第五章 循环结构程序设计

作者：石璞东

参考资料：《C程序设计（第四版）》谭浩强

5.1 为什么需要循环控制

使用场景：

- 要向计算机输入全班50个学生的成绩；
- 分别统计全班50个学生的平均成绩；
- 求30个整数之和；
- 检查30个学生的成绩是否及格；

5.2 用`while`语句实现循环

`while`语句的一般形式如下：`while(表达式) 语句`，其执行过程为先判断条件表达式，后执行循环体语句，只要当循环条件表达式为真，就执行循环体语句。

例5.1 求1+2+3+...。

```c
#include <stdio.h>
int main(){
    int num,i=1,sum=0;
    printf("请输入数值：");
    scanf("%d",&num);
    while (i<=num) {
        sum += i;
        i++;
    }
    printf("sum=%d\n",sum);
    return 0;
}
```

5.3 用`do...while`语句实现循环

`do...while`语句的一般形式为：

```c
do
  语句
while(表达式);
```

其执行过程为：先执行循环体，然后再检查条件是否成立，若成立，再执行循环体。

例5.2 求1+2+3+...。

```c
#include <stdio.h>
int main(){
    int num,i=1,sum=0;
    printf("请输入数值： ");
    scanf("%d",&num);
    do{
        sum += i;
        i++;
    }while (i<=100);
    printf("sum=%d\n",sum);
    return 0;
}
```

在一般情况下，用`while`语句和用`do...while`语句处理同一问题时，若二者的循环体部分是一样的，那么结果也一样，但是如果`while`后面的表达式一开始就为假时，两种循环的结果是不同的。

例5.3 `while`循环&&`do...while`循环比较

```c
#include <stdio.h>
int main(){
    int i = 1,sum = 0;
    printf("请输入i值:");
    scanf("%d",&i);
    while (i<=10) {
        sum += i;
        i++;
    }
    printf("sum=%d\n",sum);
    return 0;
}
```

```c
#include <stdio.h>
int main(){
    int i = 1,sum = 0;
    printf("请输入i值:");
    scanf("%d",&i);
    do{
        sum += i;
        i++;
    }while(i<=10);
    printf("sum=%d\n",sum);
    return 0;
}
```

5.4 用`for`语句实现循环

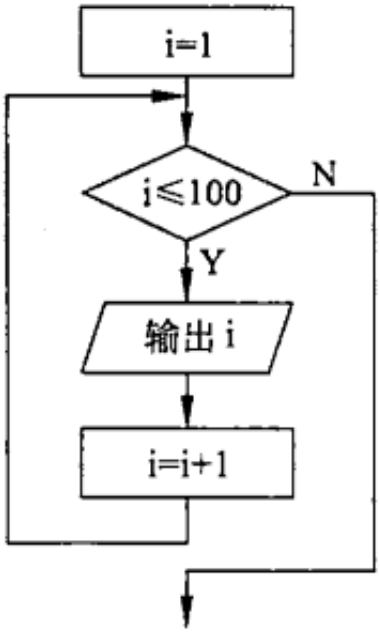
`for`语句的一般形式为：

```c
for(表达式1；表达式2；表达式3)
  语句
```

3个表达式的主要作用是：

- 表达式1：设置初始条件，只执行一次，可以为0个、1个或多个变量设置初值；
- 表达式2：是循环条件表达式，用来判定是否继续循环，在每次执行循环体前先执行此表达式，决定是否继续执行循环；
- 表达式3：作为循环的调整，例如使循环变量增值，它是在执行完循环体后才进行的；

```c
for (int i = 1; i <= 100; i++) {
      printf("%d",i);
}
```



5.5 循环的嵌套

```
(1) while()
    { ⋮
      while()
        {…}      } 内层循环
    }

(2) do
    { ⋮
        do
          {…}    } 内层循环
        while()
    } while()

(3) for(;;)
    {
      for(;;)
        {…}      } 内层循环
    }

(4) while()
    { ⋮
      do
        {…}      } 内层循环
      while();
        ⋮
    }

(5) for(;;)
    { ⋮
      while()
        {…}      } 内层循环
        ⋮
    }

(6) do
    {
        ⋮
      for(;;)
        {…}      } 内层循环
    }while();
```

