Coding Style of C Program from the Perspective of Software Engineering

IOP Conf. Series: Materials Science and Engineering **563** (2019) 052073 doi:10.1088/1757-899X/563/5/052073

Coding Style of C Program from the Perspective of Software Engineering

Jiujiu Yu*, Jishan Zhang, Ning Wu, Chun Zheng and Deqing Zhang

College of Computer Engineering, Anhui SanLian University, Hefei 230601, China

*Corresponding author: Email: yjjyjL@163.com

Abstract. Coding style is an important factor in measuring the program quality. The paper explores on the C program coding style of the program annotation writing, data declaration, program statement writing, and input/output design style from the perspective of software engineering. Good coding style plays an important role in improving the readability, testability and maintainability of the program.

1. Introduction

C language programming is a compulsory and fundamental course for science and engineering specialties in most of the domestic universities. As an introductory course, the core task of C language is to master basic grammar and algorithms, and to cultivate abilities on programming and debugging [1]. In actual teaching process, teachers often pay more attention on the correctness of the grammar and debugging result of a C program, but the problem of coding style or standard is ignored. However, from the perspective of software engineering, good coding style and standard are important to improve the testability, maintainability, and reusability of the program quality.

Although the coding style does not belong to the category of program grammar, they are still an important part of content on programming in the actual project development of IT enterprises. Many software companies revise the corresponding coding style according the actual software development environment. Such as coding requires for easy communication and maintenance, none conflict with popular habits, beautiful code arrangement, the logical structure of the code is clear and easy to understand [2].In this paper, as the case of C program design, we try to explore some coding style and train the habit of writing high-quality code from the perspective of software engineering.

2. Program annotation writing

Annotation is a very important part of the C source program, and it helps to understand the source program by developers or coders. The annotation language must be accurate, understandable, and concise. The content of the annotation needs to be clear, accurate, and have no ambiguous [3]. Of course, the amount of annotation should not be too much or too little within one source program. Usually, the amount of valid annotation for a source program must be no less than 20 %.

Preamble annotation and functionality annotation are two kinds of annotation which are mentioned in many books of software engineering. Here, we describe the annotation writing style by example.

2.1. Preamble annotation

Preamble annotation refers to the description at the beginning of each program or module, which plays a role of helping in understanding of the program. It generally includes: program name and version

number, program functional description, interface description, input/output data description, development history, other information related to the development environment, and so on [4].

Preamble annotation is used in description of modules, files (header files) and functions. Some examples are the following:

2.1.1. Module annotation

Module annotation contains module name, built date, author, functional description, version number, modify log, and so on [3].The style of module annotation writing is shown as Figure 1.

```
/*********************************
Module Name:
Module Date:
Module Author:
Description:
Others:
Revision History:
Rel Date:
Ver Notes:
*********************************/
```

Figure 1. Style of module annotation writing

2.1.2. file annotation

The file annotation mainly contains the description of the main functions which are completed by the file, the description of the list of functions that contained in the file, macro definition, and so on [5]. The style of file annotation writing is shown as Figure 2.

```
/*****************************
File Name:
File Date:
File Author:
Description:
Function List:
Others:
Revision History:
*****************************/
```

Figure 2. Style of file annotation writing

(Note: **Description**: Describes the main functions of this program file. The interface, output value, range of values, meaning, control, order, independence, and dependence of this file with other modules or functions. **Function List**: A brief description of the names of functions that contained in this file.)

2.1.3. Function annotation

Annotation should be listed at the head of each function in a C program. The purpose of this function, input parameters, output parameters, return values, call relationships, and so on [3]. The style of the function annotation writing is shown as Figure 3.

```
/************************
Function:
Description:
Calls:
Called By:
Input:
Output:
Return:
Others:
************************/
```

Figure 3. Style of function annotation writing

2.2. Functionality annotation

The writing style of functionality annotation is relatively free. It is used in a source program to describe the function of a statement or a block of one program. It is important to note that functionality annotation should not be added for every statement of a program. Annotation is modified also when the program is modified.

"//……" and "/*……*/" are the two styles of line annotation for functionality annotation in C program. For some senior programmers, from the perspective of modifiability and maintainability, when modifying the program statements, they often change the statements for the line annotation style first, but not delete them directly.

3. Data declaration

Developers need to pay more attention to the style of data declaration to make the data in the source program easier to understand and maintain. In general, the following aspects need to be achieved.

3.1. Roles of data naming

Although the data naming in programs allows customization, the naming needs to be consistent with the meaning that the data represents. Additionally, style of data naming of "Pascal" or "Camel" can be used in coding of C programs too.

3.2. *Sequence in data declaration*

The sequence in data declaration should be standardized. For example, sequence in data declaration is followed as constant declaration, simple variable type declaration, array declaration, structure declaration, pointer declaration, file declaration, and so on in a C program [6-7]. On this basis, some further standardized requirements can be made. For example, in the simple variable type declaration, we can arrange the declaration on simple variables as the following order, such as short int, long int, float, double, char, and so on.

When many variables of the same type are appeared in one declaration statement, they should be arranged in the order of the initial letters. For example, the statement on "float price, cost, size, total, length;" in a C program should be adjusted in "float cost, length, price, size, total;".

