第四十九节： 循环语句 do while 和 for。

【49.1　do while 语句的常见格式。】

格式如下：

```
do
{
      语句 1;
      语句 2;
      ……
      语句 N;
 } while(条件);
```

上述代码，单片机从上往下执行语句，先从 do 那里无条件进来，从“语句 1”开始往下执行，一直执行到“语句 N”，才开始判断 while(条件)的条件是否为真，如果为真继续返回到 do 的入口处，继续从“语句 1”开始往下执行，依次循环。大家留意到了吗，do while 和 while 语句有什么差别？差别是，do while 是先无条件进来执行一次循环体（花括号里所有的程序代码），执行到循环体最底部才判断 while(条件)的条件是否为真来决定是否继续循环，先上车再买票。而 while 语句是先判断条件是否为真再决定是否需要进入循环体，先买票再上车。

【49.2　for 语句的简介。】

for 语句也是循环语句，任何 for 语句能实现的功能都可以用 while 语句来实现同样的功能，for 语句和 while 语句有什么差别呢？for 语句把变量初始化，变量的条件判断，变量在执行循环体后的步进变化这三个常见要素集成在语句内部，以标准的格式书写出来。在很多场合下，for 在书写和表达方面比 while 语句显得更加简洁和直观。

【49.3　for 语句的自加格式。】

格式如下：

```
for(变量的初始化语句; 变量的条件判断;变量在执行一次循环体后自加的步进变化)
{
      语句 1;
      语句 2;
      ……
      语句 N;
}
```

在把上述变成更具体的代码例程如下：

```
for(i=0; i<3;i++)
{
      语句 1;
      语句 2;
      ……
```

```
        语句 N;
    }
```

上述代码，单片机从上往下，在进入循环体前，先把变量 i 初始化赋值 0（这行初始化代码在整个循环期间只被执行 1 次），然后判断 i 是否小于 3 这个条件，如果此条件为真，就开始正式进入循环体，从“语句 1”往下执行到“语句 N”，执行完一次循环体后，i 就自加 1（因为“i++”语句），此时 i 从原来初始化的 0 变成了 1，接着再返回来到 for 语句的条件判断”i<3”那里，判断 i 是否继续满足“小于 3”这个条件，如果此条件为真就继续往下执行，否则就跳过循环体结束当前循环。上述 for 语句实现的功能如果用 while 语句来写，等效于以下代码：

```
    i=0;  //进入循环体之前先初始化给予初值
    while(i<3)
    {
        语句 1;
        语句 2;
        ……
        语句 N;
        i++;   //执行一次循环体之后此变量自加发生变化
    }
```

上述的 while 循环语句只执行了 3 次循环体。

【49.4　for 语句的自减格式。】

刚才讲的 for(i=0; i<3;i++)这种格式，它的变量 i 是不断自加的。还有一种比较常见的格式是 i 不断自减的，它的格式如下：

```
    for(i=3; i>0;i--)
    {
        语句 1;
        语句 2;
        ……
        语句 N;
    }
```

上述自减的 for 语句功能如果用 while 语句来写，等效于以下代码：

```
    i=3;  //进入循环体之前先初始化给予初值
    while(i>0)
    {
        语句 1;
        语句 2;
        ……
        语句 N;
        i--;   //执行一次循环体之后此变量自减发生变化
    }
```

上述的 while 循环语句只执行了 3 次循环体。

【49.5　for 省略花括号，没带分号。】

前面讲的 if 和 while 语句中，都提到了省略花括号的情况，for 语句也有这种写法，而且省略之后默认的有效范围都是一样的。请看例子如下：

```
for(i=0; i<3;i++)   //注意，这里没带分号。
     语句 1;
     语句 2;
     ……
     语句 N;
```

分析：上述代码，跟 if 语句一样，此时循环体默认只包含“语句 1”，等效于：

```
for(i=0; i<3;i++)  //注意，这里没带分号。
{
     语句 1;
}
     语句 2;
     ……
     语句 N;
```

【49.6　for 省略花括号，带分号。】

```
for(i=0; i<3;i++);   //注意，这里带分号。
     语句 1;
     语句 2;
     ……
     语句 N;
```

分析：注意，此时循环体默认不包含“语句 1”，而是等效于：

```
for(i=0; i<3;i++)
{
    ;  //空语句。
}
    语句 1;
    语句 2;
    ……
    语句 N;
```

此时循环体内先循环执行三次空语句，然后才会结束 for 循环，接着才从“语句 1”开始往下执行。

【49.7　for 循环语句的条件判断。】

上面举的例子中，仅仅列出了 for 语句条件判断的小于号关系符“<”，其实，for 语句条件判断的关系符跟 if 语句是一样通用的，凡是 if 语句能用的关系符都可以用在 for 语句上，比如“>”，“!=”，“==”，“<=”，“>=”等等。如下：

```
    for(i=0;i<=3;i++);  //小于等于的情况。这种写法是合法的。
    for(i=0;i!=3;i++);  //不等于的情况。这种写法是合法的。
    for(i=0;i==3;i++);  //等于的情况。这种写法是合法的。
```

【49.8　例程练习和分析。】

编写一个程序来熟悉一下 do while 和 for 语句的使用。

程序代码如下：

```
/*---C 语言学习区域的开始。-----------------------------------------------*/

    unsigned char a=0;  //观察这个数最后的变化
    unsigned char b=0;  //观察这个数最后的变化
    unsigned char c=0;  //观察这个数最后的变化

    unsigned char i;  //控制循环体的条件判断变量

void main() //主函数
{
      i=3;
      do
      {
         a=a+1;  //每执行一次循环体 a 就增加 1,此行代码被循环执行了 3 次
         i=i-1;  //i 不断变小
      }while(i); //i 不断变小，当 i 变为 0 时才跳出此循环体

      for(i=0;i<3;i++)
      {
         b=b+2;  //此行代码被循环执行了 3 次
      }

      for(i=3;i>0;i--)
      {
         c=c+3;  //此行代码被循环执行了 3 次
      }

      View(a);  //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
      View(b);  //把第 2 个数 b 发送到电脑端的串口助手软件上观察。
      View(c);  //把第 3 个数 c 发送到电脑端的串口助手软件上观察。
```

```
    while(1)
    {
    }
}

/*---C 语言学习区域的结束。-----------------------------------------------*/
```

在电脑串口助手软件上观察到的程序执行现象如下：

```
开始...

第 1 个数
十进制:3
十六进制:3
二进制:11

第 2 个数
十进制:6
十六进制:6
二进制:110

第 3 个数
十进制:9
十六进制:9
二进制:1001
```

分析：

变量 a 为 3。a 从 0 开始，循环加 1，一共 3 次，因此等于 3。

变量 b 为 6。b 从 0 开始，循环加 2，一共 3 次，因此等于 6。

变量 c 为 9。c 从 0 开始，循环加 3，一共 3 次，因此等于 9。

【49.9　如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。