# 第一周学习笔记

2021年4月21日 20:30

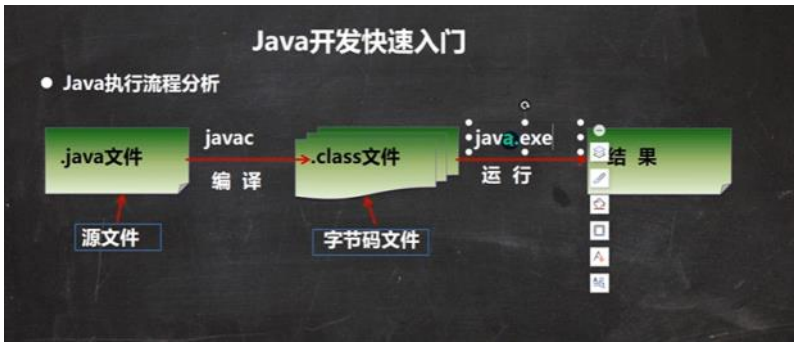
# P1-P12

2021年4月19日 13:29

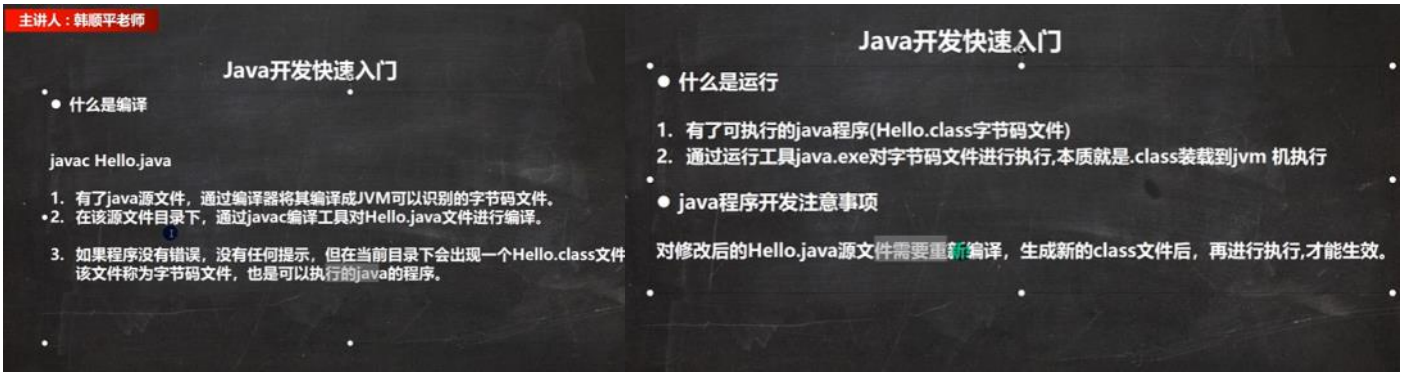
# P13-P24

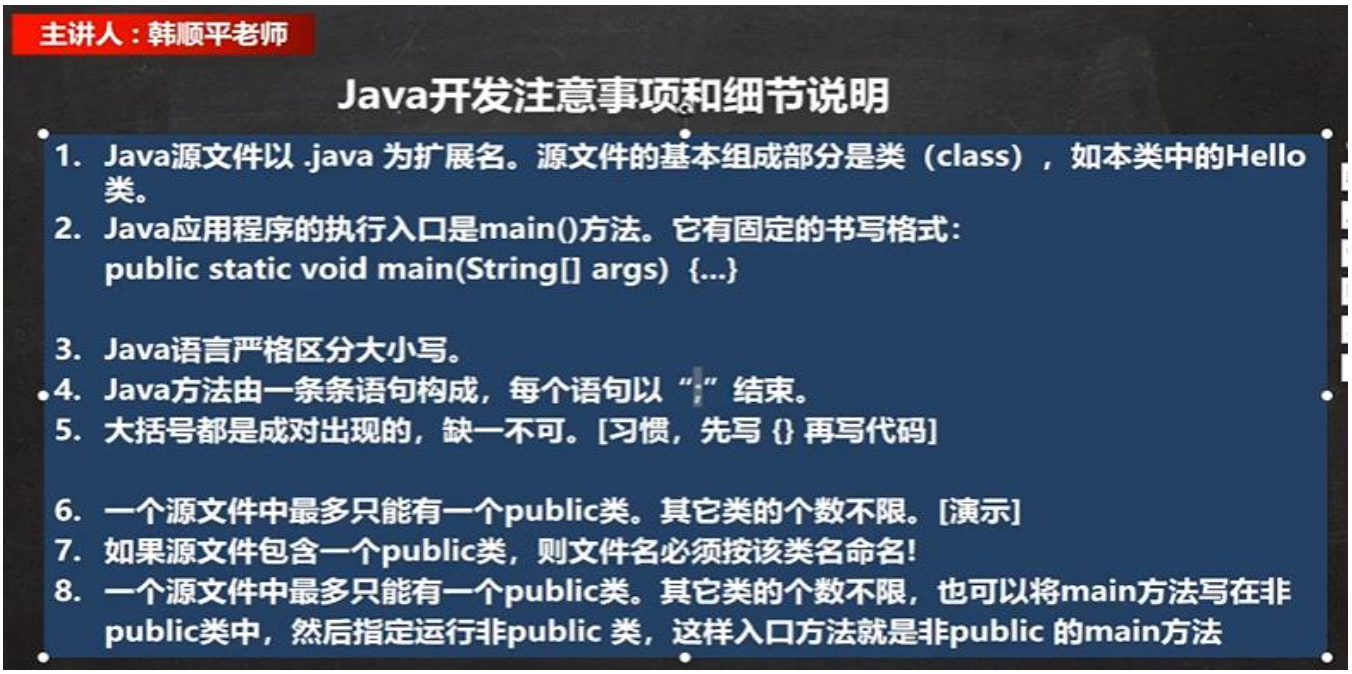
2021年4月21日 12:06

学习并使用 javac、java 命令，了解两个命令各自的使用场景




编译与运行



# P19-20

2021年4月21日 14:54

P19

1. 文件名按照public类命名

。。。。

P20

快速学习技术或知识点经验

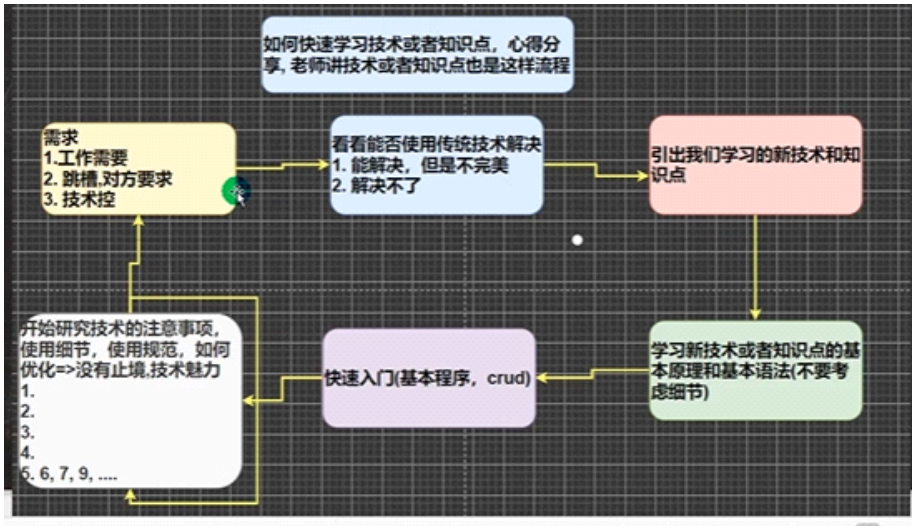
1.需求驱动：学习任务

2.学习技术的基本语句和基本语法（不考虑细节）

3.快速入门案列（基本案例，增删改查）

4.开始研究技术的使用细节、规范与如何优化

优化=》永无止境、技术魅力



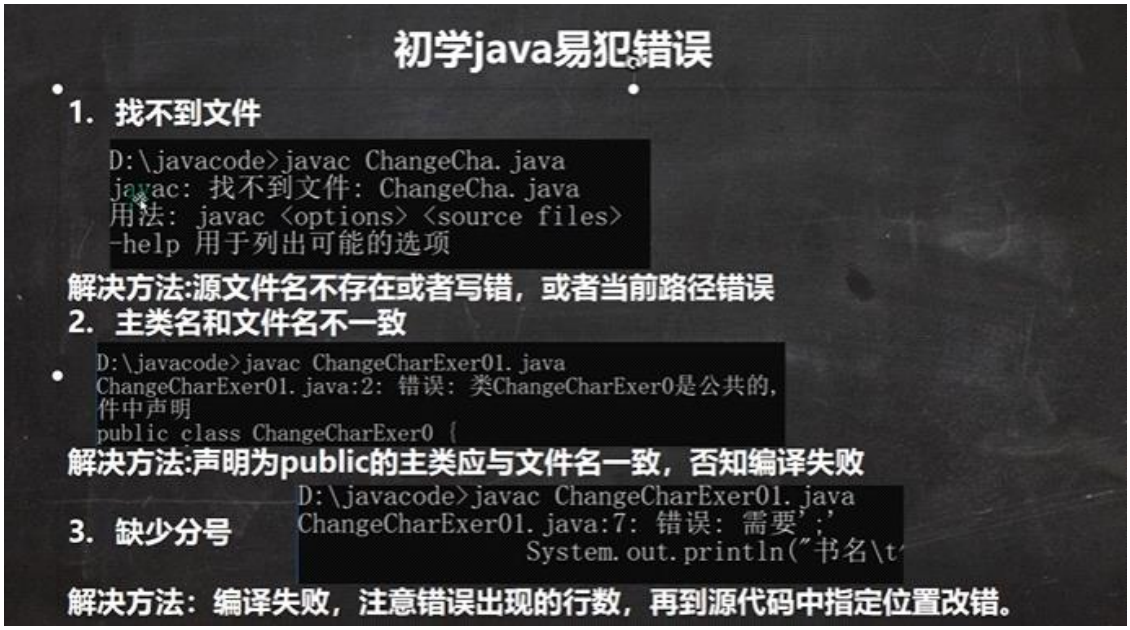
# P21转义字符-22易犯错误

2021年4月21日 18:15

```
        //\t  ：一个制表位，实现对齐的功能
        System.out.println("北京\t天津\t上海");
        // \n  ：换行符
        System.out.println("jack\nsmith\nmary");
        // \\  ：一个\  \\
        System.out.println("C:\\Windows\\System32\\cmd.exe");
        // \"  ：一个"
        System.out.println("老韩说:\"要好好学习java,有前途\"");
        // \'  ：一个'
        System.out.println("老韩说:\'要好好学习java,有前途\'");

        // \r  ：一个回车  System.out.println("韩顺平教育\r北京");
        // 解读
        // 1. 输出  韩顺平教育
        // 2. \r表示回车
        System.out.println("韩顺平教育\r\n北京"); // 北京平教育
    }

}
```




