

## **Padrão de Codificação**

Programador(a): Daniela Assis  
Linguagem: Java

### **Objetivos**

O desenvolvimento desse padrão de codificação tem o objetivo de permitir realizar uma contagem consistente de LOC (Line Of Code) dos programas a serem desenvolvidos

### **Regras do padrão de codificação**

Todos os arquivos .java desenvolvidos obrigatoriamente contém a declaração de, pelo menos uma classe ou uma interface. Além disso, podem estar presentes outras declarações e comentários.

Basicamente, as classes Java terão:

- 1 – Comentários de início de arquivo
- 2 – Declaração de package
- 3 – Declarações import
- 4 – Declarações de classes e/ou interfaces

Há uma linha em branco entre os itens citados acima

#### **1 – Comentários de início de arquivo**

Esse comentário é delimitado por “/\*” e “\*/” e contém informações técnicas sobre a classe desenvolvida como: nome da classe, autor, versão, data da implementação. Esse comentário contém as seguintes características:

- 1 – O comentário é aberto na primeira linha do arquivo.
- 2 – A primeira informação está contida na linha seguinte a abertura do comentário.
- 3 – O comentário é terminado na linha seguinte à especificação da última informação.

#### **2 – Declaração de package**

Logo após o comentário de início de arquivo há a declaração package caso seja necessária. Há uma linha em branco entre o comentário de início de arquivo e a declaração package.

#### **3 – Declarações de import**

Após a declaração package (caso não haja declaração package, considere após o comentário de início de arquivo) há uma linha em branco e, em seguida, as declarações de import, caso sejam necessárias.

Exemplos:

```
import java.util.*;  
import java.sql.*;  
import javax.sql.*;  
import java.util.LinkedList;  
import java.util.ArrayList;  
import org.daml.grci.daml.*;
```

#### **4 – Declarações de classes e interfaces**

Após as declarações de package e de import (caso não exista nenhuma, considere após o comentário de início de arquivo) há uma linha em branco e, em seguida, as declarações de classes e/ou interfaces.

As declarações de classes ou interfaces consistem nos seguintes itens na ordem especificada:

- 1 – Declaração do cabeçalho da classe ou interface, onde é especificado se a classe é derivada de uma outra classe, ou se a classe implementa uma interface.
- 2 – Variáveis estáticas (static).
- 3 – Declaração dos membros da classe.
- 4 – Construtores da classe.
- 5 – Métodos da classe.

#### 4.1 – Declaração de cabeçalho da classe ou interface.

Após a declaração de import separadas por uma linha em branco. A chave que inicia a delimitação do corpo da classe ou interface está na mesma linha da declaração do cabeçalho.

#### 4.2 – Variáveis estáticas

Há uma linha em branco entre a declaração do cabeçalho e a declaração das variáveis estáticas, caso as mesmas existam. Pode haver um comentário sobre a variável na linha anterior a sua declaração iniciado com “//”.

Cada declaração de variável estática está em uma linha e não há linha em branco entre essas declarações.

Exemplo:

```
public class Exemplo {  
    // comentário  
    static tipo variavel1  
    // comentário  
    static tipo variavel2  
    ...  
}
```

#### 4.3 – Membros de dados

Os membros de uma classe são declarados após as variáveis estáticas (ou após a declaração do cabeçalho caso as variáveis estáticas não estejam presentes). Há uma linha em branco antes da declaração dos membros de dados.

Pode haver um comentário sobre a variável assim como em variáveis estáticas. Cada declaração de membro de dados está em uma linha e não há linha em branco entre as declarações desses membros.

Exemplo:

```
a)  
public class Teste {  
    // comentário  
    private static int staticV;  
    private static int staticV2;  
  
    private int variavel1;  
    // comentário  
    private String variavel2;  
    ...  
}
```

#### 4.4 – Construtores da classe

Há uma linha em branco antes da declaração do construtor e a chave que inicia o seu corpo está na mesma linha da declaração do cabeçalho do construtor. A declaração de um construtor pode conter um comentário de documentação. Os comentários de documentação são delimitados por “/\*\*” e por “\*/”. O texto do comentário começa na linha seguinte à abertura do comentário. E o comentário é terminado na linha seguinte ao fim do texto do comentário. Há uma linha em branco entre a declaração do cabeçalho do construtor e o início da implementação do corpo do construtor. A chave que termina o corpo do construtor está na linha seguinte à última declaração ou comando executável contido no corpo do construtor. Caso haja mais de um construtor em uma classe, haverá uma linha em branco entre os mesmos.

