

FEEL THE MEANING OF THE TRIP

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

DREAM

MY DREAM WILL NEVER STOP

# 高级语言程序设计

## 实验三 编码规范&循环控制



哈尔滨工业大学(深圳)  
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

GO!  
TAKE YOU ON A TRIP

探索 从未停止

# 目录

CONTENT

01

实验回顾

02

实验目的

03

编码规范

04

实验内容



# 实验回顾

## ➤ scanf()函数的返回值为int型

表示正确按指定格式输入变量的个数；也即能正确接收到值的变量个数。

例如执行 `scanf( "%d %d" , &a, &b)`

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a = 0, b = 0;
    int status;

    status = scanf("%d %d", &a, &b);
    printf("a=%d b=%d\n", a, b);
    printf("status=%d\n", status);

    return 0;
}
```

```
D:\code\c\2022\exp2_correct\bin\Debug\exp2_correct.exe
3 4
a=3 b=4
status=2

Process returned 0 (0x0)   execution time : 6.934 s
Press any key to continue.
```

```
D:\code\c\2022\exp2_correct\bin\Debug\exp2_correct.exe
3, 4
a=3 b=0
status=1

Process returned 0 (0x0)   execution time : 3.873 s
Press any key to continue.
```

```
D:\code\c\2022\exp2_correct\bin\Debug\exp2_correct.exe
, 3 4
a=0 b=0
status=0

Process returned 0 (0x0)   execution time : 3.718 s
Press any key to continue.
```



# 实验回顾

## ➤ 三个数两两相等

错误写法: `a == b == c` 等价于 `((a == b) == c)`

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, a, b, c;
    a = 5;
    b = 5;

    c = 5;
    i = (a == b == c);
    printf("a = %d, b = %d, c = %d\n", a, b, c);
    printf("i = %d\n\n", i);

    c = 1;
    i = (a == b == c);
    printf("a = %d, b = %d, c = %d\n", a, b, c);
    printf("i = %d\n", i);
}
```

D:\codetest\experiment2\_debug\bin\Debug\experiment2\_debug.exe

a = 5, b = 5, c = 5  
i = 0

a = 5, b = 5, c = 1  
i = 1

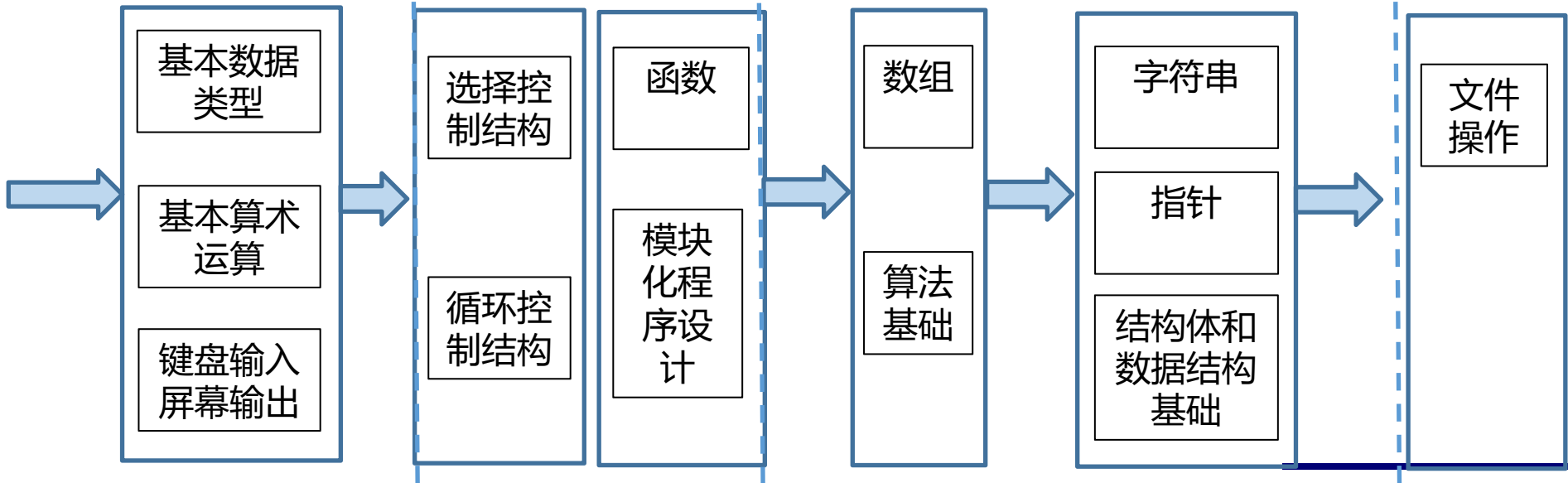
Process returned 0 (0x0) execution time : 0.274 s  
Press any key to continue.

