

1ª Aula: Instruções e Estruturas de Controle

REGRAS PARA A CODIFICAÇÃO EM JAVA

1. Não colocar mais do que uma instrução por linha.
2. O numero de caracteres por linha não deve exceder os 70. Em caso de necessidade uma instrução pode espalhar-se por várias linhas.
3. O nome das classes deve começar por maiúscula. As letras seguintes devem ser minúsculas, exceto se coincidirem com o início de novas palavras. Evitar o uso de abreviaturas. Não utilizar traços de união.

Bons Exemplos	Maus Exemplos
Pessoa	Pessoa
CidadeSobral	PESSOA
	Pessoa
	Cidade_Sobral
	Cidadesobral

4. Os nomes dos métodos devem ser verbos e começar com letras minúsculas. As letras seguintes devem também ser minúsculas, exceto se coincidirem com o início de novas palavras. Não utilizar traços de união.

Bons Exemplos	Maus Exemplos
getNome	Getnome
setNome	GetNome
correr	Get_nome
	CORRER

5. Os nomes das variáveis devem começar com letras minúsculas. As letras seguintes devem também ser minúsculas, exceto se coincidirem com o início de novas palavras. Não utilizar traços de união. Os nomes das variáveis devem ser curtos mas com significado. Nomes de variáveis com uma única letra devem ser evitados a não ser que se tratem de variáveis de caractere temporário.

Bons Exemplos	Maus Exemplos
nome	Nome
areaCasa	valOR
	areacasa
	AreaCasa

6. As constantes (variáveis finais) devem ter nomes formados apenas por letras maiúsculas devendo utilizar-se traços de união para separar as palavras.

Bons Exemplos	Maus Exemplos
PI	Pi
LIMITE	limite
MAX_LIMITE	MAXLIMITE

7. Deve haver consistência na colocação das chaves, na *indentação** e nos espaços. Em caso de dúvida consultar as convenções disponibilizadas em: <http://java.sun.com/docs/codeconv/>

Exemplos:

Estrutura *if*:

```
if (condição) {
    instruções;
}
```

Estrutura *if-else*:

```
if (condição) {
    instruções;
} else {
    instruções;
}
```

Estrutura *for*:

```
for (inicialização; condição; incremento) {
    instruções;
}
```

Estrutura *while*:

```
while (condição) {
    instruções;
}
```

Estrutura *do-while*:

```
do {
    instruções;
} while (condição);
```

Estrutura *switch*:

```
switch (condição) {
case ABC:
    instruções;
    /* falls through */
case DEF:
```

```
        instruções;
        break;
    case XYZ:
        instruções;
        break;
    default:
        instruções;
        break;
}
```

Estrutura *try-catch-finally*:

```
try {
    instruções;
} catch (ExceptionClass e) {
    instruções;
} finally {
    instruções;
}
```

Exemplo geral:

```
/*
 *
 * (c) UVA - CComputação - POO 2009.1
 *
 */
import java.util.Vector;
public class ExemploCodificacao {
    // variaveis de instancia
    private int numero;
    private String nome;
    private Vector notas;
    // construtores
    public ExemploCodificacao() {
        numero = 0;
        nome = "";
        notas = new Vector(0);
    }
    public ExemploCodificacao(String s) {
        numero = 0;
        nome = s;
        notas = new Vector(0);
    }
}
```

```
}  
// métodos seletores e modificadores  
public void setNome(String s) {  
    nome = s;  
}  
public String getNome() {  
    return nome;  
}  
// método para adicionar um elemento ao vetor notas;  
public void adicionaNota(int x) {  
    Integer i = new Integer(x);  
    notas.add(i);  
}  
//método para percorrer o vetor ...  
public int somaNotas() {  
    Integer k;  
    int soma = 0;  
    int j = 0;  
    while(j <= notas.size() ){  
        k = (Integer)notas.elementAt(j);  
        soma = soma + k.intValue();  
        j++;  
    }  
    return soma;  
}  
}
```