5.6 几种循环的比较

凡用`while`循环能够完成的，用`for`循环都能实现。

5.7 改变循环执行的状态

以上案例都是根据事先指定的循环条件正常执行和终止循环，但有时当出现某种情况，需要提早结束正在执行的循环操作。

5.7.1 用`break`语句提前终止循环

`break`：终止循环

例5.4 在全系1000学生中，征集慈善募捐，当总数达到10万元时就结束，统计此时捐款的人数，以及平局每人捐款的数目。

```
#include <stdio.h>
#define SUM 100000
int main(){
    float amount,average,total;
    int i;
    for (i = 1,total = 0; i<=1000; i++) {
        printf("请输入第%d个人的捐款数额：",i);
        scanf("%f",&amount);
        total += amount;
        if(total >= SUM){
            break;
        }
    }
    average = total / i;
    printf("共有%d人捐款，累计捐款金额为%.1f，平均每人捐款%.1f\n",i,total,average);
    return 0;
}
```

5.7.2 用`continue`语句提前结束本次循环

`continue`：跳出当前循环，进入下一次循环；

例5.5 要求输出100～200之间的不能被3整除的数。

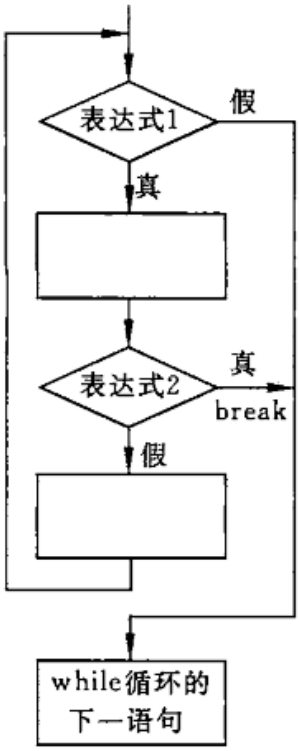
```c
#include <stdio.h>
int main(){
    for (int i = 100; i <= 200; i++) {
        if(i%3==0){
            continue;
        }
        printf("%5d",i);
    }
    printf("\n");
    return 0;
}
```

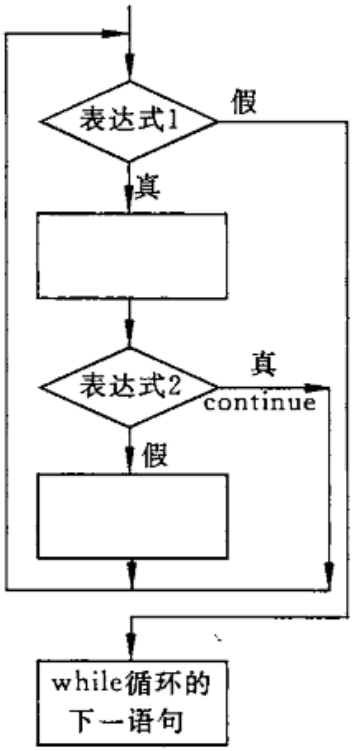
5.7.3 `break`语句和`continue`语句的区别

- 代码块

```
(1) while(表达式 1)
    {
      ⋮
    if(表达式 2)break;
      ⋮
    }
(2) while(表达式 1)
    {
      ⋮
    if(表达式 2) continue;
      ⋮
    }
```

- 对应流程图：





例5.6 输入4*5的矩阵

```c
#include <stdio.h>
int main(){
    int i,j,n=0;
    for (i = 1; i<=4; i++) {
        for (j = 1; j<=5; j++,n++) {
            if (n%5==0) {
                printf("\n");
            }
            printf("%d\t",i*j);
        }
    }
    printf("\n");
    return 0;
}
```

- `break`示例

```c
#include <stdio.h>
int main(){
    int i,j,n=0;
    for (i = 1; i<=4; i++) {
        for (j = 1; j<=5; j++,n++) {
            if (n%5==0) {
                printf("\n");
            }
```

```
            if(i==3&&j==1){
                break;
            }
            printf("%d\t",i*j);
        }
    }
    printf("\n");
    return 0;
}
```

