Short Advice for Documenting Java Programs

Paul Fischer, October 7, 2002

There are three parts of program documentation:

- The *inline documentation* usually in the form of comments in the program code explaining the meaning of variables and subroutines (also called methods, functions, or procedures depending on the programming language).
- The *programmer's manual* in the form of a report or manual describing the high level structure of the program, the design decisions made, the interaction of the different components, and also says what the program does. It should also document the tests made.
- The *user's manual* in the form of a manual explains how to use the program to those who apply it (end-users).

1 Why Program Documentation

During program development:

1. To avoid making unnecessary programming errors.
2. To find errors faster.
3. To make testing for correctness easier.

During program life cycle:

1. To make the program easier to maintain.
2. To explain to others how to use the program.

2 Inline Documentation

These are comments in the source code of the program. The different points below are not completely independent.

1. At the beginning of every program file write the name of the author(s) and what the program does.
2. Before every class and subroutine (method in JAVA) write a comment explaining what it does.

3. Explain what the arguments of a constructor or a subroutine mean.

4. In lengthy code add a comment when an important part is over or begins, like

```
// We finished the initialization of the variables.
// Now we start reading the data from the input file.
```

5. If you use a trick that is not standard give a short comment.

6. Use names for variables which explain their meaning or use a comment: The names `numberOfIntervals` or `inputFileName` are good. If you use short names like `n` or `ifn` instead you should add a comment when the variable is declared:

```
int n; // n is the number of intervals
String ifn; // ifn is the name of the input file
```

7. Use indentation to help the reader to identify the beginning and end of program structures such as loops and branching statements. For Example:

```
1    for(int i = 0; i < n; i++)
2    {
3        for(int j = 0; j < n; j++)
4        {
5           if ( distance[i][j] > 1.234)
6           {
7              System.out.println("Too far!!");
8           }
9           else
10          {
11             System.out.println("Near enough.");
12           }//ifelse
13       }//for j
14   }//for i
```

Remark: The Java environment offers a tool called *javadoc* which helps to generate documentations. The tool is, however, a little difficult to use and you should only consider to use it later in the course when the programs get larger.

3 Programmer's Manual

The different points are not completely independent.

1. Specify who wrote the program and what it is supposed to do.
2. Indicate why the program solves the problem it is supposed to solve.
3. Document the structure: What are the modules (classes) you defined. Why did you design the program this way.
4. Describe what the modules do.
5. Describe the dependencies between the modules (the class dependency).
6. Describe the tests you used and document the results of test runs.
7. Use line numbering in listings so you can refer to a certain part of the program.

4 User's Manual

1. Explain how to use the program. This includes to specify what a correct input is and what the corresponding output is.
2. Explain the error messages of the program and tell the user what to do if one appears.
3. The user's manual should not contain program listings and implementation details.