## 1. 举例

1. 1-》l
2. 0-》o
3. 中英文混用；；“"
4. 拼写错误：void-》viod
5. 不好修改的是业务、环境错误

# P23注释P24多行p25文档注释

2021年4月21日 20:31

注释（comment）

用于注解说解释程序的文字就是注释，注释提高了代码的阅读性（可读性）；注释是一个程席员必须要具有的良好编程习惯。将自己的思想通过注释先整理比来，再用代码去体现。

- java中的注释类型

1．单行注释 //…//

2．多行注释 /*…*/

3．文档注释 /**

* * *

*/

- 使用细节

1）被注释的文字，不会JVM(java虚拟机）解释执行

2）多行注释里面不允许有多行注释嵌套

文档注释：

/**

* * *

*/

注释内容可以被JDK提供的工具javadoc所解析，生成一套以网页文形式体现的该程序的说明文档，一般写在类

- 基本格式
- 如何生成对应的文档注释
- 应用实例

Javadoc -d 文件夹名 - xx -yy Dem03.java java文件

java标签名{@author,@version etc..}

# P25代码规范p26dos原理p27路径详解

2021年4月22日 21:03

## Java代码规范

1．类、方法的注释，要以javadoc的方式来写。

2．非JavaDoc的注释(单行与多行)，往往是给代码的维护者看的，着重告述读者为什么这样写，如何修改，注意什么问题等

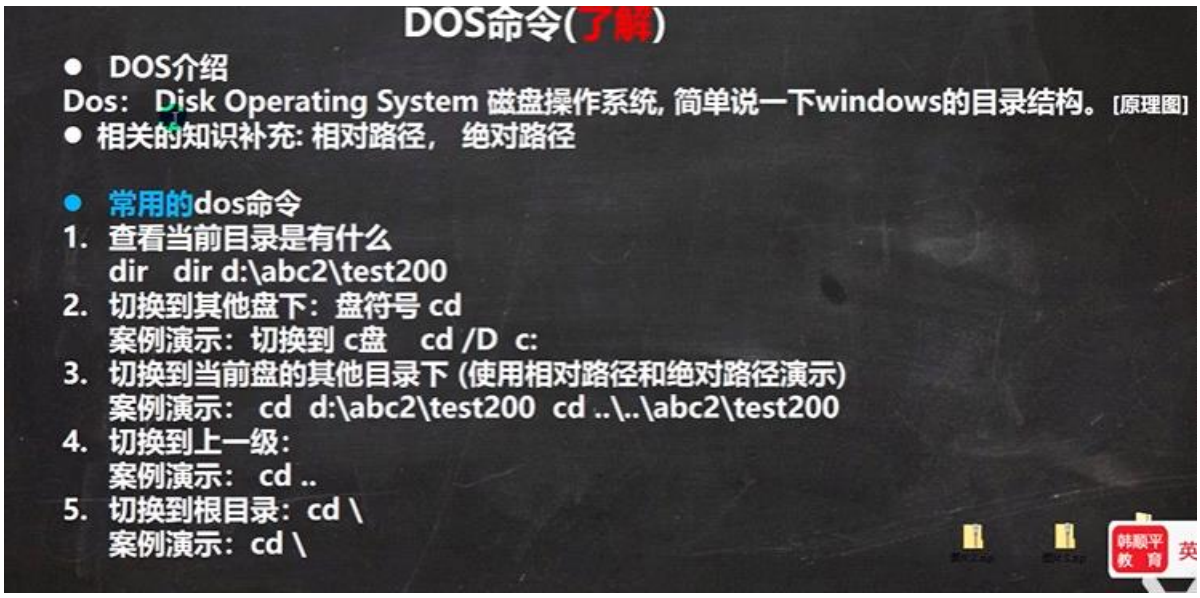
3．使用tab操作，实现缩进，默认整体向右边移动，用shift+tab整体向左移

