软件工程概论
Software Engineering

刘伟

liuwei@xidian.edu.cn

88204608

CH7. Writing the programs

Content

- Programming Standards and Procedures
- Programming Guidelines
- Documentation

7.1 Programming Standards and Procedures

- Standards for you标准对于你
  - 帮你组织思想，避免错误
  - 有助于你将设计转化成代码，维持设计组件何代码组件之间的一致性
- Standards for others标准对于别人
  - 了解你写了什么
  - 容易理解软件做什么及如何运行
- Matching design with implementation设计与实现的匹配
  - *最关键的标准就是需要在程序设计组件和程序代码组件之间有直接的对应关系*
  - 系统的一般目的是在整个软件生命周期中保持一致
  - 设计和代码之间的一致性是基本问题

Standard format for comments注释的标准格式

```
/* Statement of function:
 * Component name:
 * Programmer:
 * Version:
 * Procedure Invocation:
 * Input Parameters:
 * Output Parameters:
 */
```

Programming Guidelines编程指导

- Major aspects of programming编程的主要方面
  - control structures控制结构
  - Algorithms算法
  - data structures数据结构

Example: control structures

```
        benefit = minimum;
        if (age < 75) goto A;
        benefit = maximum;
        goto C;
        if (AGE < 65) goto B;
        if (AGE < 55) goto C;
A:      if (AGE < 65) goto B;
        benefit = benefit * 1.5 + bonus;
        goto C;
B:      if (age < 55) goto C;
        benefit = benefit * 1.5;
C:      next statement
```

```
        if (age < 55) benefit = minimum;
        elseif (AGE < 65) benefit = minimum + bonus;
        elseif (AGE < 75) benefit = minimum * 1.5 + bonus;
        else benefit = maximum;
```

Algorithms算法

- Efficiency may have hidden costs效率可能隐含代价（成本）
  - cost to write the code faster写更快代码的代价
  - cost to test the code测试代码的代价
  - cost to understand the code理解代码的代价
  - cost to modify the code修改代码的代价

Keep the program simple (1)

1. For the first $10,000 of income, the tax is 10 percent.
2. For the next $10,000 of income above $10,000, the tax is 12 percent.
3. For the next $10,000 of income above $20,000, the tax is 15 percent.
4. For the next $10,000 of income above $30,000, the tax is 18 percent.
5. For any income above $40,000, the tax is 20 percent.

```
    tax = 0.
    if (taxable_income == 0) goto EXIT;
    if (taxable_income > 10000) tax = tax + 1000;
    else{
            tax = tax + .10*taxable_income;
            goto EXIT;
    }
    if (taxable_income > 20000) tax = tax + 1200;
    else{
            tax = tax + .12*(taxable_income-10000):
            goto EXIT;
    }
    if (taxable_income > 30000) tax = tax + 1500;
    else{
            tax = tax + .15*(taxable_income-20000);
            goto EXIT;
    }
    if (taxable_income < 40000){
            tax = tax + .18*(taxable_income-30000);
            goto EXIT;
    }
    else
            tax = tax + 1800. + .20*(taxable_income-40000);
EXIT: ;
```

Keep the program simple (2)

Table 7.1. Sample tax table.

Bracket	*Base*	*Percent*
0	0	10
10,000	1000	12
20,000	2200	15
30,000	3700	18
40,000	5500	20

```
for (int i-2; level=1; i <= 5; i++)
        if (taxable_income > bracket[i])
                level = level + 1;
tax = base[level]+percent[level]*(taxable_income-bracket[level]);
```

General guidelines一般指导原则

- Localize input and output使输入输出局部化
- Include pseudocode 包括伪代码
- Revise and rewrite, rather than patch修改与重写，胜于打补丁
- Reuse复用
  - Producer reuse
  - Consumer reuse

Documentation文档

- Internal documentation内部文档
  - header comment block头部注释块
  - other program comments其他程序注释
  - meaningful variable names and statement labels有意义的变量名和声明标示
  - format to enhance understanding增进理解的格式
  - document data记录数据
- External documentation外部文档
  - describe the problem描述问题
  - describe the algorithm描述算法
  - describe the data描述数据

Information system example (1)

```
Input:  *Opposition schedule*
For each *Television company name*, create *Opposition company*.
    For each *Opposition schedule*,
        Locate the *Episode* where *Episode schedule date* = *Opposition
            transmission date* AND *Episode start time* = *Opposition
                transmission time*
    Create instance of *Opposition* program
    Create the relationships *Planning* and *Competing*
Output:  List of *Opposition programs*
```

```
Opposition schedule = * Data flow *
    Television company name
    + {Opposition transmission date
    + Opposition transmission time + Opposition program name
    + (Opposition predicted rating)}
```

Information system example (2)

```
void Match:: calv(Episode episode_start_time)
{
first_advert = episode_start_time + increment;
// The system makes a copy of Episode
// and your program can use the values directly.
}

 void Match:: calp(Episode* episode)
 {
 episode->setStart (episode->getStart());
 // This example passes a pointer to an instance of Episode.
 // Then the routine can invoke the services (such as setStart
 // and getStart) of Episode using the -> operator.
 }

void Match:: calr(Episode& episode)
{
episode.setStart (episode.getStart());
// This example passes the address of Episode.
// Then the routine can invoke the services (such as setStart
// and getStart) of Episode using the . operator.
}
```