



计算机学院

北华航天工业学院
NORTH CHINA INSTITUTE OF AEROSPACE ENGINEERING

第3章 JAVA语言基础-II

本章目标

◆ 程序的控制语句

- 选择语句
- 循环语句
- 跳转语句

程序控制语句

- ◆使用控制语句来产生执行流，从而完成程序状态的改变
- ◆程序控制语句分类
 - 顺序、选择、循环、跳转

顺序语句

- 顺序是一种按照从上到下逐步执行程序的结构，中间没有判断和跳转语句。

- ```
class HelloWorld{
 public static void main(String[] args) {
 int a=1;
 int b=2;
 System.out.println(a+b);
 }
}
```

# 选择语句

## ◆选择语句功能

允许程序根据表达式的结果或变量的状态选择不同的执行路径

## ◆选择语句分类：

单分支选择结构、双分支选择结构、多分支选择结构

## ◆if语句、switch语句

# 选择语句-if语句

语法格式一：

```
if(条件语句) {
 语句或语句块1;
}
```

语法格式二：

```
if(条件语句) {
 语句或语句块1;
} else {
 语句或语句块2;
}
```

语法格式三：

```
if(条件语句1) {
 语句块1;
} else if(条件语句2) {
 语句块2;
}
.....
else if(条件语句n) {
 语句块n;
} else {
 语句块n+1;
}
```



```
public class IfDemo{
 public static void main(String args[]) {
 boolean b1 = true;
 if(b1)
 System.out.println("true");
 else
 System.out.println("false");
 }
}
```



## 思考：输出结果

```
public class IfElseDemo {
 public static void main(String args[]) {
 int state = 3;
 if(state==1) {
 System.out.println("1...");
 }
 else if(state==2) {
 System.out.println("2...");
 }
 else if(state==3) {
 System.out.println("3...");
 }
 }
}
```



# 选择语句-switch语句

```
switch(expr) {
 case const1:
 statement1;
 break;

 case constN:
 statementN;
 break;
 [default:
 statement_dafault;
 break;]
}
```

# 选择语句-switch语句

## ◆switch语句规则

- 表达式expr的返回值必须是下述几种类型之一：int, byte, char, short;
- case子句中的值constant 必须是常量，且所有case子句中的值应是不同的；
- default子句是任选的；
- break语句用来终止switch语句的执行（在一些特殊情况下，多个不同的case值要执行一组相同的操作，这时可以不用break）。



思考：输出结果？ default是否一定执行

```
public class SwitchDemo {
 public static void main(String[] args) {
 int state = 2;
 switch(state) {
 case 1:
 System.out.println("1....");
 case 2:
 System.out.println("2.....");
 case 3:
 System.out.println("3.....");
 break;
 default:
 System.out.println("default");
 }
 }
}
```



输出结果?

```
public class SwitchDemo2{
 public static void main(String[] args) {
 short state = 2;
 switch(state) {
 case 1:
 case 2:
 case 3:
 System.out.println(“是小于等于3的
数.....”);
 break;
 default:
 System.out.println(“大于3的数”);
 }
 }
}
```



## Lab3-2、完成100分值到5分值转换

1、100分值范围：0~20~40~60~80~100

2、5分值范围：E~D~C~B~A

# 循环语句

## ◆循环语句功能

使程序能够重复执行一条或多条语句

## ◆循环语句的四个组成部分

- 初始化部分 (init\_statement)
- 循环条件部分 (test\_exp)
- 循环体部分 (body\_statement)
- 循环部分 (alter\_statement)

# 循环语句-while语句

## ◆while语句

[initialization]

```
while (termination) {
```

```
 body;
```

```
 [iteration;]
```

```
}
```

注意:

termination是boolean类型的expression;

当termination为true时，进入while，否则退出while



**//打印输出0到9的值**

```
public class WhileDemo1 {
 public static void main(String[] args) {
 int count = 0; //初始化部分
 while(count<10){ //循环条件部分
 System.out.println("count..." + count); //循环体部分
 count++; //循环部分
 }
 }
}
```





```
public class WhileDemo2 {
 public static void main(String[] args) {
 int count = 0;
 while(count>10){
 System.out.println("count..." + count);
 count++;
 }
 System.out.println("count..." + count);
 }
}
```

输出结果？

如果把`while(count>10)`改为`while(true)`的结果？



```
int x = 3;
while (x == 3);
System.out.println("sfsf");
```

添加;后结果是?  
不添加;后结果是?