#### 4.5 – Métodos da classe

Os métodos de uma classe são codificados de forma semelhante aos construtores descritos na seção anterior.

#### 4.6 – Método finalize

O método finalize é codificado de modo semelhante aos construtores e aos demais métodos da classe.

### 5 – Declarações e comandos contidos dentro dos métodos e construtores

#### 5.1 – Declaração de variáveis e comandos executáveis

Cada variável declarada ou comando executável está em uma linha separada. As variáveis utilizadas em um método serão declaradas no início do corpo do método, e haverá uma linha em branco após a declaração dessas variáveis separando-as dos demais comandos.

#### 5.2 – Regras gerais para comandos compostos contidos em um bloco delimitado por “{” e “}”

Os comandos compostos são delimitados por “{” e “}”. A chave de abertura está no final da linha que inicia o comando composto e o primeiro comando está na linha seguinte à chave de abertura ( não esqueça que nos métodos e construtores há uma linha em branco entre a declaração do cabeçalho e o início da implementação do corpo do método ). A chave que termina o bloco contendo comandos compostos está na linha seguinte ao último comando. Nas situações em que o uso das chaves for opcional, elas serão utilizadas, a fim de que no futuro, ao necessitar acrescentar um comando adicional em um bloco, não ocorram erros devido ao esquecimento de colocar as chaves.

Os comandos contidos dentro dos blocos de comandos compostos obedecem às mesmas regras para comandos e declarações contidos dentro de métodos e construtores.

##### 5.2.1 – Declarações if, if-else, if else-if else

Exemplos:

```
a)
if ( condição ) {
    ...
}
```

```
b)
if ( condição ) {
    ...
}
else {
    ...
}
```

```

c)
if ( condição ) {
...
}
else if ( condicao ) {
...
}
else {
...
}

```

#### 5.2.2 – As declaração do laço for e while.

A forma de codificação das declarações do laço for e while são semelhantes à declaração if.

Exemplos:

```

a)
for ( int i = 0; i < 10; ++i ) {
    System.out.println( i );
}

```

```

b)

int i = 0;
while ( i < 10 ) {
    System.out.println( i );
}

```

Uma declaração for ou while vazia tem as seguintes formas:

Exemplos:

```

a)
    for ( int i = 0; i < 10; ++i );

```

```

b)
    while ( i < 10 );

```

#### 5.2.4 – A declaração do-while

Exemplo:

```

do {
    System.out.println( i );
} while ( i < 10 );

```

#### 5.2.5 – A declaração switch

Há linha em branco entre os comandos case contidos na declaração de switch.

Exemplo:

```
switch ( i ) {  
    case 10:  
        System.out.println( i );  
  
    case 20:  
        System.out.println( i );  
  
    default:  
        break;  
}
```

#### 5.2.6 – Declarações de try-catch

Não há linha em branco entre o bloco try, os blocos catch e o bloco finally. Um bloco começa na linha seguinte a chave que fecha o bloco anterior.

Exemplo:

```
try {  
    String resposta = clientPronto.getClassDef( "caféina" );  
}  
catch ( PRontoException e ) {  
    System.out.println( e.getException() );  
}  
finally {  
    clientPronto.closeConnection();  
}
```

#### 5.3 – Operador ternário

O operador ternário está contido em uma única linha:

Exemplo:

```
i >= 0 ? i : -i;
```

#### 5.4 – Comentários

Os comentários são precedidos de uma linha em branco e estão na linha imediatamente anterior à declaração ou comando ao qual o comentário se refere. Os comentários de bloco contêm o texto do comentário na linha seguinte a abertura do comentário e o comentário é terminado na linha seguinte ao término do texto.

Exemplo:

```
a)  
...  
int a = 5 + 8;  
/*  
 * Realiza a contagem até o valor de a.  
 */  
for ( int i = 0; i < a; ++i ) {  
    ...
```

```
}  
...
```

b)

```
...  
int a = 5 + 8;  
// Realiza a contagem até o valor de a.  
for ( int i = 0; i < a; ++i ) {  
    ...  
}  
...
```

### 5.5 – Comando return

O comando return será precedido por uma linha em branco.

Exemplo:

```
public int getNumero() {  
    ...  
    int x = 4;  
  
    return x;  
}
```