

C语言

第3讲：简单的C程序设计

赵岩

哈尔滨工业大学软件学院

August 20, 2011

目录

① 基本架构

目录

- 1 基本架构
- 2 C语句

目录

- 1 基本架构
- 2 C语句
- 3 字符及字符串输入/输出

目录

- 1 基本架构
- 2 C语句
- 3 字符及字符串输入/输出
- 4 格式输入/输出
 - 格式输入
 - 格式输出

C程序基本构成

```
/*打印Hello, World!到屏幕*/
```

注释

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

文件包含

```
main()
```

主函数

```
{
```

程序块

库函数

函数参数

```
printf("Hello, World!\n");
```

```
}
```

C程序基本构成

- 注释语句不被编译器处理
- 函数是C语言中的功能单元
- 有且只有一个main函数,整个程序的入口和出口。

好的代码风格

- 代码风格
- cms.hit.edu.cn/mod/resource/view.php?id=155

```
#include <stdio.h>
/* 对 fahr = 0, 20, ..., 300
   打印华氏温度与摄氏温度对照表 */
main()
{
    int fahr, celsius;
    int lower, upper, step;

    lower = 0; /* 温度表的下限 */
    upper = 300; /* 温度表的上限 */
    step = 20; /* 步长 */
    fahr = lower;

    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

坏的代码风格

```
#include <stdio.h>
main(){int fahr,celsius;
int lower,upper,step;
lower=0;upper=300;step=20;fahr=lower;
while (fahr<=upper){
celsius=5*(fahr-32)/9;
printf("%d\t%d\n",fahr,celsius);
fahr=fahr+step;}}
```



C程序中的语句

- 每个语句必须以分号“;”结束。(注意是“每个语句”而不是“每行语句”)?

```
1 /* int i = 5; */  
2 int  
3 i  
4 =  
5 5  
6 ;
```

C语句分类

- 变量声明语句
 - `int x, y;`
- 表达式语句
 - `i++;`
 - `i = i + 1;`
- 空语句
 - `;`
- 复合语句
 - `t = x; x = y; y = t;`
- 控制语句

表达式语句

- 赋值表达式语句
- 函数调用语句
- `total = m * pow(1+r, n);`

复合语句

- 在什么情况下使用复合语句？
 - 条件语句和循环语句

```
1  if (x!=y)
2  {
3      t=x;
4      x=y;
5      y=t;
6  }
```

空语句

- 空语句有什么作用？
 - 自顶向下程序设计时用在那些未完成的模块中
 - 延时用的空循环

```
1 DataInitialize ()
2 {
3     ; /* Empty statement */
4 };
5 main ()
6 {
7     ...;
8     DataInitialize ();
9 }
```

输入输出函数

- 如何进行数据的输入输出？
 - 输入：从标准输入设备上输入数据到计算机内存
 - 输出：将计算机内存中的数据送到标准输出设备
- C语言中输入输出操作通过调用标准库函数来实现
 - `#include <stdio.h>` 在指定的文件目录中找该文件

字符输入输出函数

- 字符输出函数
 - putchar(ch)
 - 输出一个字符ch到STDOUT
- 字符输入函数
 - getchar()
 - 无参数
 - 函数值为从STDIN接收一个字符

字符输入输出实例

Demo 1: character input and output

```
1    char  ch;
2    /* first */
3    ch = getchar ();
4    putchar(ch); putchar ('\n');
5    /* second */
6    ch = getchar ();
7    putchar(ch); putchar ('\n');
8    /* third */
9    ch = getchar ();
10   putchar(ch); putchar ('\n');
```

实验结论

- 结论1: `getchar`从缓冲区里面读数据，而不是从键盘。
- 结论2: 调用输入函数且缓冲区为空的时候，界面会停住。
- 结论3: `getchar`每次只读一个任意字符（包括回车）。

字符串相关函数

- 字符串输入函数 `gets(char * str);`
- 字符串输出函数 `puts(char * str);`

Demo 2: character input and output

```
1 #include <stdio.h>
2 main()
3 {
4     char str[30];
5     gets(str);
6     puts(str);
7 }
```

字符串相关函数

Demo 3: string input and output