4. Program statement writing

Program statement writing follows simple principles which can be easier to read and understand. Indent, "(……) and {……}"writing in pairs, null string are used when writing a C program. Furthermore, one statement is written on one line only. Using less complex conditional decision statements and using more brackets to indicate the order of operations of expressions are followed by developers [6-7].

5. Design style of input and output

The information of program input and output is related to users. Whether the program can be accepted or not by users will be depended on the design style of input and output and coding standard for some extent [6-7]. In this section, some design style of input and output should be followed.

5.1. Style of input

Programmers need to design program statements for testing all the input data in the program in order to identify the error or illegal data inputting to ensure the validity of each input data.

For example, when we solve the area of a triangle by Heron Formula, the source code of C program is the following [8]:

```
#include "math.h"
void main( )
{
        int a, b, c;            // define three integers of a, b, c as the three sides of a triangle;
```

```
    float area, s;
    scanf("%d, %d, %d", &a, &b, &c);  // input three integers of a, b, c as the three sides of a
                                         triangle;
    s =1.0/2*(a+b+c);
    area = sqrt(s*(s-a)*(s-b)*(s-c));        // solve the area of a triangle by Heron Formula;
    printf("area = %8.3f \n", area);         // output the area value of the triangle, keep three
                                         decimal places;
}
```

There are no errors in this program, but lacking of reasonable judgment on the three input data of the three sides of a, b, and c for representing the triangle [7]. When the user inputs digitals of 5, 1 and 2 for the three sides of a triangle, the system will have an exception because of receiving the illegal data. Therefore, the program needs to add statements to determine the reasonableness of the input data, and prompt information is appeared when the data input is error.

The modified program is the following [8]:

```
#include "math.h"
void main( )
{
    int a, b, c;          // define three integers of a, b, c as the three sides of a triangle;
    float area, s;
    scanf("%d, %d, %d", &a, &b, &c);     /*input three integers of a, b, c as the three sides of a
                                         triangle;*/
    if((a+b)>c && (a+c)>b && (b+c)>a)    /*determine whether the three sides of a, b, and c can
                                         form a triangle;*/
        {
            s =1.0/2*(a+b+c);
            area = sqrt(s*(s-a)*(s-b)*(s-c)) ;
            printf("area = %8.3f \n", area) ;
        }                                    /*if inputting three integers of a, b, and c
                                         can be formed a triangle, solve the area of a
                                         triangle by Heron Formula, and output the
                                         area value of the triangle, keep three
                                         decimal places;*/
    else printf("Error!");          //information on data error inputting;
}
```

Furthermore, style of designing of input statements for C programs should be followed. Firstly, steps for inputting should be designed as simple as possible, and maintain a simple input format. Secondly, the way of freedom input and default values for inputting are allowed. Thirdly, when inputting a batch of data, it is best to design the input end identification to provide to the users.

5.2. Style of output

Functionality annotation is needed to be added to all output statements. Before the user inputting the data, it is better to design an output statement that prompts the user to enter the corresponding data [8].

For example,

```
……
printf(“please input : a, b, c：\n”);                /*to prompt for the upcoming input
                                                    information of three digitals of a, b, and c;*/
scanf(“%d，%d，%d”，&a，&b，&c)；          // input a, b, and c;
……
```

Some programmers will annotate all the output statements in the program and design the output report to facilitate white box testing and user reading.

6. Conclusions

Coding style has nothing to do with the correctness of a program, but bad coding style will reduce the readability of the program, and it will also cause difficulties for testing and maintenance later. According to the perspective of software engineering, the pursuit of standardization, clarity, and beauty of writing programs is an important component of designing style of modern coding.

Acknowledgments

This work was supported by Foundation of "Teaching Team of Software Engineering Course" of Anhui SanLian University under Grant No. 15zlgc029, "Software Engineering-Excellent Resource Sharing Course" of Anhui Province under Grant No. 2016gxk048 and "Massive Open Online Course on Software Engineering" of Anhui Province under Grant No. 2015mooc104.

In the end, as the corresponding author of this paper, I would like to express my heartfelt gratitude to my colleagues of Jishan Zhang, Ning Wu, Chun Zheng and Deqing Zhang, and all the authors of the references which are listed at the end of this paper.

References

[1] Q.J. Xong, Q. Gu, J.F. Qu and X.Y. Wang 2018 Reform on Experimental Teaching of C Language Programming Based on Micro Video *Journal of Experimental Technology and Management* **5** 13-16(in Chinese)
[2] L. Guo, J.P. Li 2017 *ASP.NET Dynamic Web Development Technology* (Beijing: POSTS&TELECOM PRESS) (in Chinese)
[3] Information on https://www.cnblogs.com/ace-wu/p/6596911.html
[4] Information on http://www.cnitpm.com/st/80072757.html
[5] Information on https://blog.csdn.net/ycguhang/article/details/7173832
[6] Information on http://c.biancheng.net/view/158.html
[7] A.C. Wang 2013 *Software Engineering Foundation and Case Analysis*(Beijing:CHINA MACHINE PRESS) (in Chinese)
[8] J.J. Yu 2015 *Concise Tutorial on Software Engineering*(Beijing:TSINGHUA UNIVERSITY PRESS) (in Chinese)