My Code Standard

Comments
Methods, classes and important statements must have comments saying WHY something is being done. Both multiline /* */ and single line comments are allowed //

Easy understandable code instead of optimizations
Code should be written for humans to read - i.e. no optimizations, that will make the code more difficult to read. The optimizations can always come at a later point, when the basics of the program are understood and implemented.

Divide and conquer
Classes/Methods should be divided into smaller classes/methods, if a method grows larger than 30 lines.

Names - Variable, constants and classes.
Variables, constants and type names should be meaningful (i.e. not short names - only if you are doing a mathematical program, where x, y and z for instance are needed, it is allowed). Also camelCase should be used for both types and variables. Constants should be noted with UPPERCASE and spaces in the constants should be underscores for instance SOME_CONSTANT. Also constants should be declared with static final.

Loops.
If something is always being done at least one time in a loop no matter what, a do-while loop should be used.
Also if we don't know how much we're going to iterate over in a loop, a while loop should be used.
And then if we KNOW how much we are going to iterate, a for loop should be used.
If we are iterating over a collection or something iterable a for each loop should be used.

Single line statements.
Single line statements after if should always be in curly brackets (i.e. no single line statement)

Composition over polymorphism
Always use comoposition over polymorphism, unless the polymorphism is needed for some cool design pattern. This is to ensure easier code reuse.

Tabs over spaces.
Never use spaces for indents but tabs :)

Indents

```
always use {
}

instead of
{
}
```