

## 5. 程序注释

注释对于程序犹如眼睛对于人的重要性一样,程序越复杂,注释就越显得有价值。没有注释的程序对于读者好比眼前一团漆黑。当然,注释并非越多越好,无意义和多余的注释如同垃圾,不但白写,还可能扰乱了读者的视线,甚至可能出现二义性,比不加注释还要糟糕。良好的注释应使用简明易懂的语言来对程序特殊部分的功能和意义进行说明,既简单明了,又准确易懂,能精确地表述和清晰地展现程序的设计思想,并能揭示代码背后隐藏的重要信息。程序员开发程序的思维体现在注释和规范的代码本身。

书写注释的最重要的功效在于传承,即让继任者能够轻松阅读、复用、修改自己的代码。所以程序员应该养成写注释的习惯。那么通常在哪些地方需要写注释呢?

(1) 在重要的程序文件的首部,对程序的功能、编程者、编程日期以及其他相关信息(如版本号等)加以注释说明。例如,C风格的注释如下:

```
/*程序功能   :介绍变量的使用
   编程者     :Su xiaohong
   日期       :31/7/2010
   版本号     :1.0           */
```

C++风格的注释如下:

```
// 程序功能   :介绍变量的使用
// 编程者     :Su xiaohong
// 日期       :31/7/2010
// 版本号     :1.0
```

(2) 在用户自定义函数的前面,对函数接口加以注释说明。

(3) 在一些重要的语句行的右方,如在定义一些非通用的变量、函数调用、较长的多重嵌套的语句块结束处,加以注释说明。

(4) 在一些重要的语句块的上方,尤其是在语义转折处,对代码的功能、原理进行解释。

写注释时,要注意以下几点:

- (1) 注释不是白话文翻译,不要鹦鹉学舌。
- (2) 不写做了什么,要写想做什么,如何做。
- (3) 注释可长可短,但应画龙点睛,重点加在语义转折处。
- (4) 边写代码边注释。
- (5) 修改代码的同时也修改注释。

(6) 供别人使用的函数必须严格注释,特别是入口参数和出口参数,内部使用的函数以及一些简单的函数可以简单注释。本书为节省篇幅,均使用了这种简单的注释方法。例如:

```
// 两数互换
void Swap(int* x, int* y);
```

注释是与代码距离最近的文档,也是程序员在编写代码时最方便修改的文档。很多软件可