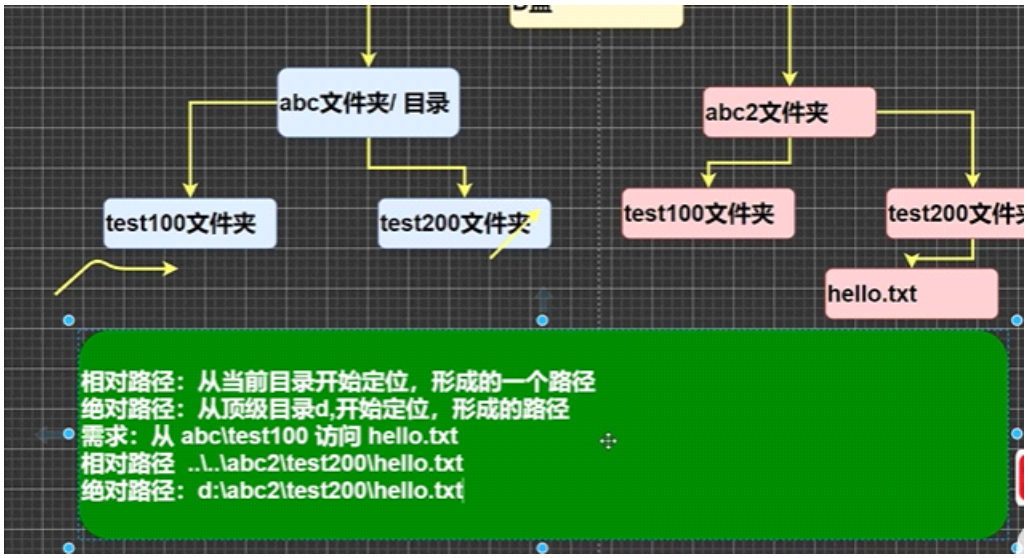
4．运算符和＝两边习惯性各加一个空格。比如：2＋4＊5＋345－89

5．源文件使用utf-8编码

6．行宽度不要超过80字符

7．代码编写次行风格和行尾风格



# P29-31常用dos命令

2021年4月22日 21:42

- dir：查看当前目录包含内容
- cd（change directory）切换

  /D改变根目录（d盘换c盘）

```
:\javacode>cd /D c:

:\>_
```

- 相对，绝对
- 案例演示： cd d:\abc2\test200 cd ..\..\abc2\test200


**6.t**ree tree c:\

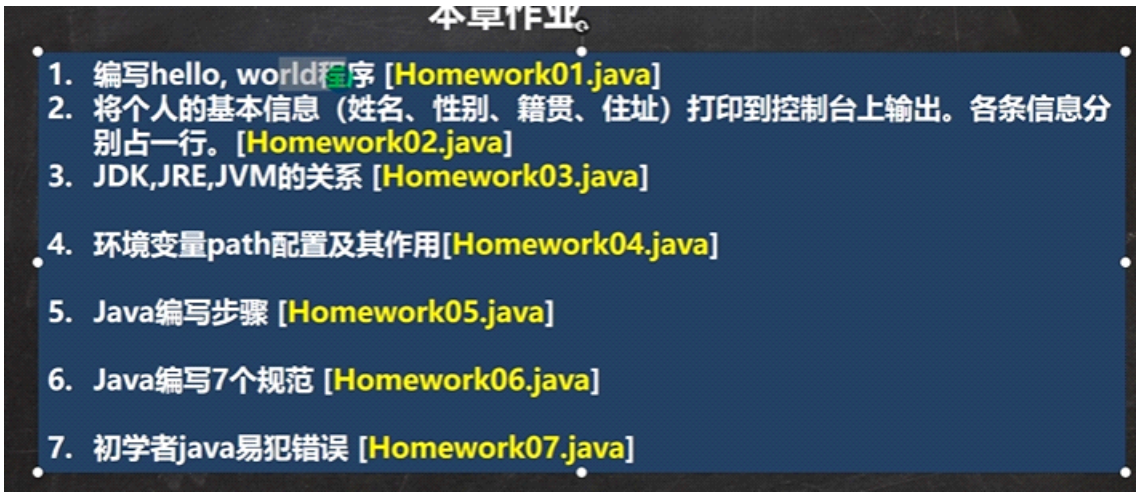
7.cls

8.exit

9.md创建文件rd删除目录copy拷贝文件del删除文件echo输入内容到文件type/move剪切

# P31-33作业+34内容梳理

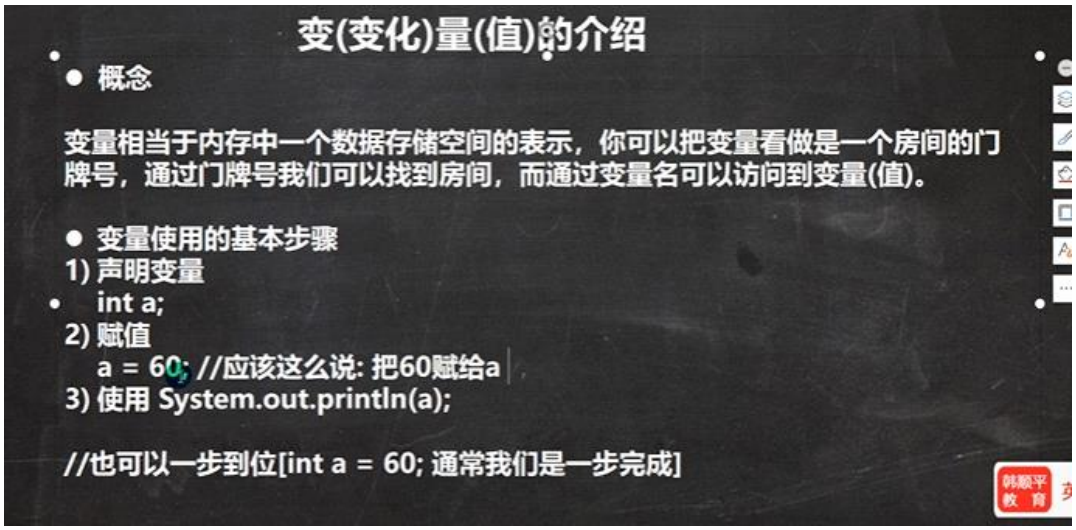
2021年4月22日　　22:03



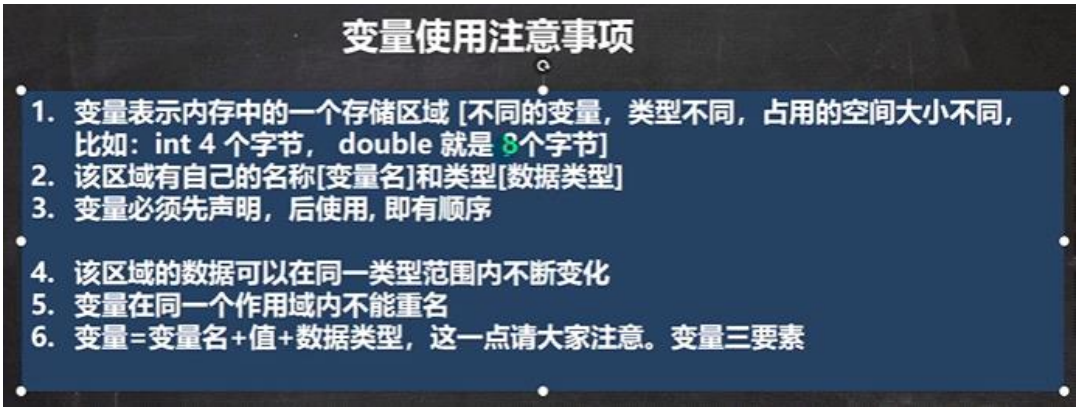

之后的作业先自己写，再看视屏讲解

# p35-38变量

2021年4月23日 15:24

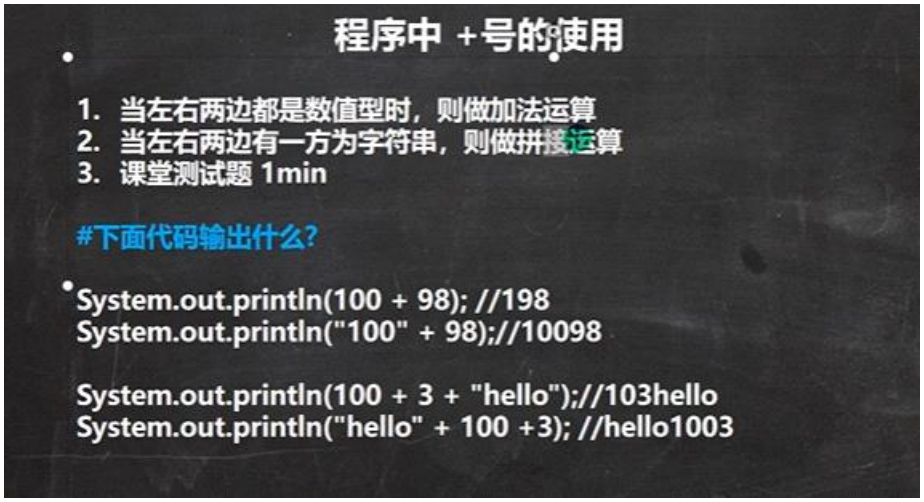



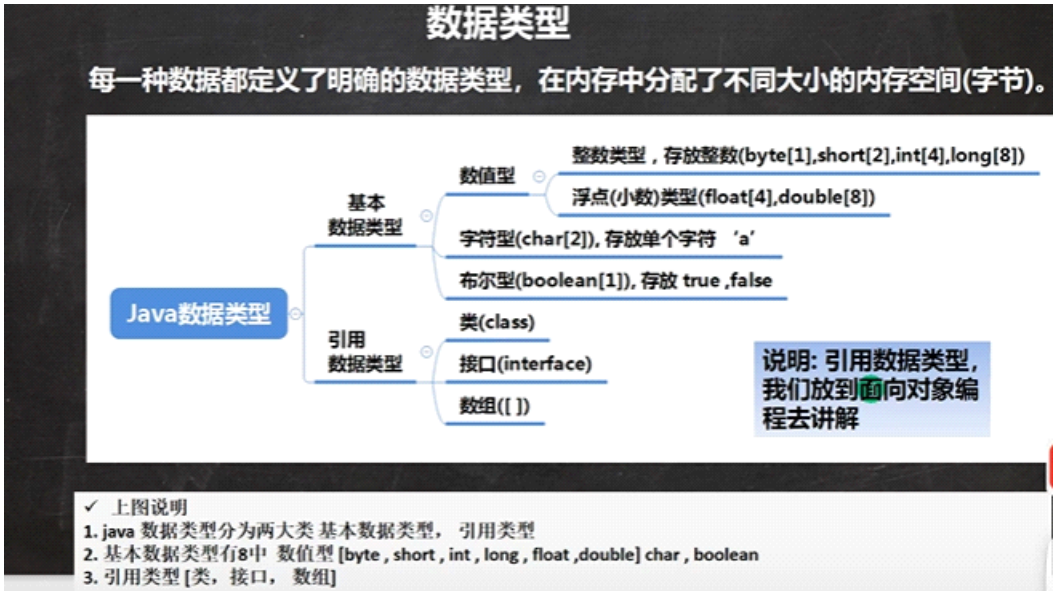




# P39 + 使用 p40数据类型

2021年4月23日 16:26




字符串Srtring属于类

背诵下来

# P41-45数值型（整型、浮点数使用）

2021年4月23日　16:33

## 整数类型

- 整型的类型

| 类　型 | 占用存储空间 | 范围 |
| --- | --- | --- |
| byte [字节] | 1字节 | -128 ~ 127<br>为啥存放的范围是这个=>二进制(二进制我们详解) |
| short [短整型] | 2字节 | $-(2^{15})$ ~$2^{15}$-1<br>-32768 ~ 32767 |
| int [整型] | 4字节 | $-2^{31}$ ~ $2^{31}$-1<br>-2147483648 - 2147483647 |
| long [长整型] | 8字节 | $-2^{63}$ ~ $2^{63}$-1 |

## 整数类型

- 整型的使用细节 IntDetail.java

1. Java各整数类型有固定的范围和字段长度，不受具体OS[操作系统]的影响，以保证java程序的可移植性。
2. Java的整型常量（具体值）默认为 int 型，声明long型常量须后加 'l' 或 'L'
3. java程序中变量常声明为int型，除非不足以表示大数，才使用long
4. bit: 计算机中的最小存储单位。byte:计算机中基本存储单元,1byte = 8 bit。[二进制再详细说，简单举例一个 byte 3 和 short 3 ]

思考题: long 类型，有几个 bit
[8 * 8 = 64 bit]
long n = 3;// 内存中

## 浮点类型

- 浮点型的分类

| 类　型 | 占用存储空间 | 范围 |
| --- | --- | --- |
| 单精度float | 4字节 | -3.403E38 ~ 3.403E38 |
| 双精度double | 8字节 | -1.798E308 ~ 1.798E308 |

- 说明一下

1. 关于浮点数在机器中存放形式的简单说明,浮点数=符号位+指数位+尾数位
2. 尾数部分可能丢失，造成精度损失(小数都是近似值)。

## 浮点类型

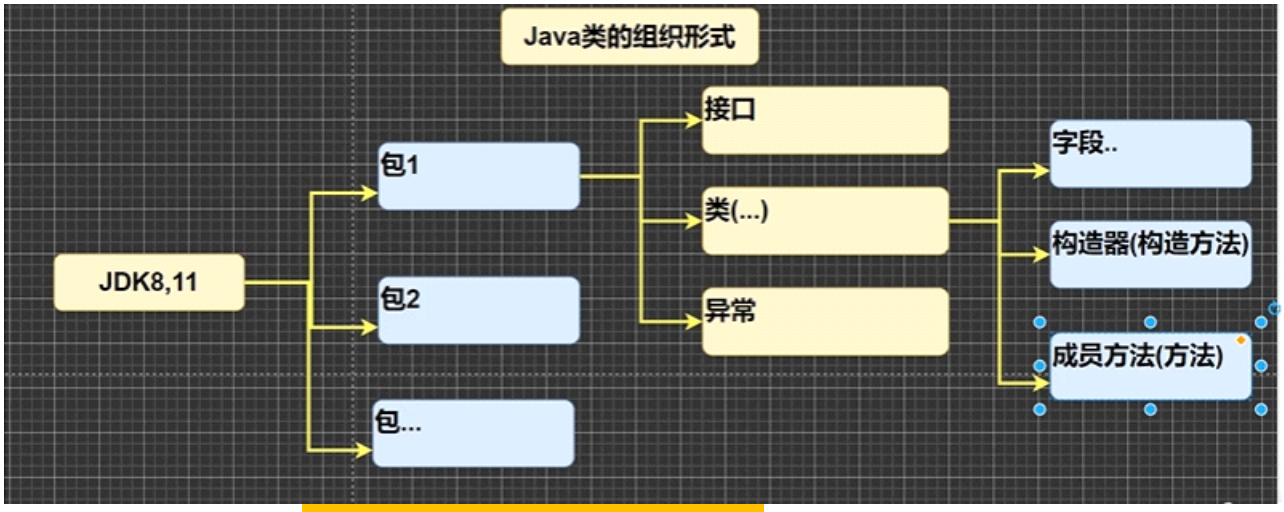
- 浮点型使用细节 FloatDetail.java

1. 与整数类型类似，Java 浮点类型也有固定的范围和字段长度，不受具体OS的影响。[float 4 个字节 double 是 8个字节]
2. Java 的浮点型常量(具体值)默认为double型，声明float型常量，须后加 'f' 或 'F'
3. 浮点型常量有两种表示形式
   十进制数形式：如：5.12　　512.0f　　.512　(必须有小数点)
   科学计数法形式:如：5.12e2 [5.12*10的2次方 ] 5.12E-2　[5.12/10的2次方]
4. 通常情况下，应该使用double型，因为它比float型更精确。[举例说明]
   double num9 = 2.1234567851;
   float num10 = 2.1234567851F;
5. 浮点数使用陷阱: 2.7 和 8.1 / 3 比较

