

# Code Conventions v0.3

2015.11.12

This document is still under review and will change later on.

## Inhaltsverzeichnis

	Seite
<b>1 Tabs</b> . . . . .	<b>2</b>
<b>2 Line length</b> . . . . .	<b>2</b>
2.1 Breaking function parameters . . . . .	2
2.2 Breaking if's . . . . .	2
<b>3 Getters and Setters</b> . . . . .	<b>2</b>
<b>4 Function Naming</b> . . . . .	<b>2</b>
<b>5 Variable Naming</b> . . . . .	<b>2</b>
5.1 Example . . . . .	3
<b>6 Spacing</b> . . . . .	<b>3</b>
6.1 Space after if, for and while . . . . .	3
6.2 Space between ) and { . . . . .	3
6.3 Space in for loop definitions . . . . .	4
6.4 One Statement in one Line . . . . .	4
6.5 K&R Style tabs and parentheses . . . . .	4
<b>7 Function Parameters</b> . . . . .	<b>4</b>
7.1 Ordering . . . . .	4
7.2 All parameters are constant . . . . .	4
<b>8 Errors</b> . . . . .	<b>4</b>
<b>9 No ternary expressions</b> . . . . .	<b>4</b>
<b>10 Fully Qualify Imports</b> . . . . .	<b>5</b>

## 1 Tabs

This is one of points where teams often fight about. We will use a tab size of 4 spaces.

## 2 Line length

Lines should not get longer than 140 characters (TODO: verify this)

### 2.1 Breaking function parameters

if they get too long, break like this:

```
private static synchronized horkingLongMethodName(  
    int anArg,  
    Object anotherArg,  
    String yetAnotherArg,  
    Object andStillAnother  
){  
    code();  
}
```

### 2.2 Breaking if's

```
if( (condition1 && condition2)  
    || (condition3 && condition4)  
    ||!(condition5 && condition6)  
){  
    doSomethingAboutIt();  
}
```

## 3 Getters and Setters

All variables are private!

Getters and setters will be used for accessing and setting the variables, if needed.

## 4 Function Naming

Marc will assign names to Functions.

## 5 Variable Naming

**camelCase**

Variables start small and then every new Word is written big.

## Hungarian Notation

We will use specifiers defining the Type of the variables. These will be mostly be one letter long and stand in front of the name.

If a type of Object is used only once or twice it has the specifier o, else it got to have something else. This will help differentiating the Object types.

## Constants

written in big LETTERS and underscores split the words.

type	specifier
integer	i
unsigned integer	ui
boolean	b
string	s
float	f
double	d
object <sup>1</sup>	o
array	a<data type>
Character	chr
textfield	txt
textarea	txa
Button	btn
Menu	mnu
label	lbl

## 5.1 Example

iLoop for an integer which is used in loops

sName for an String which holds a name

# 6 Spacing

(TODO: discuss about this)

## 6.1 Space after if, for and while

no space after if, for or while and the opening parentheses.

## 6.2 Space between ) and {

no space or newline between ) and {.

---

<sup>1</sup>If only one instance is used

### 6.3 Space in for loop definitions

```
for(int i = 2; i<1; i--){  
    ...  
}
```

### 6.4 One Statement in one Line

After each Statement insert one newline and tabs.

### 6.5 K&R Style tabs and parentheses

K&R Style tabs and parentheses:

```
if(x < 0){  
    System.out.println("Negative");  
    negative(x);  
}else{  
    System.out.println("Non-negative");  
    nonnegative(x);  
}
```

## 7 Function Parameters

(TODO: discuss about this)

### 7.1 Ordering

From the most common variables to the ones that are changing most often.

### 7.2 All parameters are constant

The Parameters given to a function should be constant (final).

The return statement is used for retuning stuff!

## 8 Errors

If the function has enough knowledge to prevent errors, it should do so!

If not then it should throw the Error to the caller.

This should escalate till some function handles the Error. If the Error can not be handled then the Program should print the Error and exit (System.exit(1)).

(TODO: use Common function for error printing and exit)

## 9 No ternary expressions

This one should be simple.

## 10 Fully Qualify Imports

no stars in imports!