

Code Convention for Java

1 Introduction

This document is based on recommendations from SUN and individual experience. It is important to be able to write readable source code independent of any development environment.

2 Files

Each java source file should contain only **one** class or interface, except for private classes/interfaces declared within a public class.

Braces should be placed vertically aligned.

Example:

```
public class testClass
{
}

```

All files should have the following layout:

1. Package and import statements.
2. Class/Interface declaration.

Class/Interface declaration

Order of appearance	Part of class/interface declaration	Notes
1	Class/Interface comment	See comments
2	<code>class</code> or <code>interface</code> statement	
3	Static variables	public, protected, private
4	Instance variables	public, protected, private
5	Constructors	
6	Methods	

3 Indentation

Indentation should be set to 4.

Wrapping Lines

When an expression will not fit on a single line, break it according to these principles:

- Break after comma
- Break before operator

Example:

```
doSomeStuff(int firstArgument, float secondArgument,
            double thirdArgument);

```

4 Comments

Javadoc comments should be used through out the project. For more information on javadoc, go to <http://java.sun.com/javadoc>.

5 Declarations

Declarations should be written on separate lines. Use tab between identifier and type. Use tab between '=' and initialisation.

Example:

```
int    intValue    = 2;
float  floatValue  = 2.2;
double doubleValue = 2.3;
```

6 Statements

if, if-else, else if statement

The if, if-else and else if should have the following form:

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

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

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

for statement

The for statement should have the following form:

```
for(initialisation; condition; update)
{
    statements;
}
```

while, do-while statements

The while statement should have the following form:

```
while(condition)
{
    statements;
}

do
{
} while(condition);
```

switch statements

The switch statement should have the following form:

```
switch(condition)
{
case ABC:
    statements;
    break;

case XYZ:
    statements;
    break;
}
```

7 Naming conventions

Packages

Package names should be unique and in lower case.

Example:

```
package se.kth.hydra.km.servlet;
package se.kth.hydra.ui.ejb;
```

Classes

Class names should be in mixed case with the first letter of each internal word as uppercase.

Example:

```
class DataBase;
class ServiceProvider;
```

Interfaces

Interface names should be declared in the same way as classes.

Methods

Method names should start with lowercase. The first letter of each internal word should be capitalized.

Example:

```
getNumber();
setColor();
run();
```

Variables

Variable names should be declared in the same way as methods.

Example:

```
int    maxConnections;
float  price;
```

Constants

Constants should be declared in uppercase.

Example:

```
final int MAX_CONNECTION = 5;
final int MIN_WIDTH = 20;
```

8 Example Code

```
package se.kth.hydra.test;

import java.whatever.stuff.Something;

/**
 * Class description...
 *
 * @version    1.82
 * @author    FirstName LastName
 */
public class testClass extends someClass
{
    /**
     * classVariable1 documentation
     * multiple line comment
     */
    public static int classVariable1;
    /** classVariable3 documentation comment */
}
```

```
private static float classVariable3;

/** instanceVariable1 documentation comment */
public int instanceVariable1;

/** instanceVariable2 documentation comment */
protected int[] instanceVariable2;

/** instanceVariable3 documentation comment */
private float instanceVariable3;

/**
 * Constructor documentation comment for testClass()
 */
public testClass()
{
}

/**
 * getUser documentation comment...
 * @param userId description
 */
public User getUser(int userId)
{
    ...
}
}
```