## 循环语句-do while语句

### ◆do-while语句

[initialization]

do {

body;

[iteration;]

} while (termination);

注意：

无论[initialization] 为何值时，都会执行一次while;  
while (termination); 后面以”；“结束的



输出结果为?

```
public class DoWhileDemo1 {
 public static void main(String[] args) {
 int count=0;
 do {
 System.out.println("count="+count);
 count++;
 } while (count<10);
 }
}
```

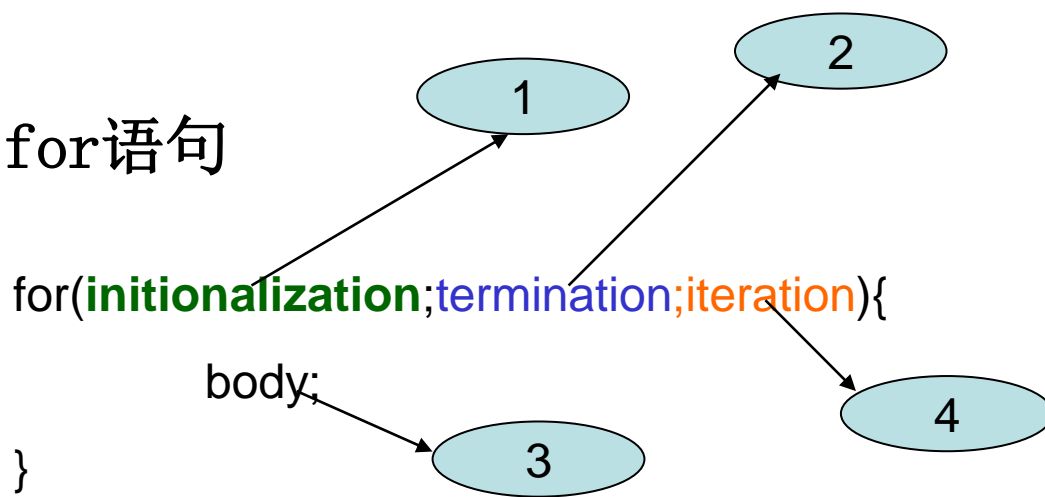


输出结果？

```
public class DoWhileDemo2 {
 public static void main(String[] args) {
 int count=0;
 do {
 System.out.println("count..." + count);
 count++;
 } while (count==10);
 }
}
```

## 循环语句-for语句

### ◆ for语句



- for语句执行时，首先执行初始化操作，然后判断终止条件是否满足，如果满足，则执行循环体中的语句，最后执行循环部分。完成一次循环后，重新判断终止条件。
- 初始化、终止以及循环部分都可以为空语句(但分号不能省)，三者均为空的时候，相当于一个无限循环(dead loop)。
- 在初始化部分和循环部分可以使用逗号语句，来进行多个操作。逗号语句是用逗号分隔的语句序列。



输出结果是？

```
public class ForDemo1 {
 public static void main(String[] args) {
 for(int i=0;i<15;i++) {
 System.out.println("i...."+i);
 }
 }
}
```



输出结果是?

```
for(int i=0;;i++) {
 System.out.println("i..." + i);
}
```

```
for(int i=0;i<10;) {
 System.out.println("i..." + i);
}
```

```
int i=0;
for(;;) {
 System.out.println("i...");
}
```



# 跳转语句

## ◆ 跳转语句功能

允许程序以非线性的方式执行

## ◆ 跳转语句分类:

break语句、continue语句、return语句

# 跳转语句-break语句

## ◆break语句

- 在switch语中，break语句用来终止switch语句的执行。
- 在Java中，可以为每个代码块加一个标号，一个代码块通常是用大括号{}括起来的一段代码。break语句用于终止某个语句块的执行

```
{
 break;

}
```

- 它可被用作goto语句的一种“先进”方式



输出结果?

```
public class BreakDemo1 {
 public static void main(String[] args) {
 int i=0;
 while(i<=100) {
 if(i==10) break;
 i++;
 System.out.println("i= " +i);
 }
 }
}
```

## 跳转语句-break语句

- ◆break语句出现在多层嵌套的语句块中时，可以通过标签指明要终止的是哪一层语句块

```
label1: {

 break label1;

}
```



输出结果?

```
public class BreakDemo2 {
 public static void main(String[] args) {
 lbla: {
 System.out.println("lbla....");
 lblb: {
 System.out.println("lblb....");
 break lbla;
 }
 System.out.println("lbla end...");
 }
 }
}
```

# 跳转语句-continue语句

## ◆continue语句

- 用来结束本次循环，跳过循环体中下面尚未执行的语句，接着进行终止条件的判断，以决定是否继续循环。
- 对于for语句，在进行终止条件的判断前，还要先执行循环语句。它的格式为：

`continue;`

- 用continue跳转到标号指定的外层循环中，格式为：  
`continue outerLabel;`



## Lab3-5

打印输出1~50之间的前10个数

# 跳转语句-return语句

## ◆return语句

- return语句从当前方法中退出，返回到调用该方法的语句处，并从紧跟该语句的下一条语句继续程序的执行。
- return语句通常用在一个方法体的最后，否则会产生编译错误，除非用在if-else语句中。

- 语法格式：

`return expression ;`

或

`return;`





# 总结

## ◆ 选择语句

选择语句分类: if语句、switch语句

## ◆ 循环语句

循环语句分类: while语句, do-while语句, for语句, for-each语句

## ◆ 跳转语句

跳转语句分类: break语句、continue语句、return语句



计算机学院



谢谢！