





#### 关于Java语言基础知识同学们需要学会什么?

#### 类型转换知识

#### 运算符的使用

怎么调用Java提供的程序

如何实现程序中不同 类型的变量或者数据 互相传输,不同类型 及一些逻辑运算、 据的类型如何确定?

程序中怎么进行数据 的基本算术运算、以 的数据运算的结果数 Java提供了具体哪些 运算方式

程序员需要调用Java 写好的一些程序,这 些程序如何去调用



## > 类型转换

- ◆ 自动类型转换
- ◆ 表达式的自动类型转换
- ◆ 强制类型转换
- > 运算符
- > 案例知识: 键盘录入技术



## 什么是自动类型转换,为什么要学习自动类型转换?

● **类型范围小**的变量,可以**直接赋值**给**类型范围大**的变量。

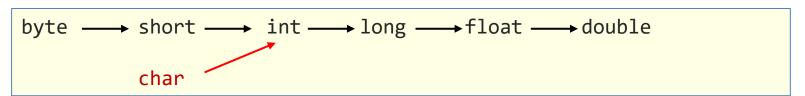
#### 自动类型转换的底层原理

```
byte a = 12 ;
int b = a;
System.out.println(b); // 12
```

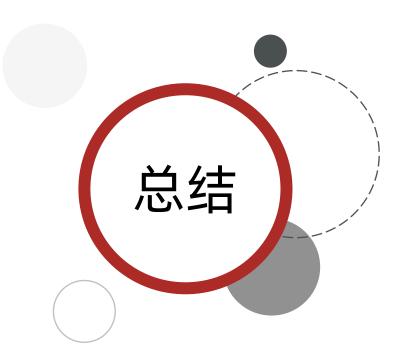
a **00001100** (8位)

b 00000000 00000000 0000000 00001100 (32位

#### 自动类型转换的其他形式







- 1. 为什么要进行类型转换?
  - 存在不同类型的变量赋值给其他类型的变量
- 2. 自动类型转换是什么样的?
  - 类型范围小的变量,可以直接赋值给类型范围大的变量。



## > 类型转换

- ◆ 自动类型转换
- ◆ 表达式的自动类型转换
- ◆ 强制类型转换
- > 运算符
- > 案例知识: 键盘录入技术



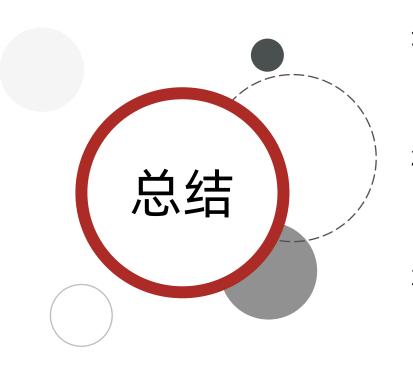
#### 表达式的自动类型转换

● 在表达式中,小范围类型的变量会自动转换成当前较大范围的类型再运算。

#### 注意事项:

- 表达式的最终结果类型由表达式中的最高类型决定。
- 在表达式中, byte、short、char 是直接转换成int类型参与运算的。





- 1. 表达式的自动类型转换是什么样的?
  - 小范围的类型会自动转换成大范围的类型运算。
- 2. 表达式的最终结果类型是由谁决定的?
  - 最终类型由表达式中的最高类型决定。
- 3. 表达式的有哪些类型转换是需要注意的?
  - byte short char是直接转换成int类型参与运算的。



## > 类型转换

- ◆ 自动类型转换
- ◆ 表达式的自动类型转换
- ◆ 强制类型转换
- > 运算符
- > 案例知识: 键盘录入技术



#### 场景

```
int a = 功能1();
功能2(byte b);
```

```
int a = 20;
byte b = a; //报错
```

#### 问题

● 类型范围大的数据或者变量,不能直接**赋值**给类型范围小的变量,会报错。

#### 强制类型转换

● 可以强行将类型范围大的变量、数据赋值给类型范围小的变量。

数据类型 变量2 = (数据类型)变量1、数据

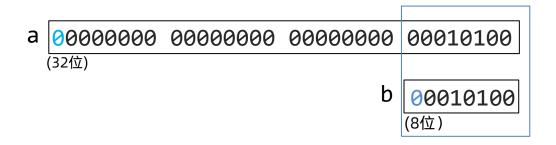
```
int a = 20;
byte b = (byte)a;
```



#### 强制类型转换底层原理

```
int a = 20;
byte b = (byte)a;
System.out.println(b); // 20
```

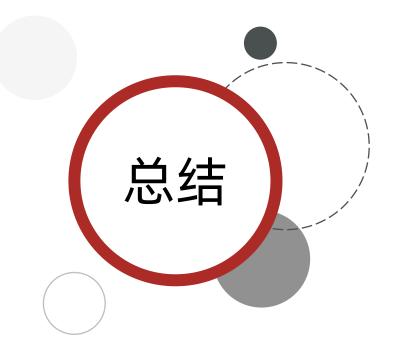
```
int i = 1500;
byte j = (byte)i;
System.out.println(j); // -36
```