应改为: `a == b && b == c`



# 实验课程安排与考核标准

编号	1	2	3	4	5	6	7	8	9	10
学时数	2	2	2	2	2	2	2	2	2	2
实验项目	初级编程	算数运算	循环控制	模块化程序设计	数组算法	字符串二维数组	指针	结构体	文件管理	
分数	1	2	2	2	3	3	3	3	3	
授课内容	集成开发环境	程序调试	编码规范	程序测试				软件文档	期末检查1	期末检查2





# 实验目的

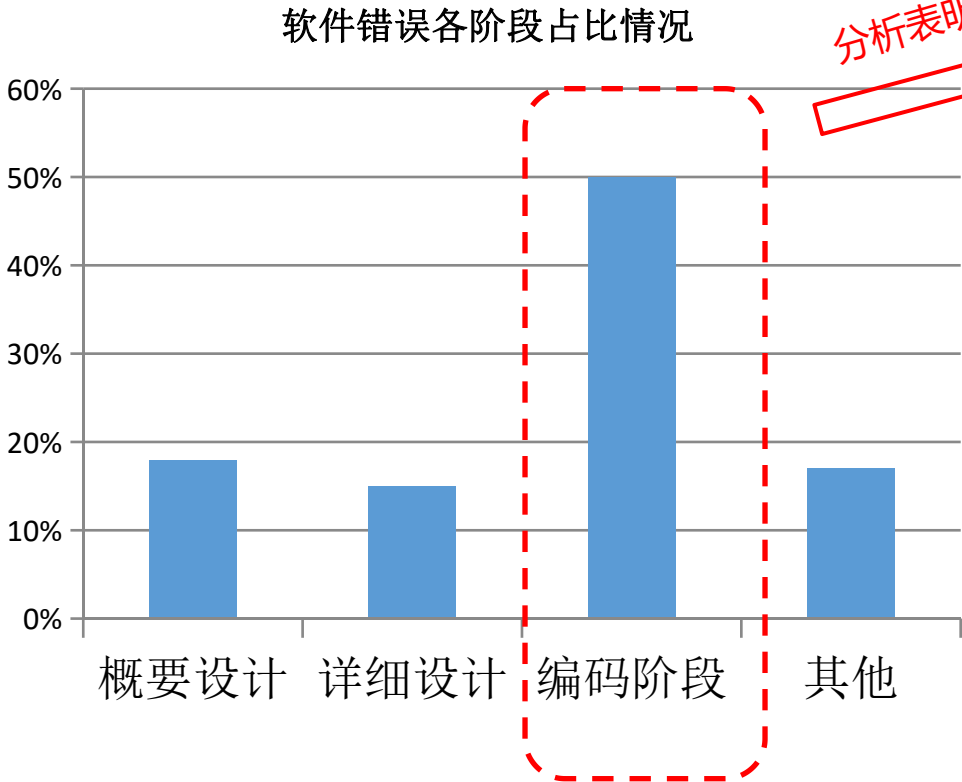
---

- 了解编码规范的重要性
- 掌握循环语句及循环控制方法



# 编码规范

Bell实验室的研究资料表明，软件错误中，编码阶段产生的错误占的比例接近50%。



分析表明

“编码阶段” 错误种类	比例
语法错误	20%
函数(模块)之间接口错误 代码可理解度低导致优化维护阶段对代码的错误修改引起的错误 未严格检查软件逻辑导致的错误	50%以上
其他	10~30%

解决办法

Bell实验室的研究人员制定了**详细的软件编程规范**，并培训每一位程序员，最终的结果把编码阶段的错误率**降低至10%左右**。



# 编码规范

```
#include<math.h>
#include<stdio.h>
#define a 3.14159
main()
{
    double b, c, d;
    printf( "Input b:" );
    scanf( "%lf" , &b);
    c= 4 * a * pow(b, 2);
    d= 4.0 / 3.0 * a * pow(b, 3);
    printf( "c=%lf\n" , c);
    printf( "d=%lf\n" , d);
}
```

VS

```
*****
** 程序描述:用于计算球体面积及体积
** 日期:2020/10/15
** 作业题目:实验二任务一
** 输入: r
** 输入范围: 正数, 大于零
** 输出: surface, volume
** surface: 球体面积, volume: 球体体积
*****/
#include<math.h>
#include<stdio.h>
main()
{
    const double PI= 3.14159;

    double r, surface, volume;

    //输入半径r
    printf("Input r:");
    scanf("%lf", &r);

    //输出球体表面积
    surface = 4 * PI * pow(r, 2);
    printf("surface=%lf\n", surface);

    //输出球体体积
    volume = 4.0 / 3.0 * PI * pow(r, 3);
    printf("volume=%lf\n", volume);
}
```