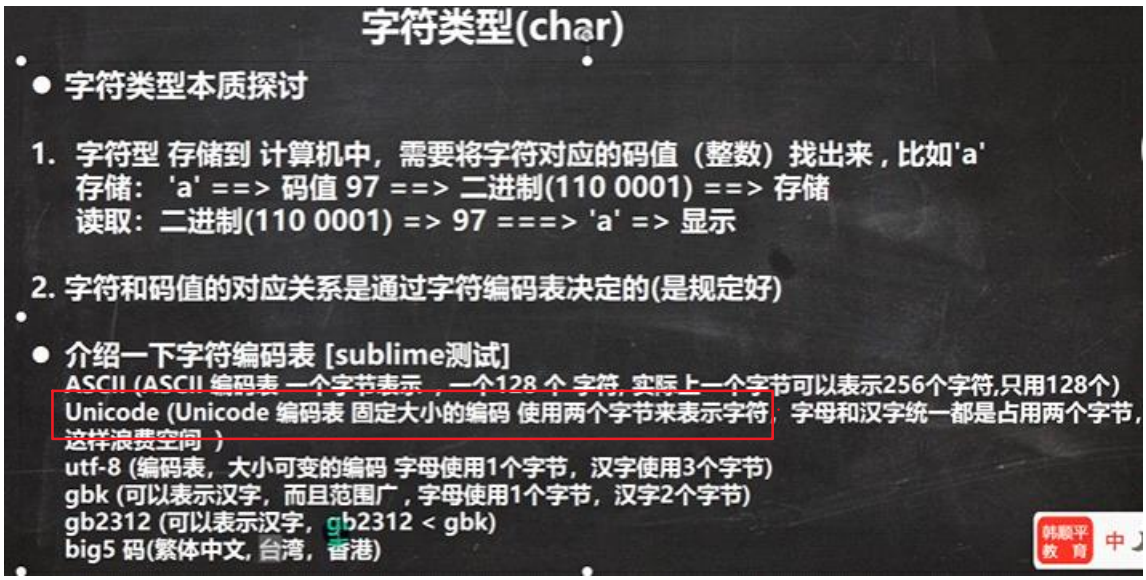
```
double num7 = 2.7;
double num8 = 8.1 / 3;
if( Math.abs(num7 - num8) < 0.00001) {
        System.out.println("相等~~");
}
```

