Abstract

This document gives you an overview how to appropriately write our documentation. You should follow these instructions slavishly to ensure that nobody messes up while programming.

General style suggestions

In order to not get eye cancer of looking at the documentation you should follow some styleguidelines. Use for headings Words predefined headings. Start with Heading 1 then Heading 2 and so forth, for text also use Words predifined style.

Table of contents

In the real documentation, there will be a table of contents. You always have to update the table of contents.

Classes

After you wrote new classes use their names as headings. Then, write who is the author(s) of the class. Subsequently, write about the class itslef and describe the variables of the class.

Methods

Not every method must be documented we only have to document those where their complexity forces us to. This means methods such as getters and setters do often just contain one or two lines of code where no big logical problems get solved. When documenting methods you write about the general porpuse of the methods. After that, describe the signature of the methods. Then, describe their parameters, their return values and their expception cases.

Example

SampleJointClass

This classes porpuse is to demonstrate to you the coding convention of the company captainAndroid. It is a class which was written by mutliple people namly, C. P. Wutti, P. L. Lagger, J. Weiler, S. Sarkovic.

Vars

This class has two vars namely,

```
private static final int SAMPLECONST = 1000000;
private String sampleMessage = null;
```

where SAMPLECONST is a constant which is used to calculate something. Where sampleMessage is a String which is a dummy variable.

Calculate HighlyComplicatedMethods

```
public int calculateHighlyComplicatedMethod(int prime, int notPrime) throws
Exception{
```

This method computes a highly complicated mathematical formula. It takes two variables, prime and not prime which are both int. It produces an integer. It throws the Exception: “1 is not a prime” if the parameter “prime” is equal to 1. It throws the Exception: “2 is a prime” if the parameter “notPrime” is equal to 2.

Example repository: https://github.com/c-p-wutti/SampleDocumentationRepV2