# 编码规范

---



**Bad Code**



**Good Code**

---

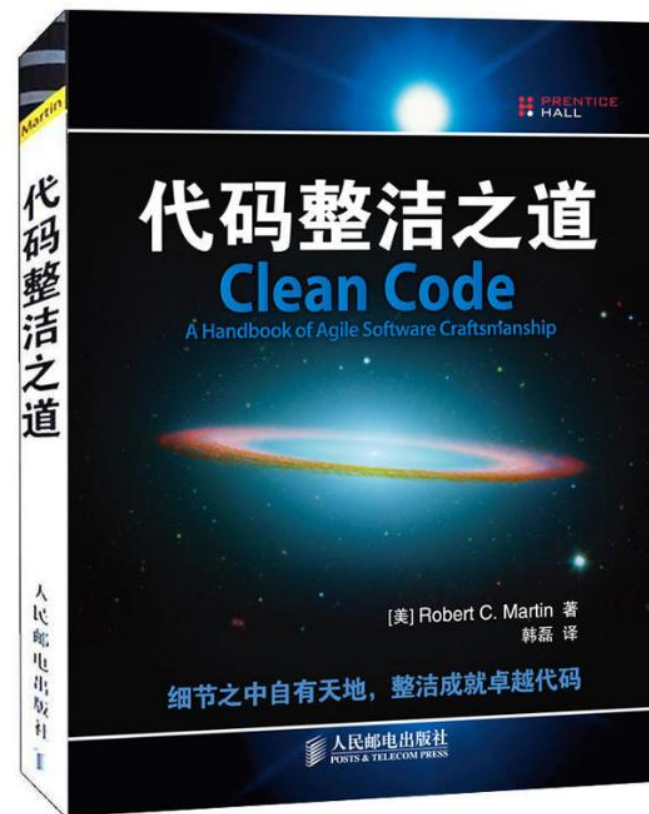


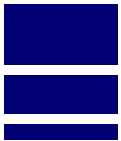
# 编码规范

---

- 命名规范
- 程序排版
- 程序注释

推荐书目：





# 编码规范-命名规范

## ➤工程命名

- 选择有实际意义的单词
- 单词与单词之间应使用下划线分开或首字符大写
- Project、\*.c文件、及保存文件的路径，严禁出现中文和空格

experiment3\_task1\_MoneyChanger  
Experiment3Task1MoneyChanger

## ➤变量命名

- 名副其实，必须具有一定的实际意义；
- 命名形式为xAbcFgh，x由变量类型确定，Abc、Fgh表示连续意义字符串；
- 如果连续意义字符串仅两个，可都大写，如OK。

内容	类型	不规范的变量名	规范的变量名
身高	float	h	fHeight
年龄	int	a	iAge
数学成绩	float	s	fMathScore

## ➤函数命名

- 必须具有一定的实际意义；
- 第一个字母必须使用大写字母，要求用大小写字母组合规范函数命名，用下划线间隔。

float Male\_HeightCompute (float fFatherHeight, float fMotherHeight);

## ➤宏和常量的命名

- 必须具有一定的实际意义；
- 常量和宏定义必须全部以大写字母来撰写，中间可根据意义的连续性用下划线连接。

#define PI 3.14159265



# 编码规范-程序排版

## ➤缩进

大括号内的程序块采用缩进对齐。

## ➤空行

两段代码，分别完成不同的功能，这两段代码之间可使用一空行，完成逻辑上的代码区分。

## ➤一行只写一条语句

## ➤操作符前后要加空格

Codeblocks: 右键Format use AStyle

```
float  Male_HeightCompute (float fFaHeight, float fMoHeight)
{if(sex== 'm' ) myHeight=(faHeight + moHeight)*0.54;
else myHeight = (faHeight*0.923 + moHeight)/2.0;
if(sports== 'y' ) myHeight*=(1+0.02);
if(diet== 'y' ) myHeight *= (1+0.015);
else
```

```
float  Male_HeightCompute (float fFaHeight, float fMoHeight)
```

```
{
```

```
if ( sex == 'm' )
```

```
myHeight = (faHeight + moHeight) * 0.54;
```

```
else
```

```
myHeight = (faHeight * 0.923 + moHeight) / 2.0;
```

```
if ( sports == 'y' )
```

```
myHeight *= (1 + 0.02);
```

```
if ( diet == 'y' )
```

```
myHeight *= (1 + 0.015);
```

```
else
```

```
myHeight *= 0.9;
```

```
}
```