# P46-49JAVA文档与字符型使用

2021年4月24日 19:07

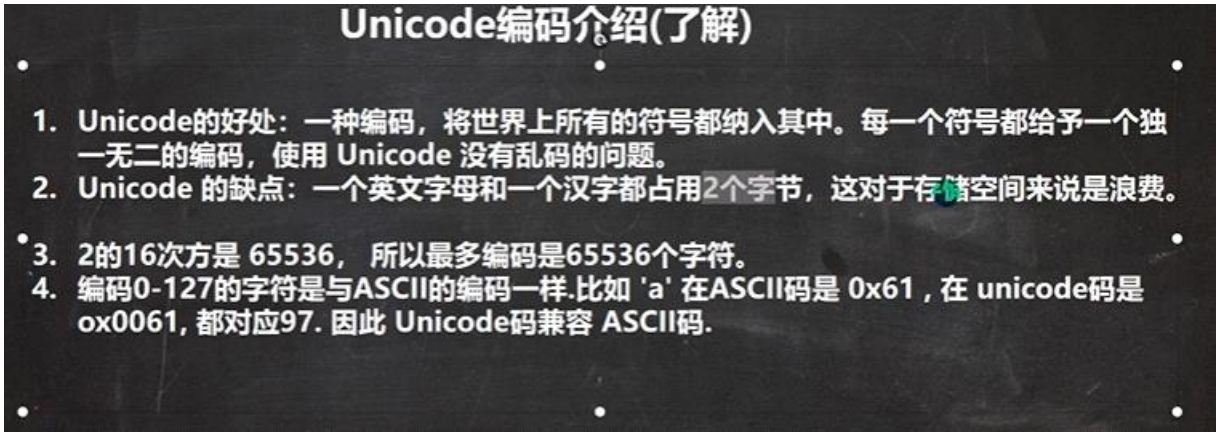


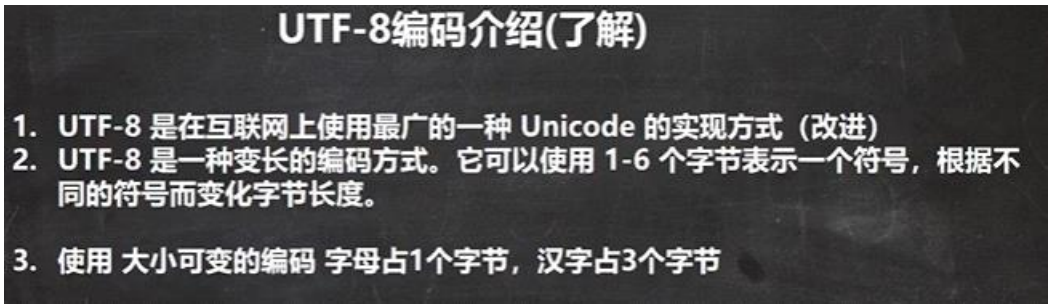

按照包» 类» 方法/或者直接检索

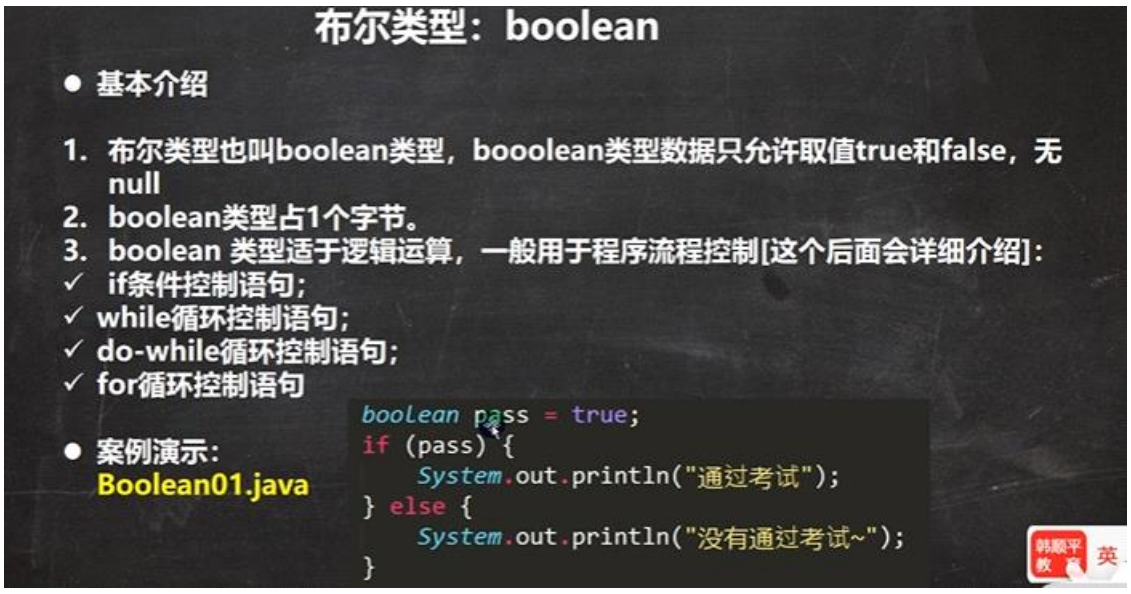



# P50-51常用编码与布尔类型

2021年4月24日　　20:01

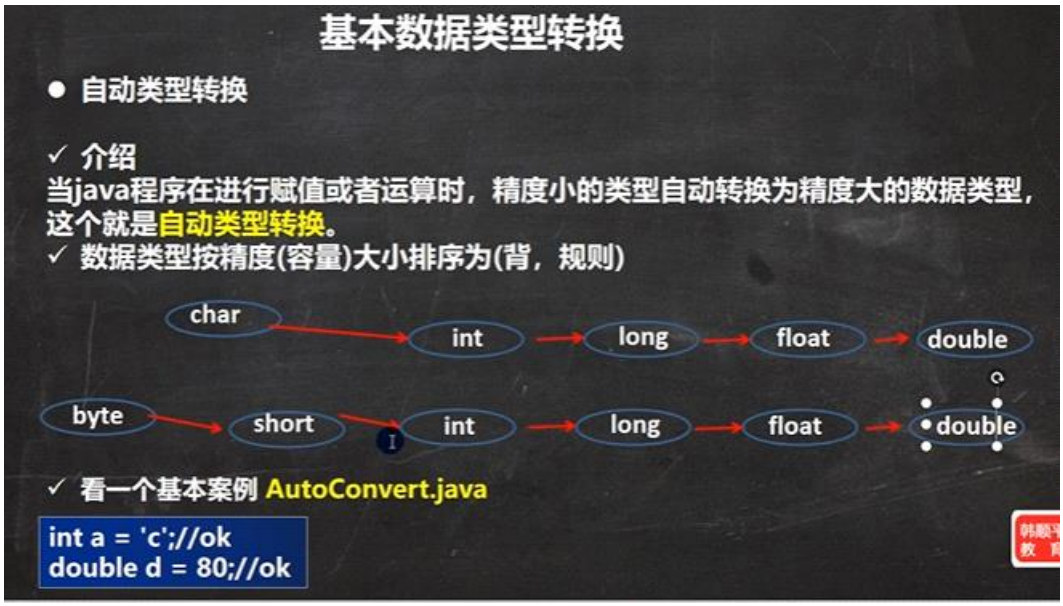


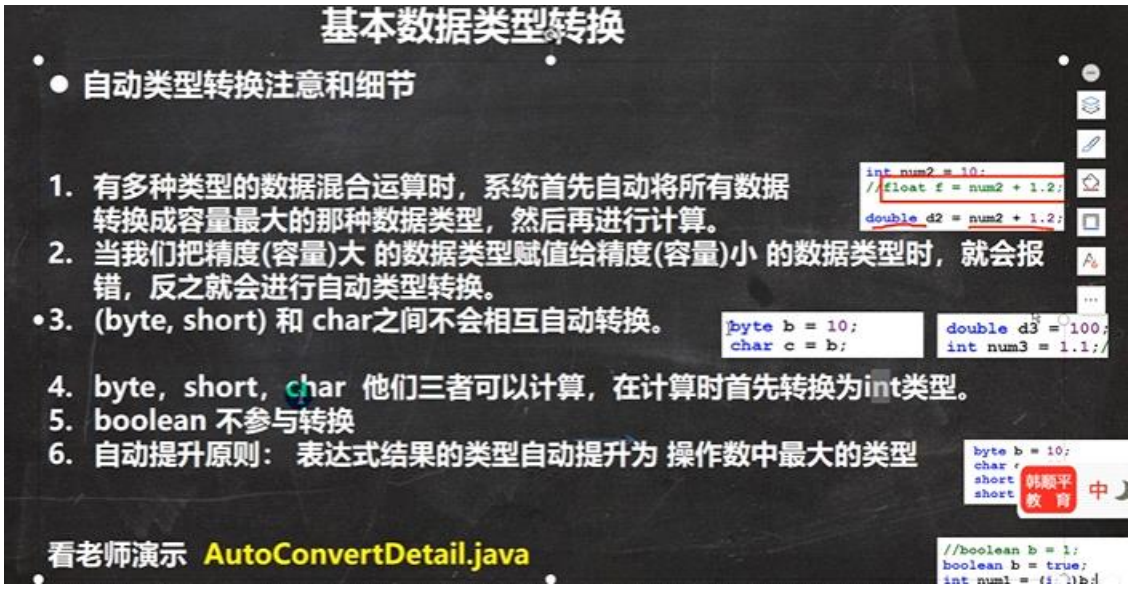



# P52-54自动类型转换

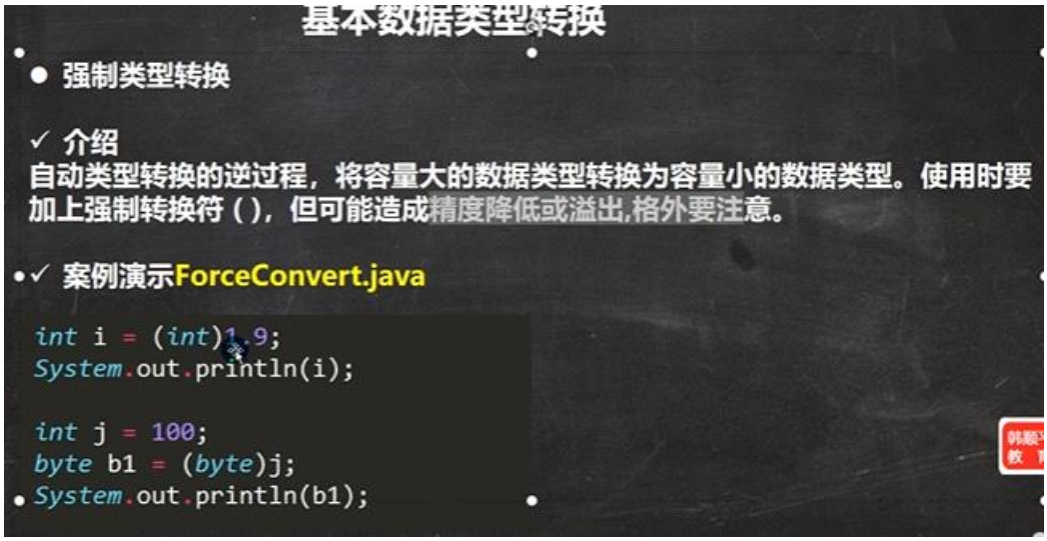
2021年4月24日 20:26

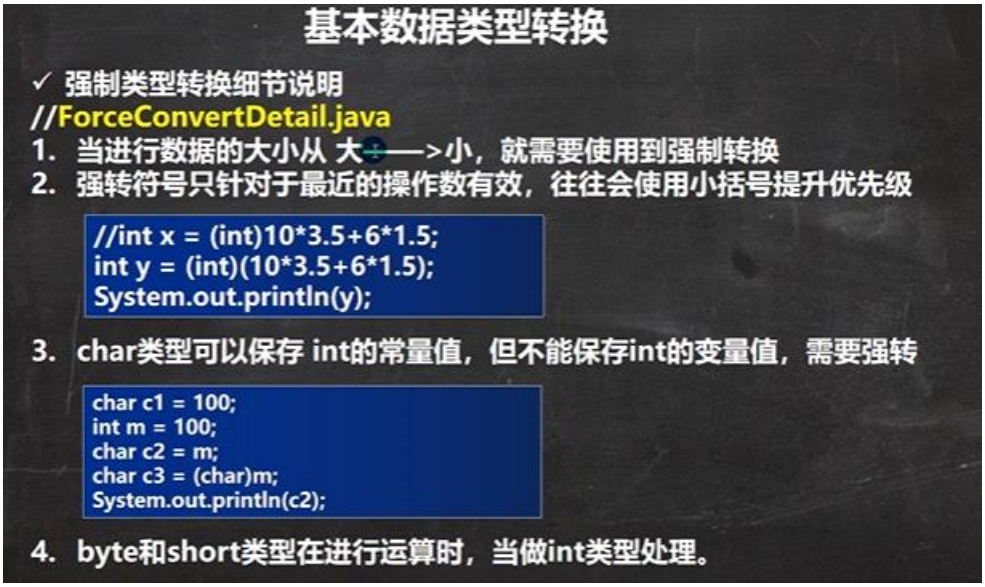


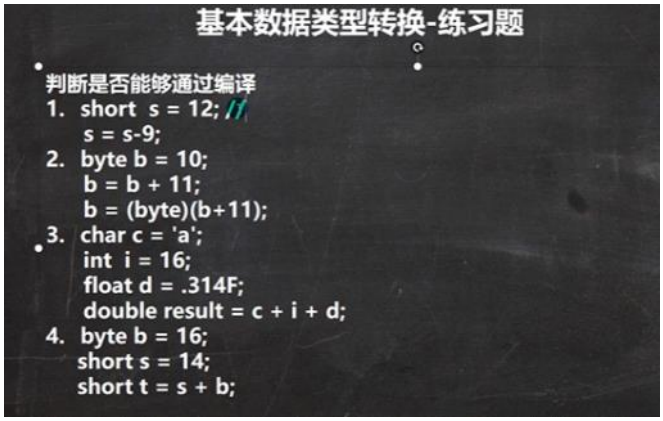



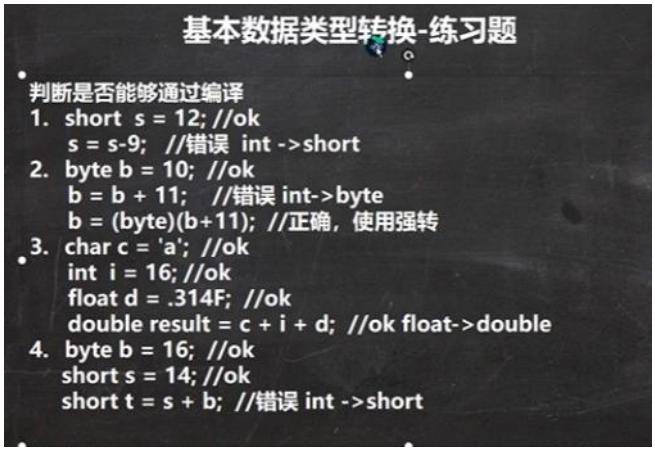
# P55-57强制类型转换

2021年4月24日 22:35



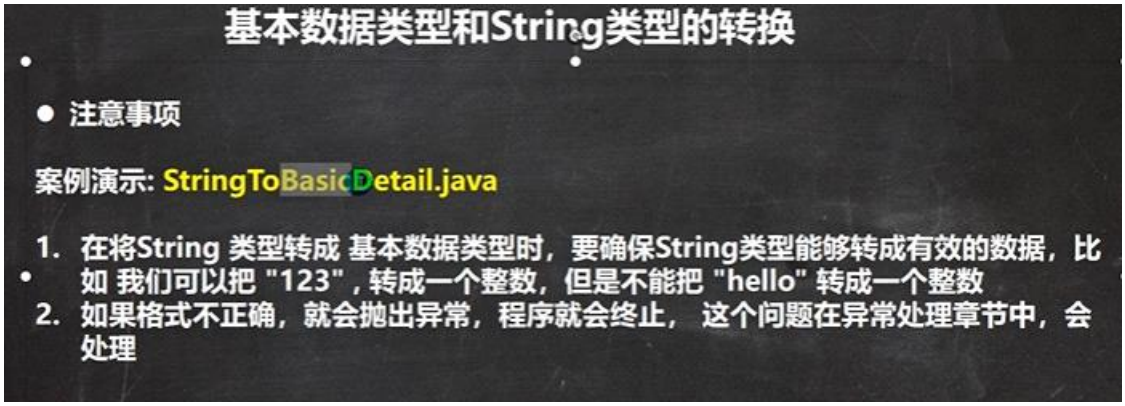



# P58-59string转基本类型

2021年4月24日 22:59

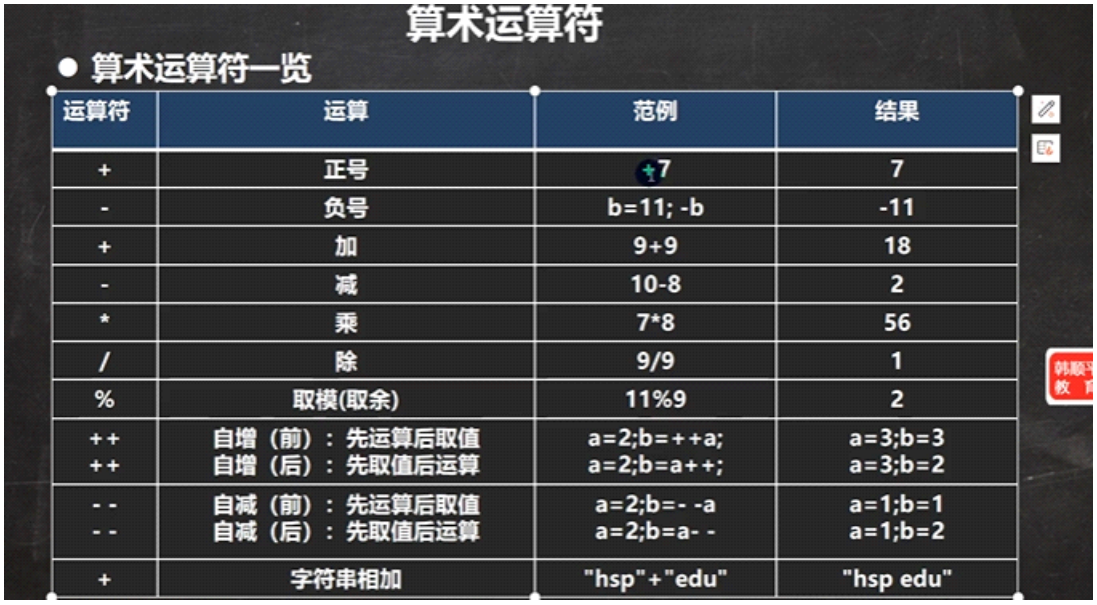



# P60-62作业与小结

2021年4月25日 13:40

# P63-68算术运算符

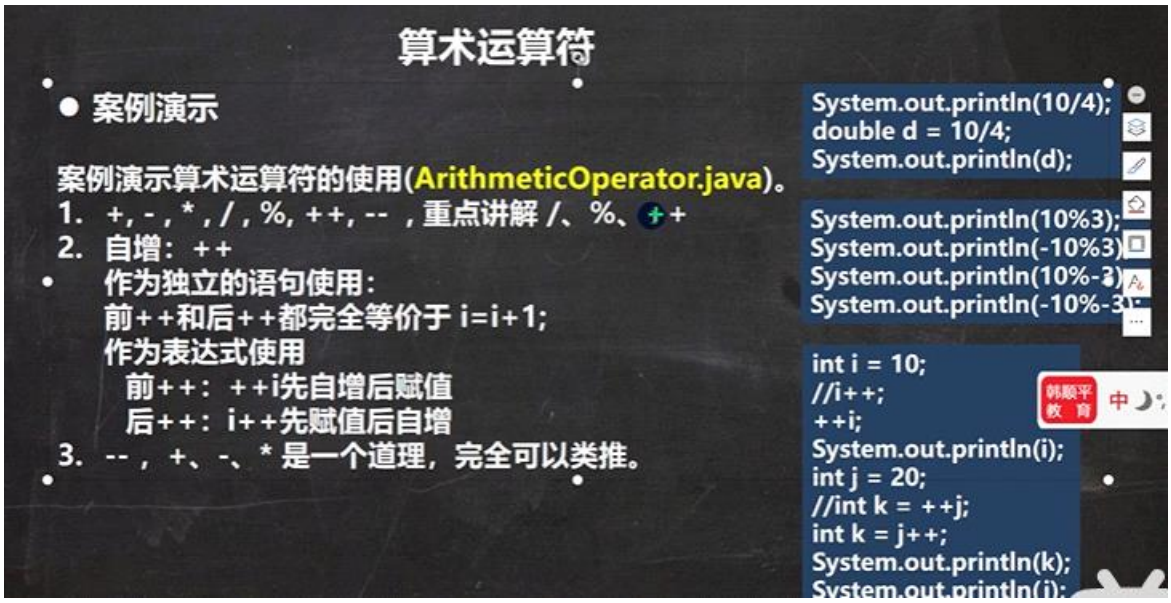
2021年4月25日 19:11



## 算术运算符

● 算术运算符一览

| 运算符 | 运算 | 范例 | 结果 |
|---|---|---|---|
| + | 正号 | +7 | 7 |
| - | 负号 | b=11; -b | -11 |
| + | 加 | 9+9 | 18 |
| - | 减 | 10-8 | 2 |
| * | 乘 | 7*8 | 56 |
| / | 除 | 9/9 | 1 |
| % | 取模(取余) | 11%9 | 2 |