#### 注意事项

- 强制类型转换**可能**造成数据(丢失)溢出;
- 浮点型强转成整型,直接丢掉小数部分,保留整数部分返回。





- 1. 什么是强制类型转换?
  - 可以强行将类型范围大的变量、数据赋值给类型范围小的变量。
  - 数据类型 变量 = (数据类型)变量、数据

- 2. 强制类型转换有哪些需要注意的?
  - 可能出现数据丢失。
  - 小数强制转换成整数是直接截断小数保留整数。



- > 类型转换
- > 运算符
  - ◆ 基本算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



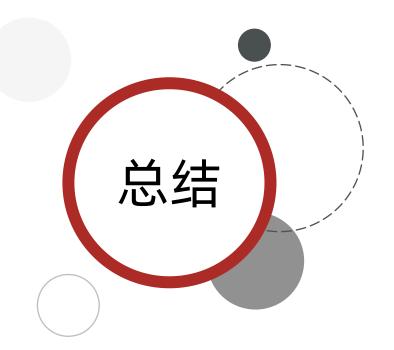
## 运算符

● 运算符:对字面量或者变量进行操作的符号。

## 算数运算符

符号	作用	说明 ····································
+	加	参考小学一年级
-	减	参考小学一年级
*	乘	参考小学二年级,与"×"相同
/	除	与 "÷"相同,注意:在Java中两个整数相除结果还是整数。
%	取余	获取的是两个数据做除法的 <mark>余数</mark>





1. 算数运算符有哪些?

• + \ -\ \*\ /\ %

- 2. / 需要注意什么, 为什么?
  - 如果两个整数做除法,其结果一定是整数,因为最高类型是整数。



# **富**案例

## 数值拆分

需求:一个三位数,将其拆分为个位、十位、百位后,打印在控制台

#### 请输入整数:

#### 123

整数123的个位为:3 整数123的十位为:2 整数123的百位为:1

#### 公式总结:

个位: 数值%10

十位: 数值 / 10 % 10

百位: 数值 / 10 / 10 % 10

千位: 数值 / 10 / 10 / 10 % 10;

• • •



- > 类型转换
- > 运算符
  - ◆ 基本算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



#### "+" 做连接符

● "+"符号与字符串运算的时候是用作连接符的,其结果依然是一个字符串。

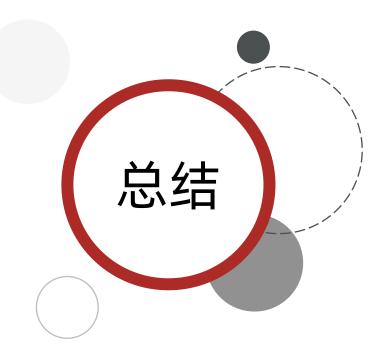
```
int a = 5;
System.out.println("abc" + 'a'); // abca
System.out.println("abc" + a); //abc5
System.out.println(5 + a); // 10
System. out.println("abc" + 5 + 'a');
System.out.println(15 + "abc" + 15);
System.out.println(a + 'a');
System.out.println(a + "" +'a');
System.out.println(a + 'a'+" itheima ");
System.out.println("itheima"+ a + 'a');
System.out.println("itheima"+ (a + 'a'));
```

【支付宝】支付宝 251\*@qq.\* 花 呗 05月 月账单 13839.17元,还 款日 05月20日,你还有1个花呗金可用于抵扣还款

#### 独门秘籍:

● 能算则算,不能算就在一起。(计算机很聪明)





