

# 小马加编信息学教案(七)

## C++循环结构(一)

- 一. 课程内容
- 二. 知识讲解
  - 1. 循环结构
  - 2. while 语句
    - \* 2.1 while 语句的格式
    - \* 2.2 while 语句的含义
    - \* 2.3 while 语句的应用举例
  - 3. do-while 语句
    - \* 3.1 do-while 语句的格式
    - \* 3.2 do-while 语句的含义
    - \* 3.3 do-while 语句的应用举例
    - \* 3.4 do-while 使用注意
  - 4. while 与 do-while 的综合应用
- 三. 经典例题
- 四. 提高巩固

## 一. 课程内容

1. 循环结构
2. while 语句
3. do-while 语句

## 二. 知识讲解

### 1. 循环结构

在程序设计中，经常需要**反复执行某一条语句或一个语句块**，这种结构称为“循环结构”或者“重复结构”

其实，计算机最擅长的就是重复做一件事情，所以循环结构在程序设计中应用非常广泛。  
在 C++ 中，循环结构有三种实现语句：**while 语句**、**do-while 语句**和 **for 语句**。

### 知识储备

- 我们通常将“表达式”称为循环条件，把“语句块”称为循环体，整个循环的过程就是不停判断循环条件、并执行循环体代码的过程。

## 2. while 语句

### 2.1 while 语句的格式

while 语句的格式如下：

```
while (表达式) {
    循环体
}
```

### 2.2 while 语句的含义

其含义为：**先计算表达式（一般称为循环条件）的值，当表达式的值为真（循环条件成立）时，去执行一次循环体。**

和 if 语句不同的是，执行完一次循环体后，while 语句又回到开始处，继续计算和判断表达式的真假，决定是否再次执行循环体。

也就是“当表达式成立时，不断重复执行循环体”，所以**又称为“当型循环”**。

### 2.3 while 语句的应用举例

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int i = 1, sum = 0;
    while(i <= 100) {
        sum += i;
        i++;
    }
    cout << sum << endl;
    return 0;
}
```

- 代码解释

上述代码将*i*初始化为1，*sum*初始化为0，

通过循环条件  $i \leq 100$  限制 *i* 只能在 100 范围以内，并且每次循环过程中会将 *i* 累加至 *sum*，再令 *i* 自加

通过以上代码，可以让 *sum* 累加从 1 至 100 的所有整数值，即求出  $\sum_{i=1}^{100} i$

## 3. do-while 语句

### 3.1 do-while 语句的格式

do-while 语句的格式如下：

```
do {  
    循环体  
} while (表达式) ;
```

### 3.2 do-while 语句的含义

其含义为：先执行一次循环体，然后判断表达式是否成立，如果成立，则返回继续执行循环体，直到表达式不成立，才退出循环。

do-while 一般被称为“直到型循环”。

### 3.3 do-while 语句的应用举例

```
#include<cstdio>  
using namespace std;  
  
int main() {  
    int i = 1;  
    do {  
        printf( " @ " );  
        i++;  
    } while(i <= 10);  
    return 0;  
}
```

### 3.4 do-while 使用注意

- do-while 中的\*\* while 圆括号后的分号一定注意不可少\*\*

## 4. while 与 do-while 的综合应用

- 尝试写出以下各代码段的循环体循环次数与输出结果

```
int i = 1;  
while(i <= n) {  
    cout << i <<endl;  
    i++;  
}
```

```
int i = 0;
while(i < n) {
    cout << i <<endl;
    i++;
}
```

```
int i = 0;
while(i <= n) {
    i++;
    cout << i <<endl;
}
```

```
int i = 1;
do {
    cout << i <<endl;
    i++;
} while(i <= n);
```

```
int i = 0;
do {
    i++;
    cout << i <<endl;
} while(i < n);
```

## 三. 经典例题

1. 阅读下列程序，写出程序结果，并上机编程检验答案

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int i = 100, x = 0, y = 0;
    while(i > 0) {
        i--;
        x = i % 8;
        if(x == 1)
            y++;
    }
    cout << y << endl;
    return 0;
}
```

2. 自制求和器

现在需要你制作一个求和器，求出 $1+2+3+\dots+n$ 的和

即输入正整数 $n$ ，求 $\sum_{i=1}^n i$

输入格式

一行一个正整数  $n$ ;

输出格式

一个正整数，表示 $\sum_{i=1}^n i$

样例输入	样例输出
10	55
100	5050

数据范围

$1 \leq n \leq 100000$

### 3. 统计正数

输入若干整数，以 0 结尾，统计其中有多少个正整数。

输入格式

一行若干整数，最后一个为 0。

输出格式

一行一个整数，表示输入的数据中正整数的个数。

样例输入	样例输出
3 6 -3 2 0	3
0	0

## 四. 提高巩固

### 1. Peter的烟

Peter 有 $n$ 根烟，他每吸完一根烟就把烟蒂保存起来， $k(k>1)$ 个烟蒂可以换一个新的烟，那么 Peter 最终能吸到多少根烟呢？

输入格式

一行包括两个正整数  $n, k$

### 输出格式

对于每组测试数据，输出一行包括一个整数表示最终烟的根数。

样例输入	样例输出
2 3	8
3 2	9

### 数据范围

$n, k \leq 10^8$

## 2. 投资收益

小华在去年赚了一大笔钱。他想把这些钱用于投资，并对自己能得到多少收益感到好奇。已知投资的年利率为  $r$ ，小华现有  $m$  元钱。他想知道投资多少年后，他的钱将会超过  $y$  元。

### 输入格式

一行三个整数  $r$ 、 $m$  和  $y$ ，相邻两个整数之间用一个空格隔开。

### 输出格式

一行一个整数，即要投资的年数。保证答案的范围在  $1 \sim 400$  之间。

样例输入	样例输出
5 5000 6077	4

### 数据范围

$m, r$  为整数

且  $0 \leq r \leq 20, 100 \leq m \leq 1000000$

### 样例说明

第一年后： $1.05 * 5000 = 5250$

第二年后： $1.05 * 5250 = 5512.5$

第三年后： $1.05 * 5512.5 = 5788.125$

第四年后： $1.05 * 5788.125 = 6077.53125$

## 3. 数字和

输入一个正整数，输出它的各位数字之和。

### 输入格式

一行一个正整数， $int$  范围以内。

输出格式

一行一个正整数，表示答案。

样例输入	样例输出
258	15
2333	11