| ++<br>++ | 自增（前）：先运算后取值<br>自增（后）：先取值后运算 | a=2;b=++a;<br>a=2;b=a++; | a=3;b=3<br>a=3;b=2 |
| --<br>-- | 自减（前）：先运算后取值<br>自减（后）：先取值后运算 | a=2;b=- -a<br>a=2;b=a- - | a=1;b=1<br>a=1;b=2 |
| + | 字符串相加 | "hsp"+"edu" | "hsp edu" |



## 算术运算符

● 案例演示

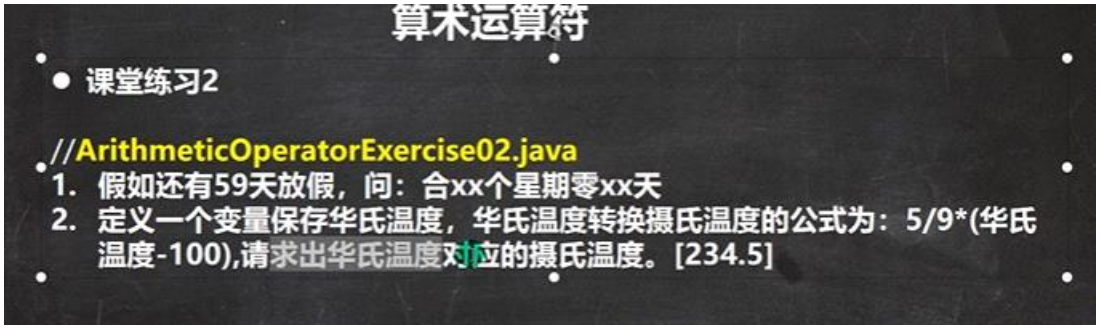
案例演示算术运算符的使用(ArithmeticOperator.java)。

1. +, - , * , / , %, ++, -- ，重点讲解 /、%、++
2. 自增：++
   作为独立的语句使用：
   前++和后++都完全等价于 i=i+1;
   作为表达式使用
   前++：++i先自增后赋值
   后++：i++先赋值后自增
3. -- ， +、-、* 是一个道理，完全可以类推。

```
System.out.println(10/4);
double d = 10/4;
System.out.println(d);

System.out.println(10%3);
System.out.println(-10%3);
System.out.println(10%-3);
System.out.println(-10%-3);

int i = 10;
//i++;
++i;
System.out.println(i);
int j = 20;
//int k = ++j;
int k = j++;
System.out.println(k);
System.out.println(j);
```



## 算术运算符

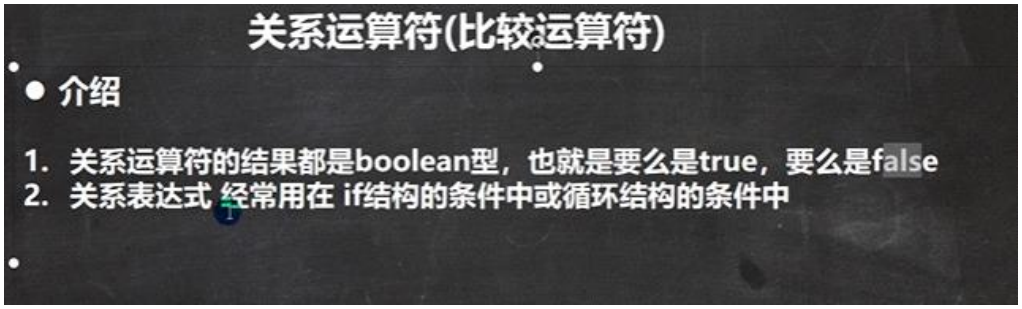
● 课堂练习2

//ArithmeticOperatorExercise02.java

1. 假如还有59天放假，问：合xx个星期零xx天
2. 定义一个变量保存华氏温度，华氏温度转换摄氏温度的公式为：5/9*(华氏温度-100),请求出华氏温度对应的摄氏温度。[234.5]

