



USP - Universidade
de São Paulo



IME - Instituto de
Matemática e Estatística

MAC0332
Engenharia de Software

Implementação

Marco Aurélio Gerosa
gerosa@ime.usp.br



Convenção de código

- O estilo de escrita de código deve ser padronizado?
- Em um projeto? Em uma instituição? Para uma linguagem?
Para todas as linguagens?



Convenções de código

- Java -
<http://java.sun.com/docs/codeconv/>
:
- “Why have code conventions?
Code conventions are important to programmers for a number of reasons:
 - 80% of the lifetime cost of a piece of software goes to maintenance.
 - Hardly any software is maintained for its whole life by the original author.
 - Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly. “
- XHTML

```
/*
 * Copyright (c) 1993-1996 Sun Microsystems, Inc. All Rights Reserved.
 *
 */
package java.blah;

import java.blah.blahdy.BlahBlah;

/**
 * Class description goes here.
 *
 * @version 1.10 04 Oct 1996
 * @author Firstname Lastname
 */
public class Blah extends SomeClass {
    /* A class implementation comment can go here. */

    /** classVar1 documentation comment */
    public static int classVar1;

    /**
     * classVar2 documentation comment that happens to be
     * more than one line long
     */
    private static Object classVar2;

    /** instanceVar1 documentation comment */
    public Object instanceVar1;

    /** instanceVar2 documentation comment */
    protected int instanceVar2;

    /** instanceVar3 documentation comment */
    private Object[] instanceVar3;

    /**
     * ...method Blah documentation comment...
     */
    public Blah() {
        // ...implementation goes here...
    }
}
```



A entrega final dessa 1ª etapa e o projeto da disciplina devem seguir fielmente o Java Code Convention e as restrições do XHTML 1.1



Notação Húngara

- O tipo da variável é identificado em seu nome, classes e interfaces são diferenciadas. Ex:

```
01. public interface IMinhaInterface {
02.     public void meuMetodoAbstrato();
03. }
04.
05. public class CMinhaClasse {
06.
07.     protected Tipo1 m_atributo1;
08.     protected Tipo2 m_atributo2;
09.
10.     public void meuMetodo(Tipo1 _argumento1, Tipo2 _argumento2) {
11.         Tipo2 variavelLocal1;
12.         Tipo3 variavelLocal2;
13.
14.         m_atributo1 = _argumento1;
15.         m_atributo2 = _argumento2;
16.     }
17.
18. }
```

- Deve ou não ser usada? Depende da linguagem (ela foi criada para C/C++)?



Práticas de programação

- Em qual língua escrever o código (comentários, variáveis etc.)?
- Uso de nomes de variáveis significativos
 - coordenadaXDaPosicaoDoBracoDoRobo ou coordX?
- Uso consistente de nomes de variáveis
 - Uso de termo Ex: freqMedia, maximoDaFrequencia, frMin, frqncTotal
 - Formação da variáveis Ex: maximoDaFrequencia, frequenciaMinima
- Código autodocumentado ou uso de comentários
- Comentários preliminares (no início do arquivo)
- Identação e emprego de linhas em branco

```
if (latitude > 30 && longitude > 120) if (latitude <= 60 && longitude <= 150)
mapSquareNo = 1; else if (latitude <= 90 && longitude <= 150) mapSquareNo = 2
else print "Not on the map"; else print "Not on the map";
```

- Uso de convenções e métricas que possam ser verificados automaticamente



Práticas de Programação

- Geração de código
- Uso de uma ferramenta de controle de versão
- Princípio DRY (Don't Repeat Yourself)
- Convenção sobre configuração