java.
- O uso de uma convenção é importante por:
  - Aproximadamente 80% do custo de um sistema está em sua manutenção;
  - Dificilmente o autor original irá realizar a manutenção de seu código durante o tempo de vida do sistema;
  - A padronização deixa o código mais legível e fácil de ser entendido;

# Convenção de código Java

3

- A convenção trata dos seguintes tópicos:
  - Organização dos arquivos
  - Endentação
  - Comentários
  - Declarações
  - Comandos (if, while, for, etc.)
  - Nomenclatura

# Organização dos arquivos

- Os arquivos devem ser divididos em seções e cada seção separadas por uma linha em branco ou um comentário.
- Evitar arquivos com mais de 2000 linhas.
- Cada arquivo deve conter apenas uma classe ou interface.
- Um arquivo de código deve conter a seguinte estrutura:
  - Comentários iniciais;
  - Declaração do pacote e importação das classes;
  - Declaração da classe ou interface;
  - Declaração dos atributos;
  - Declaração dos construtores;
  - Declaração dos métodos.

# Indentação

5

- A unidade de medida para a indentação são 4 espaços em branco.
- Evitar linhas de código com mais de 80 caracteres.
- Quando uma expressão não couber completa numa linha, quebre-a da seguinte forma:
  - Quebrar após a vírgula;
  - Quebrar antes do operador;
  - Alinhe a linha de baixo com a expressão da linha anterior;

# Indentação

6

```
function(longExpression1, longExpression2, longExpression3,  
        longExpression4, longExpression5);
```

```
var = function1(longExpression1,  
               function2(longExpression2,  
                          longExpression3));
```

```
longName1 = longName2 * (longName3 + longName4 - longName5)  
            + 4 * longname6; // PREFER
```

# Indentação

7

```
if ((condition1 && condition2)
    || (condition3 && condition4)
    ||!(condition5 && condition6)) {
doSomethingAboutIt();
}
```

**NÃO UTILIZAR,  
POIS A LINHA  
DENTRO DO IF  
PODE PASSAR  
DESAPERCEBIDA**

```
if ((condition1 && condition2)
    || (condition3 && condition4)
    ||!(condition5 && condition6)) {
doSomethingAboutIt();
}
```

**UTILIZAR ESTES  
PADRÕES!!**

```
if ((condition1 && condition2) || (condition3 && condition4)
    ||!(condition5 && condition6)) {
doSomethingAboutIt();
}
```

# Comentários

- Os comentários servem para descrever classes, métodos, estruturas de dados ou algoritmos.
- **LEMBREM-SE** :: Quem criou o código foi você, o outro desenvolvedor não sabe o que se passa na sua cabeça. **Deixe documentado.**
- Os comentários podem aparecer da seguinte forma:
  - Blocos de comentários;
  - Linha simples de comentário;
  - Comentário de fim de linha;



# Comentários

9

```
/*  
 * Here is a block comment with some very special  
 * formatting that I want indent(1) to ignore.  
 *  
 *     one  
 *         two  
 *             three  
 */
```



**Comentário  
de bloco**

```
if (condition) {  
    /* Handle the condition. */  
    ...  
}
```



**Linha  
simples**

```
if (foo > 1) {  
    // Do a double-flip.  
    ...  
}  
else  
    return false;           // Explain why here.
```



**Fim de  
linha**

# Declarações

10

- ❑ Cada declaração de variável por linha.
- ❑ Evitar declaração de variáveis com o mesmo nome, mas escopos diferentes.
- ❑ Declarar as variáveis no início do bloco.
- ❑ Tentem inicializar as variáveis locais no momento que declararem.
- ❑ **CUIDADO** com o escopo das variáveis.

```
int level;  
int size;
```

```
int foo, fooarray[];
```

# Declarações

11

```
public class TesteJava {  
  
    int varDeClasse = 0;  
  
    public void metodo () {  
  
        int varDeMetodo = 0;  
        System.out.println(varDeClasse);  
        for (int j = 0; j < 10; j++) {  
            System.out.println(j);  
        }  
        System.out.println(j);  
    }  
  
    public void metodo2 () {  
        System.out.println(varDeClasse);  
        System.out.println(varDeMetodo);  
    }  
}
```



# Comandos

12

- ❑ Um comando em cada linha.
- ❑ Evitar usar os comandos de laço e condição sem as chaves { }

```
argv++;  
argc--;
```

```
argv++; argc--;
```

```
if (condition) {  
    statements;  
}
```

```
if (condition)  
    statement;
```

# Nomenclatura

- Todas as classes e interfaces devem ter nome de substantivos e a primeira letra de cada nome em maiúsculo (ConectorBanco).
- Nome de métodos devem ser verbos e com a primeira letra minúscula e as primeiras das palavras seguintes maiúsculas.
- Variáveis devem ter nomes curtos, porém explicativos.
- Constantes devem ter o nome todo MAIÚSCULO.