Guia de estilo
Boas práticas de programação[1]

1 Atribuição de nomes

1.1 Utilize nomes significativos

Utilize nomes significativos para todos os identificadores (nomes de classes, variáveis e métodos). Evite ambiguidades. Evite abreviaturas. Métodos modificadores simples deem ser nomeados *setAlgo(...)*. Métodos de acesso simpels devem ser nomeados *getAlgo(...)*. Métodos de acesso com valores de retorno booleanos muitas vezes são chamados *eAlgo(...)* - por exemplo *eVazio()*.

1.2 Nomes de classes começam com letra maiúscula

1.3 Nomes de classes são substantivos singulares

1.4 Nomes de métodos e variáveis começam com letra minúsculas

Os nomes de classes, métodos e variáveis utilizam a letra maiúscula no meio para aprimorar a legibilidade dos identificadores compostos, por exemplo, *numeroItens*.

1.5 Constantes são escritas com LETRAS MAIÚSCULAS

Constantes ocasionalmente utilizam sublinhados pra indicar os identificadores compostos: *_MAXIMUM_SIZE*.

2 Layout

2.1 Um nível de recuo costuma ter quatro espaços, ou uma tabulação

2.2 Todas as instruções dentro de um bloco são recuadas em um nível

```
if(status){
    System.out.println("Cadastro aberto.");
}else{
    System.out.println("Cadastro fechado.");
}
```

[1]Fonte: Barnes, Kolling, Programação orientada a objetos com Java, Apendice J (adaptado), 4ª ed. Editora Pearson.

2.3 Sempre utilize chaves nas estruturas de controle

Se um bloco contém somente um comando, o uso das chaves é opcional, mas fica mais legível a utilização das chaves em condicionais e repetições mesmo nesses casos.

2.4 Utilize um espaço antes dos operadores

```
resultado = aumento + saldo;
```

3 Documentação

3.1 Cada classe tem um comentário de classe no topo

O comentário de classe contém pelo menos:

- uma descrição geral da classe;
- o(s) nome(s) do(s) autor(es);
- um número de versão.

Cada pessoa que contribuiu à classe tem de ser identificada como um autor ou do contrário receber créditos apropriados.

Um número de versão pode ser um número simples, uma data ou outro formato. O importante é que um leitor precisa ser capaz de reconhecer que as duas versões não são idênticas e de determinar qual é a mais recente.

3.2 Cada método tem um comentário de método

3.3 Comentários são legíveis ao javadoc

Comentários de classe e métodos dem ser reconhecidos pelo javadoc. Eles devem começar com o símbolo de comentário `/**` e encerrar com `*/`.

3.4 Comentário de código (apenas) onde é necessário

Comentários no código devem ser incluídos onde o código não é óbvio ou difícil de entender, e onde ele ajuda a entender um método. Não comente instruções óbvias - suponha que o leitor entende o Java.

4 Restrições de uso da linguagem

4.1 Ordem das declarações: atributos, construtores, métodos

Os elementos de uma definição de classe aparecem (se presentes) nesta ordem: instrução de pacotes, instruções de importação; comentário de classe; cabeçalho de classe; definições de atributos, construtores; métodos.

4.2 Sempre utilize modificadores de acesso

Especifique todos os atributos e métodos como privados, públicos ou protegidos. Nunca utilize acesso padrão (pacote privado).

4.3 Importe classes separadamente

Instruções de importação que explicitamente nomeiam cada classe são preferíveis à importação de pacotes inteiros. Por exemplo:

```
import java.util.ArrayList;
import java.util.HashSet;
```

é melhor do que:

```
import java.util.*;
```

4.4 Sempre inclua um construtor (mesmo se o corpo estiver vazio)

4.5 Inicialize todos os atributos do construtor