- `continue`示例

```
#include <stdio.h>
int main(){
    int i,j,n=0;
    for (i = 1; i<=4; i++) {
        for (j = 1; j<=5; j++,n++) {
            if (n%5==0) {
                printf("\n");
            }
            if(i==3&&j==1){
                continue;;
            }
            printf("%d\t",i*j);
        }
    }
    printf("\n");
    return 0;
}
```

5.8 循环程序举例

例5.7 乘法口诀表

```
#include <stdio.h>
int main(){
    for (int i = 1; i <= 9; i++) {
        for (int j = 1; j <= i; j++) {
            printf("%dx%d=%d\t",j,i,i*j);
        }
        printf("\n");
    }
}
```

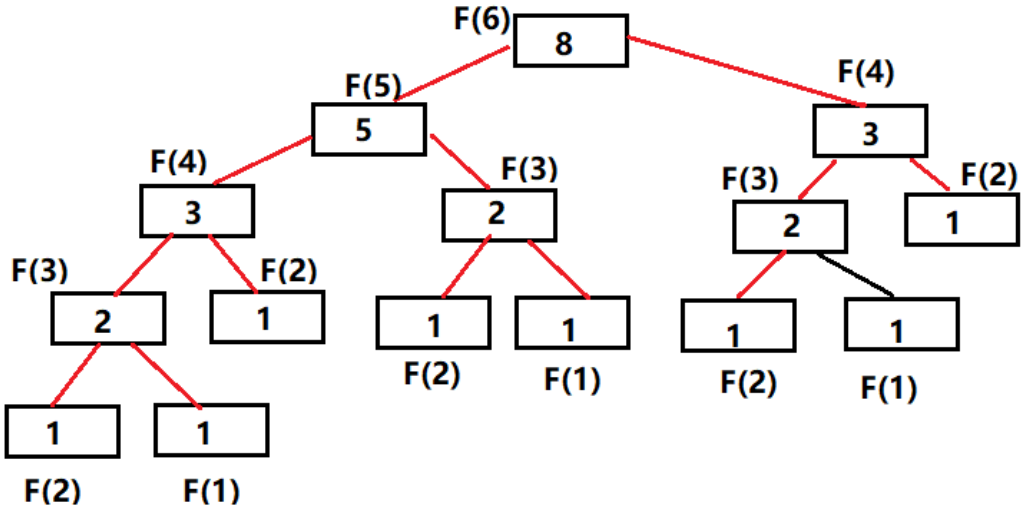
例5.8 斐波那契数列

- 方法一

```c
#include <stdio.h>
int main(){
    int f1 = 1,f2 = 1,f3;
    int i;
    printf("%12d\n%12d\n",f1,f2);
    for (i = 1; i <= 38; i++) {
        f3 = f1+f2;
        printf("%12d\n",f3);
        f1 = f2;
        f2 = f3;
    }
    return 0;
}
```

- 方法二：`F(0) = 0` `F(1) = 1` `F(n) = F(n-1)+F(n-2)`

```c
#include<stdio.h>
int fib(int n)
{
    if(n==1||n==2)
        return 1;
    else
        return fib(n-1)+fib(n-2);
}
int main()
{
    int n;
    printf("请输入需要查询的斐波那契数列索引：");
    scanf("%d",&n);
    printf("斐波那契数列中第%d项为%d\n",n,fib(n));
    return 0;
}
```



例5.9 翻译密码

```c
#include<stdio.h>
int main(){
    char c;
    c = getchar();
    while (c!='\n') {
        if((c>='a' && c<= 'z')||(c>='A' && c<='Z')){
            if((c>='W' && c<='Z')||(c>='w' && c<='z')){
                c = c-22;
            }else{
                c = c+4;
            }
        }
        printf("%c",c);
        c = getchar();
    }
    printf("\n");
    return 0;
}
```

课后题：3、4、5、6、7、8、9、10、16