

Software Development II – Code Formatting

Markus Roggenbach

May 2017



Code conventions

Twice the same – in different formatting

```
public static String printPolynomial (int[] p) {
    if (p == null)
        return "0";
    else {
        String s = "";
        for(int i=0;i<=p.length-1;i=i+1){
            s = s + p[i] + " * x^" + (p.length-i-1);
        }
        ...
    }
}

public static String printPolynomial(int[] p) {
    if (p == null) {
        return "0";
    } else {
        String s = "";
        for (int i = 0; i <= p.length - 1; i++) {
            s = s + p[i] + " * x^" + (p.length - i - 1);
        }
        ...
    }
}
```

Every company defines its own style :-)

- JavaRanch – a friendly place for Java greenhorns

<http://www.javaranch.com/style.jsp>

- GeoSoft – Geotechnical Software Services

<http://geosoft.no/development/javastyle.html>

- . . .

here:

Sun Code Conventions

[http://www.oracle.com/technetwork/java/
codeconventions-150003.pdf](http://www.oracle.com/technetwork/java/codeconventions-150003.pdf)

“Order within a file”

Java source files have the following ordering:

- Beginning comments, e.g.

```
/*  
 * Classname  
 *  
 * Version info  
 *  
 * Copyright notice  
 */
```

- Import statements, e.g. `import java.awt.peer.CanvasPeer;`
- Class declarations

“Order within a class”

1. Class documentation comment (`/**...*/`)
2. Class statement
3. Class (static) variables:
First public, then protected, then private.
4. Constructors
5. Methods

Indentation/Line length

Indentation Four spaces – unit of indentation.

Tabs or white-space are both allowed.

MR's comment: tabs are devilish.

Line Length Avoid lines longer than 80 characters.

Declarations

Number Per Line One declaration per line.

```
int level;      // indentation  
level int size; // size of table
```

is preferred over

```
int level, size;
```


Placement of declarations

Put declarations only at the beginning of blocks.

A block is any code surrounded by curly braces “{“ and “}”.

```
void MyMethod() {  
    int int1;           // beginning of method block  
    if (condition) {  
        int int2;       // beginning of "if" block  
        ...  
    }  
}
```

Java classes

- No space between a method name and the parenthesis “(” starting its parameter list.
- Open brace “{ “ appears at the end of the same line as the declaration statement.
- Closing brace “}” starts a line by itself indented to match its corresponding opening statement.
- Methods are separated by a blank line.

An example

```
class Sample extends Object {  
    int ivar1;  
    int ivar2;  
    Sample(int i, int j) {  
        ivar1 = i;  
        ivar2 = j;  
    }  
  
    int emptyMethod() {}  
    ...  
}
```

if, if-else, if-else-if-else Statements

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

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

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

Loops

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

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

```
do {  
    statements;  
} while (condition);
```

Spaces

- A keyword followed by a parenthesis should be separated by a space.

```
while (true) {  
    ...  
}
```

But no blank space between method name and its opening parenthesis.

- A blank space after commas in argument lists.

- All binary operators except “.” should be separated from their operands by spaces. Blank spaces should never separate unary operators such as
 - unary minus,
 - increment (++), and
 - decrement (--)from their operands.

```
a += c + d;  
a = (a + b) / (c * d);  
while (d++ = s++) {  
    n++;  
}  
prints("size is " + foo + "\n");
```

Tool Checkstyle

Checkstyle is a development tool to help programmers write Java code that adheres to a coding standard. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard.

<http://checkstyle.sourceforge.net/>

Tool Demo