缩进

空行



- 函数开头的注释模板

```
/******  
** 函数名:  
** 输入: a,b,c  
** a---  
** b---  
** c---  
** 输出: x---  
** x 为 1, 表示...  
** x 为 0, 表示...  
** 功能描述:  
** 全局变量:  
** 调用模块:  
** 作者:  
** 日期:  
** 修改:  
** 日期:  
** 版本  
******/
```

- 文件开头的注释模板

```
/******  
** 文件名:  
** 创建人:  
** 日期:  
** 修改人:  
** 日期:  
** 描述:  
** 版本:  
**-----  
******/
```

- 程序中的注释模板

```
/*-----*/  
/* 注释内容 */  
/*-----*/
```

或者

```
//注释内容
```

能用函数和变量名说明时就不要用注释！



# 捉虫时刻

**【改错题】** 使用程序调试方法，改正下面程序中的错误，并简单描述调试的过程。

找出200以内的所有完全数，并输出其因子。

```
#include <stdio.h>
int main()
{
    /*****
    代码功能：找出200以内的所有完全数，并输出其真因子
    *****/
    int i,j,s=1;
    for(i=2;i<=200;i++)
    {
        for(j=2;j<=i/2;j++)
        {
            if(i%j==0){
                s = s + i;
            }
        }
        if(s==i)
        {
            printf("%d=1",i);
            for(j=2;j<=i/2;j++)
            {
                if(i%j==0)
                {
                    printf("+%d",j);
                }
            }
            printf("\n");
        }
    }
    return 0;
}
```

**完全数：**如果一个数恰好等于它的真因子（即除了自身以外的约数）之和，则称其为完全数。

例如：6=1+2+3

1、2、3为6的真因子，且6恰好也是其真因子之和。

**输出格式：**

6=1+2+3

28=1+2+4+7+14



## 实验内容 – 编程题

### 【编程题1】 素数求和

有N个数 ( $0 < N < 1000$ )，请编程实现，求N个数中所有素数之和。

【输入】 第一行给出整数M ( $0 < M < 10$ )，代表有多少组测试数据；  
每组测试数据第一行为N，表示该组测试数据的数量，接下来的N个数为要测试的数据，每个数都小于1000。

【输出】 每组测试数据结果占一行，输出测试数据的所有素数和。

【样例输入】

```
3
5
1 2 3 4 5
8
11 12 13 14 15 16 17 18
10
21 22 23 24 25 26 27 28 29 30
```

【样例输出】

```
10
41
52
```

```
3
5
1 2 3 4 5
8
11 12 13 14 15 16 17 18
10
21 22 23 24 25 26 27 28 29 30
10
41
52
```

输入

输出

```
Process returned 0 (0x0)   execution time : 28.709 s
Press any key to continue.
```



## 实验内容 – 编程题

### 【编程题2】 计算天数

根据输入年月日，计算该日在本年中是第几天。

【输入】 年,月,日

【输出】 请输入日期 (年, 月, 日)

"\n%d月%d日是%d年的第%d天。"

【样例输入】

1990,2,14

【样例输出】

请输入日期 (年, 月, 日)

1990,2,14

2月14日是1990年的第45天。

## 实验内容 – 编程题

### 【编程题3】 计算时间夹角

钟面上的时针和分针之间的夹角总是在  $[0 - 180]$  度之间。举例来说，在12点两针之间的夹角为 0 度，而在6点夹角为 180 度，在3点为 90 度。

【输入】输入必须为小时h分钟m的格式，如12h00m, 5h43m等，其他形式的输入都不接收。  
注意：要求对“小时”和“分钟”进行数值有效判断（小时在 $[0, 12]$ ，分钟在 $[0, 60)$ 之间），不满足以上要求，则需重新输入。

【输出】用常用格式显示时间以及这个时候时针和分针间的最小夹角，精确到小数点后一位。

#### 【样例输入】

5h43m

#### 【样例输出】

Please input time

5h43m

At 5:43 the angle is 86.5 degrees.

```
D:\code\c\2022\exp3\Pro2\bin\Debug\Pro2.exe
Please input time
88h607m
Please input time
5h43m
At 5:43 the angle is 86.5 degrees.
Process returned 0 (0x0)   execution time : 9.383 s
Press any key to continue.
```

非法输入需重输入

#### 【说明】

- 1.以表中心到12点的连线为基准，分针每走1分钟是6度，时针与基准的夹角每个小时是30度。从整点开始，每过1分钟时针再增加0.5度。要求结果角度为正值。
- 2.二者之间角度不应大于180度，如果大于，应用360度减去该角度。



青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

FEEL THE MEANING OF THE TRIP

DREAM

MY DREAM WILL NEVER STOP

请同学们开始实验!



哈爾濱工業大學(深圳)  
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

GO!  
TAKE YOU ON A TRIP

探索 从未停止