Professional Issues II
Unit 1: code commenting
The idea in SE context – part 2

Markus Roggenbach

February 2020

Communication

In case you have questions:

$\rightsquigarrow$ post them on the Discussion Board on Blackboard.

If you think it's a really 'special' matter:

$\rightsquigarrow$ M.Roggenbach@swansea.ac.uk

Note: you can't expect responses out of work time
(Mo – Fr, 9am – 5pm)

You will learn

- How guidelines can be written
- That programming languages come with a language definition, including a ‘grammar’
- How to comment within methods

How to write guidelines

Presentation of Guidelines

Coding rules often are "soft" / rule of thumb / . . .

Seminal example of a guideline of "good design":

- C Alexander, S Ishikawa: *A Pattern Language*.
  Oxford University Press, 1977

Alexander's scheme:

Rule . . . **Example** . . . **Discussion** . . . **Link to related rules** . . .

Commenting along the program structure

1. variable declarations
2. branching structures (if-, switch-statements)
3. loops (while-, do-, for-statement)
4. methods declarations (already discussed)

   $\leadsto$ require commenting

Def: Context free grammar

A context-free grammar consists of a number of productions. Each production has an abstract symbol called a nonterminal as its left-hand side, and a sequence of one or more nonterminal and terminal symbols as its right-hand side. For each grammar, the terminal symbols are drawn from a specified alphabet.

Starting from a sentence consisting of a single distinguished nonterminal, called the goal symbol, a given context-free grammar specifies a language, namely, the set of possible sequences of terminal symbols that can result from repeatedly replacing any nonterminal in the sequence with a right-hand side of a production for which the nonterminal is the left-hand side.

The syntax {x} on the right-hand side of a production denotes zero or more occurrences of x.

from: The Java® Language Specification, available at
https://docs.oracle.com/javase/specs/jls/se8/html/index.html

Excerpt from the Java Grammar (modified by MR)

```
BlockStatements:
  BlockStatement {BlockStatement}

BlockStatement:
  LocalVariableDeclarationStatement
  Statement

Statement:
  Assignment
  IfThenStatement
  ForStatement
  MethodInvocation
  BlockStatements
```

```
ForStatement:
  for ( [ForInit] ; [Expression] ; [ForUpdate] ) Statement

IfThenStatement:
  if ( Expression ) Statement
```

from: The Java® Language Specification, available at
https://docs.oracle.com/javase/specs/jls/se8/html/index.html

A sample method

```
public static void bubble_srt(int array[]) {
    int n = array.length;
    int k;
    for (int m = n; m >= 0; m--) {
        for (int i = 0; i < n - 1; i++) {
            k = i + 1;
            if (array[i] > array[k]) {
                swapNumbers(i, k, array);
            }
        }
        printNumbers(array);
    }
}
```

code taken from http://www.java2novice.com/java-sorting-algorithms/bubble-sort/

MR's Commenting Standard

Rule: Variable declarations

Explain for each variable
what it represents
in the context of the program.

Example:

```
[] int p      /* polynomial */
int    i,j    /* counters   */
```

Rule: Branching structures

State the purpose of the whole branching structure, explain what happens in each of the cases.

Example:

```
/* add polynomials only if they are of the same degree */
if degree(p) = degree(q)
then   /* add the polynomials */
  {...}
else  /* error case */
  {...}
}
```

Rule: Loops

State the purpose of the whole loop structure,
explain how the computation takes place,
discuss termination of the loop.

Example:

```
/* evaluate the polynomial p                                    */
/* implements the Horner Scheme                                 */
/* terminates as k is not touched in the loop body and          */
/*              k is initialised to a positive value            */
b = p[n];
for (k = degree(p);  k >=0; k = k - 1)
  {...}
{
```

Rule: method declarations

see previous slides

Rule: Over-commenting

Avoid trivial comments.

Example:

```
for (k = degree(p);  k >=0; k = k - 1)
    /* k is a counter variable */
  {...}
```

What you have learned in this unit

Definitions

- Context Free Grammar

You should be able to explain by example

- How to derive a program from the Java grammar.
- How guidelines in the style of Alexander look like.
- How to comment a program along its structure.