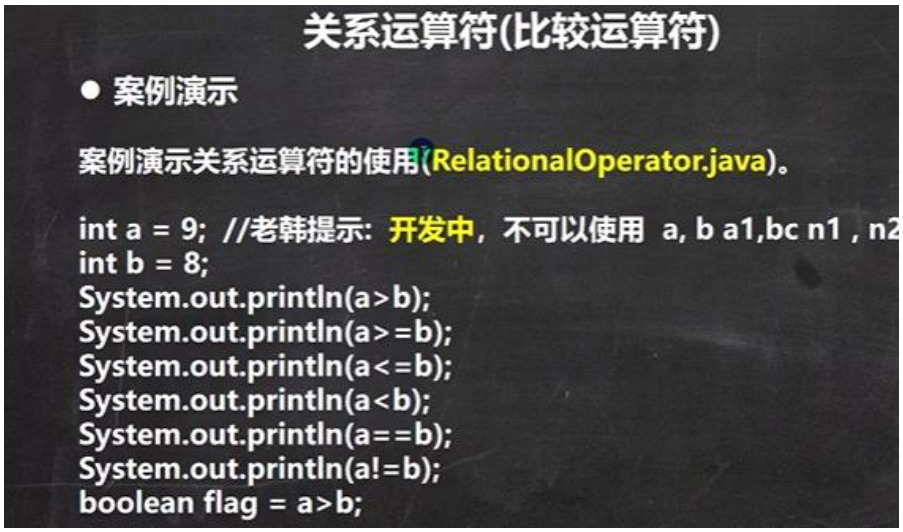
# P69-70关系运算符

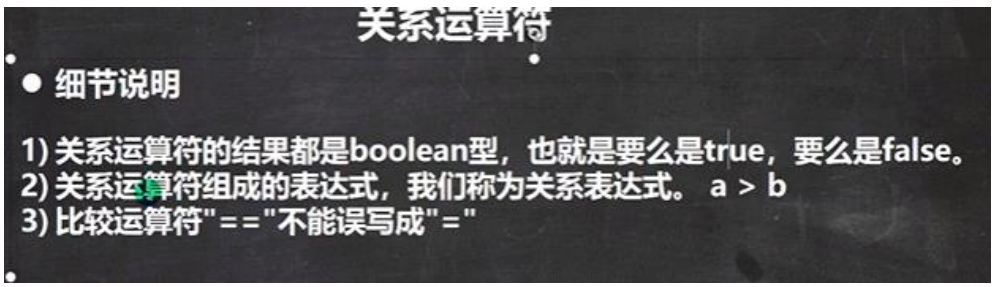
2021年4月25日 20:11





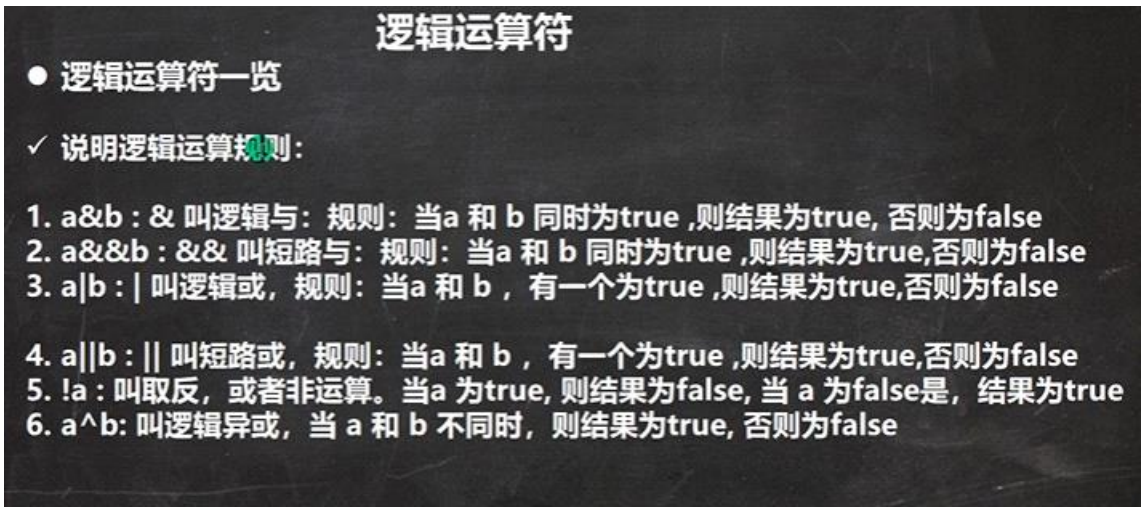
| 运算符 | 运算 | 范例 | 结果 |
|---|---|---|---|
| == | 相等于 | 8==7 | false |
| != | 不等于 | 8!=7 | true |
| < | 小于 | 8<7 | false |
| > | 大于 | 8>7 | true |
| <= | 小于等于 | 8<=7 | false |
| >= | 大于等于 | 8>=7 | true |
| instanceof | 检查是否是类的对象 | "hsp" instanceof String | true |

