

Java语言与系统设计

第3讲 流程控制

- ☐ if语句
 - ☐ switch语句
 - ☐ for语句
 - ☐ while语句
 - ☐ do-while语句
 - ☐ 跳转类语句
-

1. if 语句

□ if语句格式定义:

if (<表达式>) <语句1> [else <语句2>]

作为条件使用的<表达式>是结果为逻辑值的关系表达式或逻辑表达式，<语句1>是if语句中的if子句，<语句2>是if语句中的else子句。

if语句格式举例：

- ☐ (1) **`if(x!=-1) c++;`**
 - ☐ (2) **`if(x<=a) s1+=x; else s2+=x;`**
 - ☐ (3) **`if(grade>=60 && grade<=100)`
`System.out.println("pass");`**
 - ☐ (4) **`if(x*x+y*y==z*z) {c++; w=x+y+z;}`**
-

2. switch 语句

```
switch ( <表达式> ) {  
    case <常量表达式1>: <语句序列1>  
    case <常量表达式2>: <语句序列2>  
        :  
    [default: <语句序列n>]  
}
```

2. switch 语句

- ❑ 语句可以带有若干个case子句，但只能带有一个default子句。
 - ❑ 当没有匹配的case时，执行default子句
 - ❑ 每个case子句中都带有一个常量表达式，接着为冒号，接着为若干条语句，包括不含有任何语句的情况。
 - ❑ 通常在每个case子句的最后使用一条break语句，以便结束整个switch语句的执行过程，否则将接着执行下面case子句中。
-

```
switch(ch) {    //ch是字符变量
    case 'a':
    case 'A': System.out.println("优秀"); break;
    case 'b':
    case 'B': System.out.println("良好"); break;
    case 'c':
    case 'C': System.out.println("一般"); break;
    default: System.out.println("较差");
}
```

3. for 语句

for语句格式定义:

- `for (<表达式1>; <表达式2>; <表达式3>) <语句>`
 - 关键字for后面圆括号内的<表达式1>称为初值表达式, 为循环变量赋初值, 同时兼有定义变量的功能, <表达式2>称为控制表达式, 它必须是一个关系或逻辑表达式, 判断循环变量的值是否在允许的范围内, <表达式3>称为修改表达式, 用来修改循环变量的值, for和后面的括号合起来称为循环头, 循环头后面的<语句>称为循环体, 循环体可以是简单语句, 也可以是复合语句。
-

for语句的格式举例：

□ (1) `for(i=0; i*i <10; i++)`
`System.out.print(i*i+" ");`

`//0 1 4 9`

□ (2) `for(int j=1; j<=n; j++)`
`{a+=j; b*=j;}` //累加、累乘

□ (3) `for(int k=1; k<=5; k+=2) s+=k*(k+1);`
`//1*2+3*4+5*6=44`

4. while 语句

while语句格式定义:

- while (<表达式>) <语句>
 - 关键字while和后面的括号为循环头，再后面的<语句>为循环体，循环头中的表达式为循环控制表达式，它是一个关系式或者逻辑式，取逻辑值真或假。
-

while语句的格式举例：

□ **(1) while (x>=10) {s+=x; x--;}**

//当x的值小于10时结束循环

□ **(2) while (aa[i]>-1)**
 { if(aa[i]>=60) c1++;
 else c2++;
 i++;}

while语句编程举例

- 要求编写一个静态成员函数maxCom，采用辗转相除法求出两个整数参数的最大公约数。
 - 分析：设两个整数参数分别为a和b，a除以b的整余数为r，若 $r==0$ ，则表明b能被a整除，则b就是它们的最大公约数，否则，将b的值赋给a，r的值赋给b，再接着上述过程，依次循环，直到整余数r为0时止，此时的b的值就是原有两个整数的最大公约数。
-

```
public static int maxCom(int a, int b)
{ while(a<=0 || b<=0) {
    System.out.println("参数错误!");
    System.exit(1);
}
    int r=a%b;
    while(r!=0) {
        a=b; b=r;
        r=a%b;
    }
    return b;
}
```

5. do while 语句

do <语句> while (<表达式>);

- 关键字do和while之间的<语句>为循环体，关键字while后面圆括号的<表达式>为循环控制表达式，它是一个关系式或逻辑式，取逻辑值真或假。
-

语句格式举例：

❑ `do s+=i++; while (s<100);`

//直到s的值大于等于100为止

❑ (2) `do`

`{System.out.print(a[i]+ " ");`
`s+=a[i++];} while (i<a.length);`

三种循环语句的比较：

- ❑ (1) for和 while循环语句是先判断循环条件（即求循环控制表达式的值）再执行循环体，而do-while循环是先执行循环体，然后再判断循环条件。因此，for和while循环语句的循环体可能不会被执行一次，而do-while循环的循环体至少被执行一次。
-



三种循环语句的比较：

- (2) 一般情况下，它们之间可以相互转换，编程时可以任选使用，通常对于能够事先确定循环次数的处理问题，使用for循环较简单和方便。
- (3) 在任一种循环的循环体内都可以使用break语句使之终止循环的执行，使用continue语句结束一次循环体的执行，进入下一轮的循环过程。



6. 跳转语句

break语句

- 语句格式：通常使用该语句只包含语句关键字 break，不包含任何语句体。
- 语句功能：从任一种循环语句和分情形switch语句中退出执行，接着执行其后面的语句。
- 语句应用举例：

```
for(int i=2; i<=20; i+=2)  
{s+=i; if(s>=30) break;}
```



6. 跳转语句

continue语句

- 语句格式：该语句只包含语句关键字continue，不包含任何语句体。
 - 语句功能：用于任一种循环语句中，结束一次循环体的执行，接着执行下一轮循环。
-

6. 跳转语句

语句应用举例：

```
int []a={3,7,6,9,3,8};
```

```
int k,max=a[0];
```

```
for(k=0; k<a.length; k++) {
```

```
    if(a[k]<=max) continue;
```

```
    max=a[k];
```

```
}
```



return语句

- 语句格式: `return [<表达式>];`
 - 语句功能: 在函数体中当执行到该语句时, 将结束整个函数的调用过程, 返回到调用它的位置, 继续向下执行。当语句中不带有<表达式>时, 则不返回任何值, 否则返回<表达式>的值。
-

三种语句比较：

- ❑ `return`：并非专门用于结束循环的，它的功能是结束一个方法。当一个方法执行到一个`return`语句时，这个方法将被结束。
 - ❑ 与`break`和`continue`不同的是，`return`直接结束整个方法，不管这个`return`处于多少层循环之内
-

三种语句比较：

- ❑ break只能用于switch语句和循环语句中。continue只能用于循环语句中。二者功能类似，但continue是终止本次循环，break是终止本层循环。
 - ❑ break、continue之后不能有其他的语句，因为程序永远不会执行其后的语句。
-

练习题：

□ 下列循环语句的循环次数是（ ）。

```
int i=5;
```

```
do { System.out.println(i--);
```

```
    i--;
```

```
}while(i!=0);
```

A . 5

B . 无限

C . 2

D . 3

练习题：

找出下面代码的错误部分，说明错误类型及原因，并更正。

❑ (1) `int x = 1;`
 `while (x <= 10);`
 `{ i++; }`

❑ (2) `switch (n) {`
 `case 1: system.out.println("The name is 1");`
 `case 2: system.out.println("The name is 2");`
 `break;`
 `}`