```
1 #include <stdio.h>
2 main(){
3     char str[30];
4     char c;
5     gets(str);
6     puts(str);
7
8     c = getchar();
9     putchar(c);
10
11 }
```

实验结论

- 结论1: `getchar`从缓冲区里面读数据，而不是从键盘。
- 结论2: 调用输入函数且缓冲区为空的时候，界面会停住。
- 结论3: `getchar`每次只读一个任意字符（包括回车）。
- 结论4: `gets`把回车从缓冲区读走。

scanf输入函数

- scanf(格式控制字符串, 地址列表);

Demo 4: scanf example

```
1 #include <stdio.h>
2 main()
3 {
4     int a, b;
5     printf("Please input a and b:");
6     scanf("%d%d", &a, &b);
7     printf("a=%d, b=%d, a+b=%d\n", a, b, a+b);
8 }
```

scanf常见错误1

Demo 5: scanf error 2

```
1 #include <stdio.h>
2 main()
3 {
4     int a, b;
5
6     printf("Please input a and b:");
7     scanf("%d,%d", &a, &b);
8     printf("a=%d, b=%d, a+b=%d\n", a, b, a+b);
9 }
```

scanf常见错误2

Demo 6: scanf error 1

```
1     int a;
2     do
3     {
4         printf("*\n");
5         scanf("%d",&a);
6         if(111==a)
7         {
8             break;
9         }
10    } while(1);
```

全真七子！

- 结论1: getchar从缓冲区里面读数据，而不是从键盘。
- 结论2: 调用输入函数且缓冲区为空的时候，界面会停住。
- 结论3: getchar每次只读一个任意字符（包括回车）。
- 结论4: gets把回车从缓冲区读走。
- 结论5: scanf 中%d提取数字，忽略空格，tab，回车。
- 结论6: 如果缓冲区不空，scanf按自己的格式提取
- 结论7: 提取成功，从缓冲区提走数据；提取失败，不从缓冲区提走数据。你可以手工fflush(stdin)清空缓冲区。

scanf思考题?

Demo 7: scanf error 3

```
1 int a; char b;  
2 printf("please input an integer:");  
3 scanf("%d",&a);  
4 printf("interger: %d\n",a);  
5  
6 printf("please input a character:");  
7 scanf("%c",&b);  
8 printf("character: %c\n",b);
```

格式输出函数

- printf(格式控制字符串, 输出项表列);
- 输出若干个任意类型的数据

```
1 printf("a=%d b=%f", a, b);
```

格式转换说明符

- d 以带符号十进制整数输出
- o 以八进制无符号整数输出（无前导0）
- x 以十六进制无符号整数输出（无前导0x）
- u 以十进制无符号整数输出
- c 以字符形式输出单个字符
- s 输出一个字符串
- f 以小数形式输出浮点数（6位小数）
- e 以标准指数形式输出（6位小数）
- g 选用

格式修饰符

- `m` 表示全部数据占用的最小宽度
 - 数据宽度大于`m`, 按实际宽度输出
 - 数据宽度小于`m`, 补空格
- `.n` 对实数表示输出`n`位小数对字符串表示最多输出的字符个数
- - 改变输出内容的对齐方式缺省为右对齐

Demo 8: printf example

```
1 float f1 = 1234.56789,
2 printf("%3.2f", f); /* 1234.57 */
3 printf("%9.2f", f); /* --1234.57 */
4 printf("%-9.2f", f); /* 1234.57-- */
5 printf("%.2f", f); /* 1234.57 */
```

printf实例

Demo 9: printf example

```
1 float   f1 = 100.15799, f2 = 12.55, f3 = 1.7;
2 int     n1 = 123, n2 = 45, n3 = 6;
3 printf("printf WITHOUT width or precision
4   specifications:\n");
5 printf("%f\n%f\n%f\n", f1, f2, f3);
6 printf("%d\n%d\n%d\n", n1, n2, n3);
7
8 printf("printf WITH width and precision
9   specifications:\n");
10 printf("%5.2f\n%6.1f\n%3.0f\n", f1, f2, f3);
11 printf("%5d\n%6d\n%3d\n\n", n1, n2, n3);
```

代码风格的建议

- 乐学网上的资源（来自苏晓红老师）

总结



谢谢大家，欢迎提问！