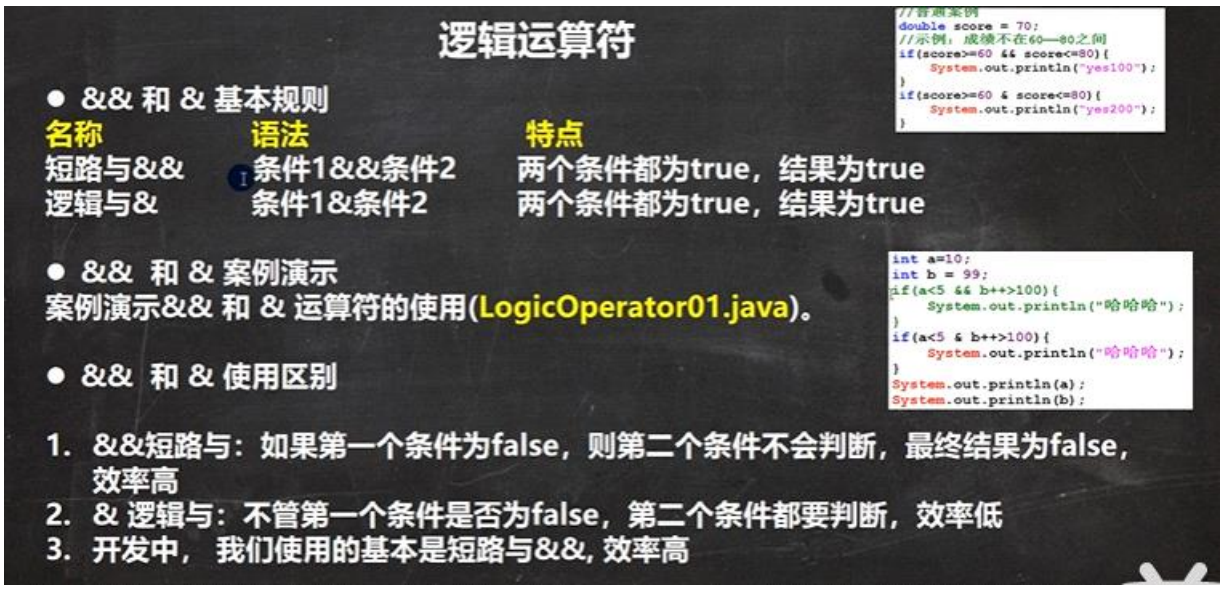


# P71-72逻辑运算符

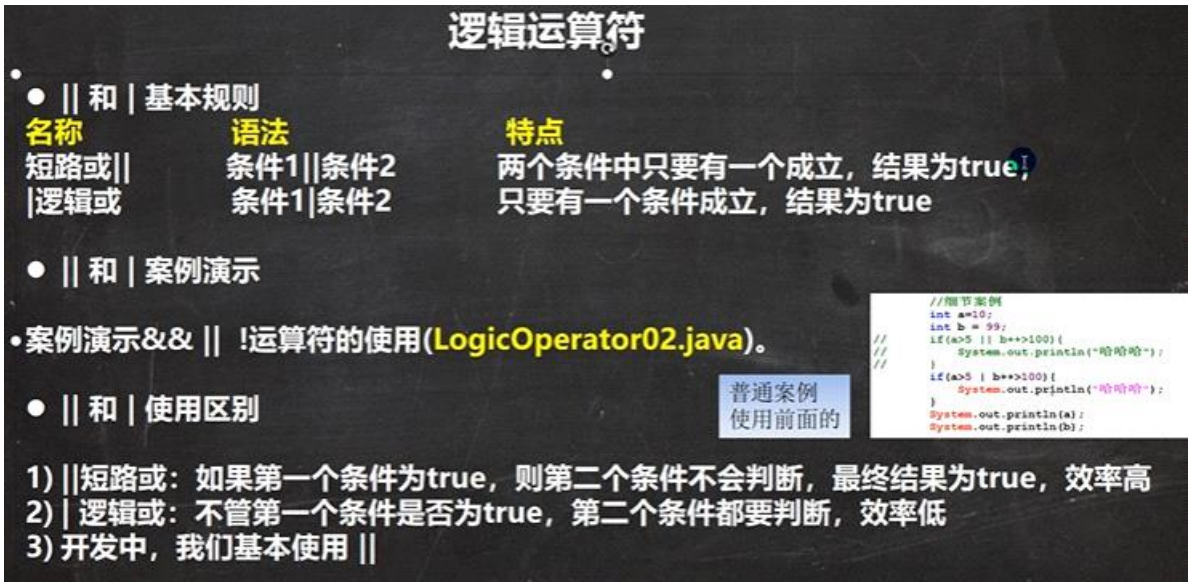
2021年4月25日 20:20

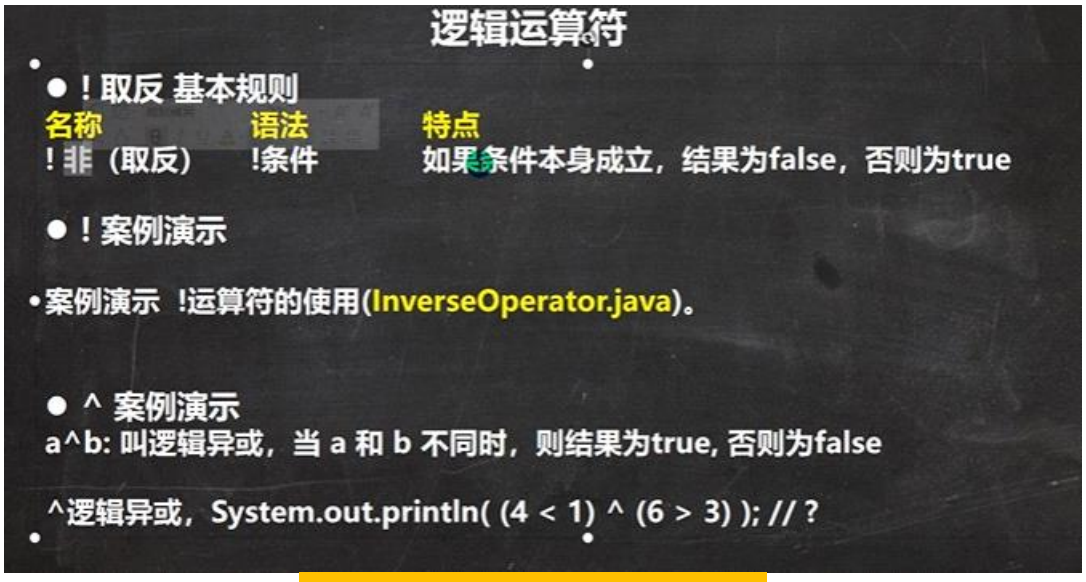



# P73-75短路或逻辑或；非；异或

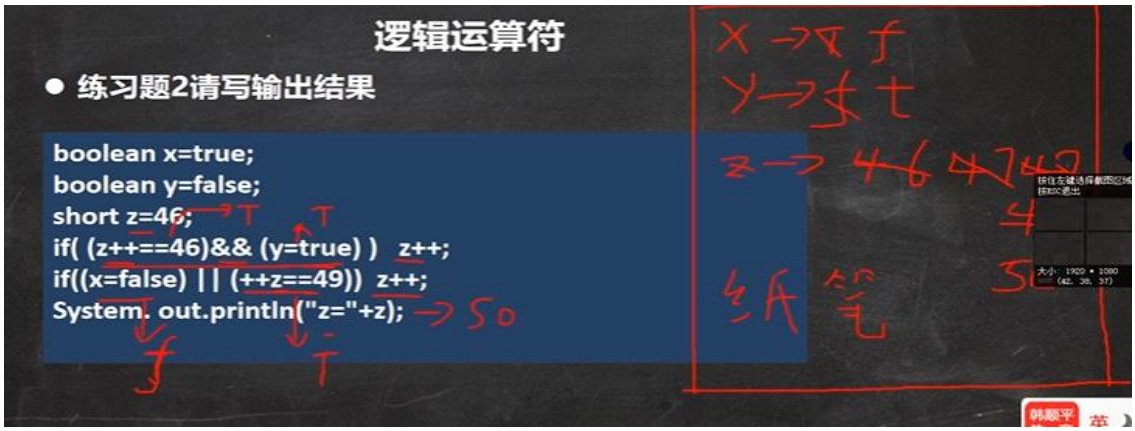
2021年4月26日　　23:25



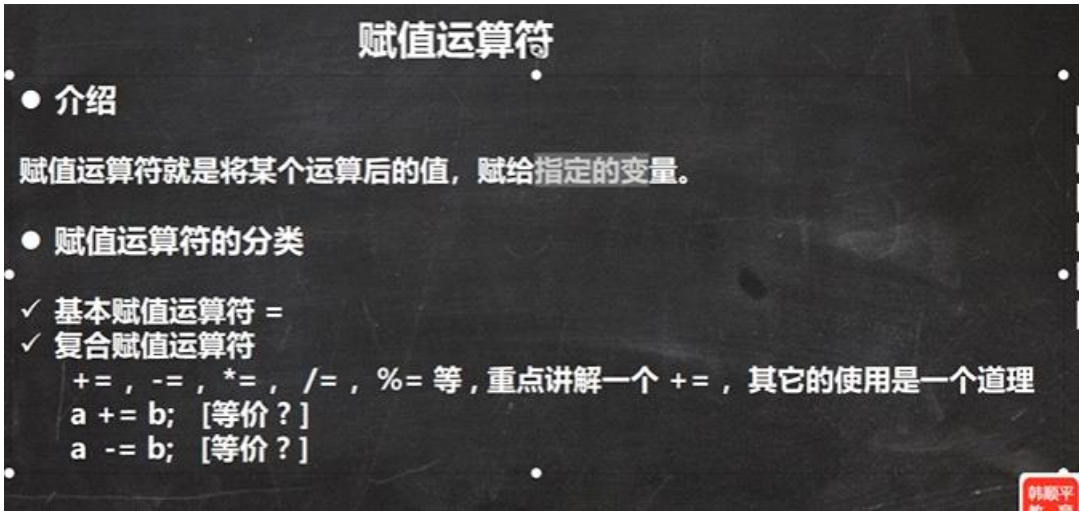

x++==6；后++先比较在自增、、

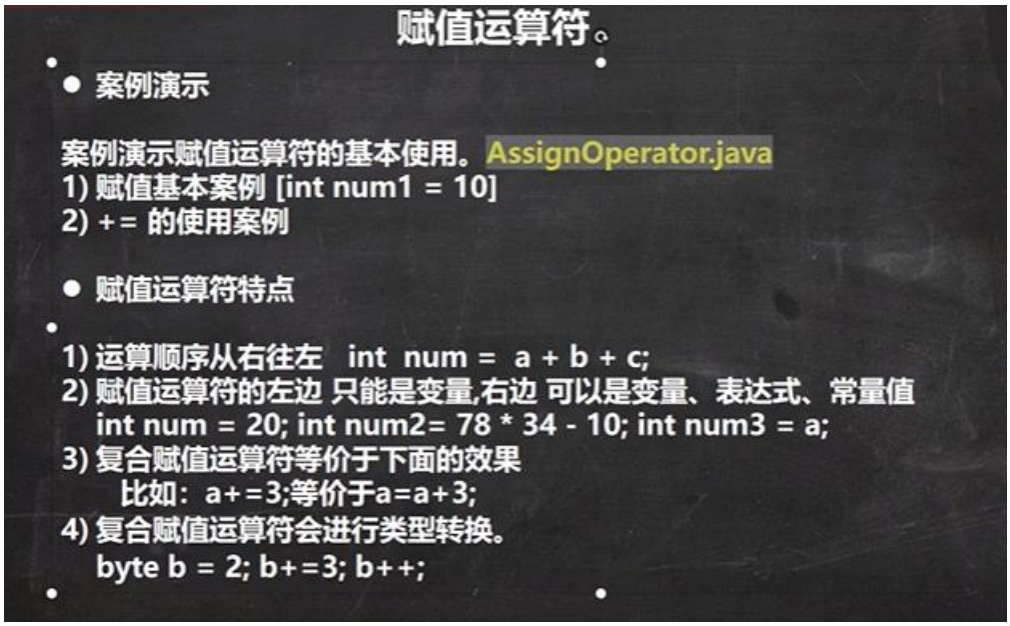
++x==6；前++先自增在比较、、



# P75-76赋值运算符

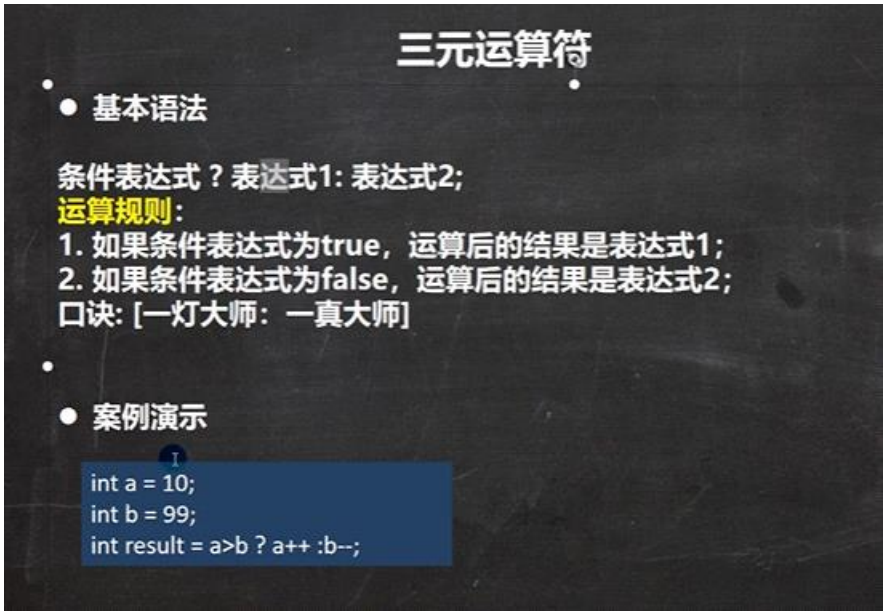
2021年4月27日 0:06

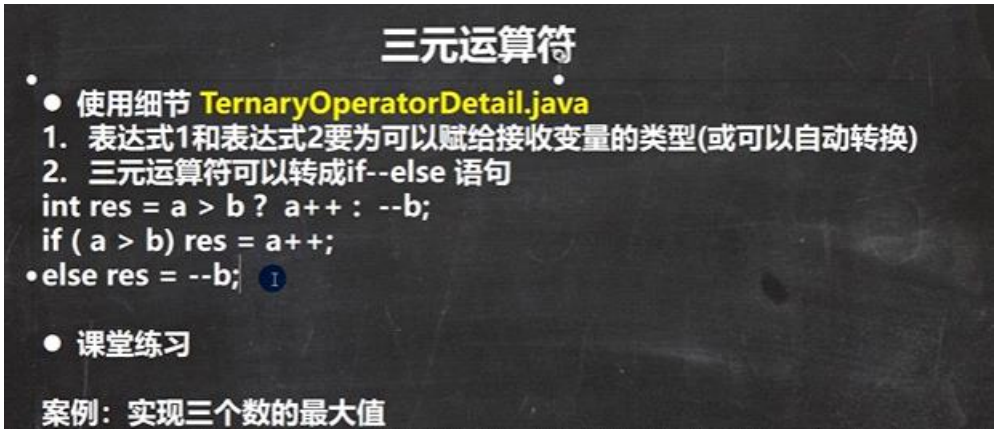


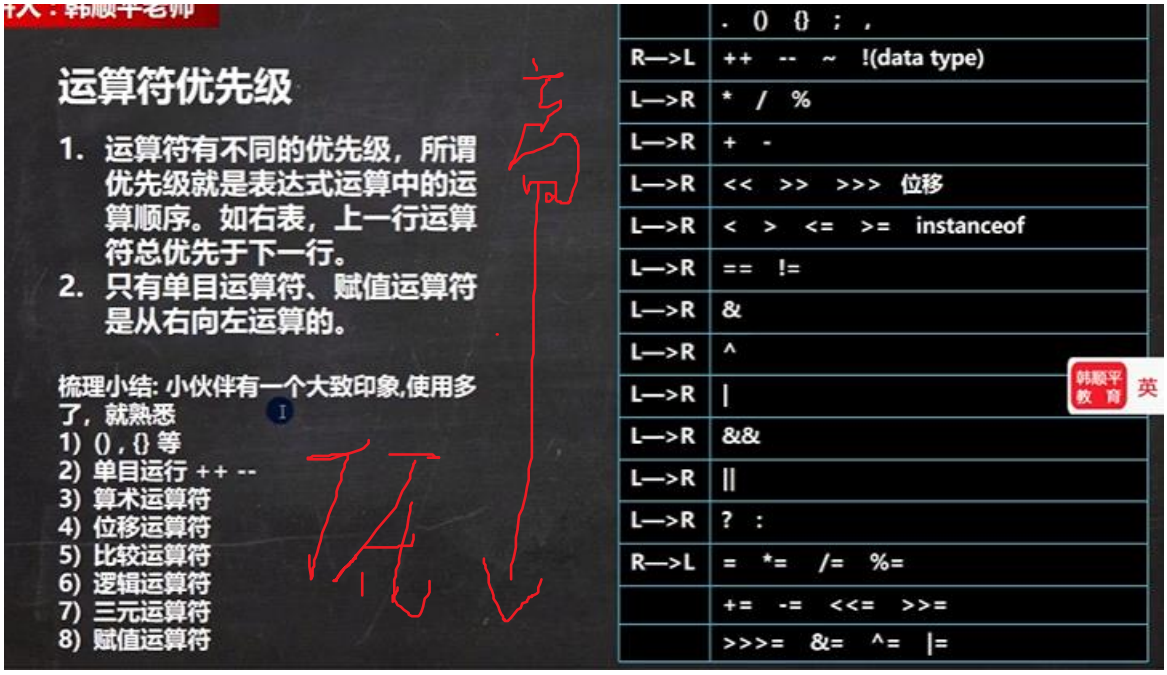



# P78-81三元运算符，优先级

2021年4月27日 0:16



# P82-84标识符

2021年4月27日 11:23