- 1. + 除了做基本数学运算, 还有哪些功能?
  - 与字符串做+运算时会被当成连接符,其结果还是字符串。
  - 能算则算,不能算就在一起。



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术

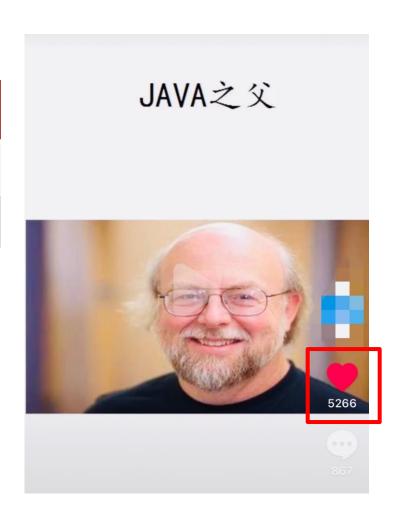


#### 自增自减运算符

符号	作用	说明
++	自增	变量自身的值加1
	自减	变量自身的值减1

#### 注意:

- ++ 和 -- 既可以放在变量的后边,也可以放在变量的前边。
- ++ 、-- 只能操作变量,不能操作字面量的。





#### 自增自减的使用注意事项

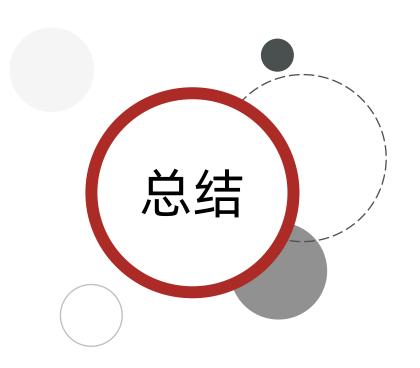
- ++、--如果不是单独使用(如在表达式中、或者同时有其它操作),放在变量前后会存在明显区别
  - ▶ 放在变量的前面, 先对变量进行+1、-1, 再拿变量的值进行运算。

```
int a = 10;
int rs = ++a;
```

▶ 放在变量的后面, 先拿变量的值进行运算, 再对变量的值进行+1、-1

```
int b = 10;
int rs = b++;
```





- 1. 自增、自减运算符是什么,有什么作用?
  - ++、--。对当前变量值+1、-1
- 2. 自增、自减运算符在变量前后有区别吗?
  - 如果单独使用放前后是没有区别的。
  - 非单独使用:在变量前 ,先进行变量自增/自减,再使用变量。
  - 非单独使用:在变量后,先使用变量,再进行变量自增/自减。





## 自增、自减拓展案例 (有些面试题会出现)

```
int c = 10;
int d = 5;
int rs3 = c++ + ++c - --d - ++d + 1 + c--;
System.out.println(rs3);
System.out.println(c);
System.out.println(d);
```



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



#### 基本赋值运算符

● 就是 "="。

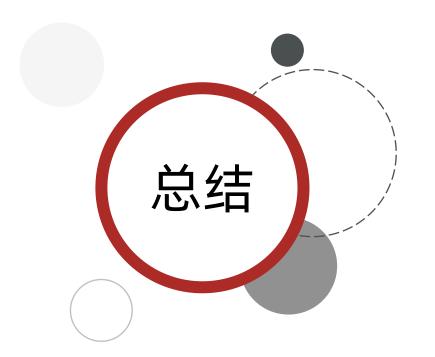
int a = 10; // 先看"="右边,把数据10赋值给左边的变量a存储。

#### 扩展赋值运算符

符号	作用	说明
+=	加后赋值	a+=b 等价于 a = (a的数据类型)(a+b); 将a + b的值给a
-=	减后赋值	a-=b 等价于 a = (a的数据类型)(a-b); 将a - b的值给a
*=	乘后赋值	a*=b 等价于 a = (a的数据类型)(a*b); 将a * b的值给a
/=	除后赋值	a/=b 等价于 a = (a的数据类型)(a/b); 将a / b的商给a
%=	取余后赋值	a%=b 等价于 a = (a的数据类型)(a%b); 将a % b的商给a

注意:扩展的赋值运算符隐含了强制类型转换。





- 1. 赋值运算符有哪些?
  - =, +=, -=, \*=, /=, %=
- 2. 作用是什么?
  - 基本作用是赋值。
  - +=还可以实现数据的累加,把别人的数据加给自己。



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



#### 关系运算符

● 是对数据进行条件判断的符号,最终会返回一个比较的布尔结果(false,true)。

符号	说明
==	a==b,判断a和b的值是否 <mark>相等</mark> ,成立为true,不成立为false
!=	a!=b,判断a和b的值是否 <mark>不相等</mark> ,成立为true,不成立为false
>	a>b, 判断a是否 <mark>大于</mark> b,成立为true,不成立为false
>=	a>=b,判断a是否 <mark>大于等于</mark> b,成立为true,不成立为false
<	a <b, 判断a是否<mark="">小于b,成立为true,不成立为false</b,>
<=	a<=b,判断a是否 <mark>小于等于</mark> b,成立为true,不成立为false

注意: 在做相等判断是, 千万不要把 == 误写成 =



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



## 逻辑运算符概述

● 可以把多个条件的布尔结果放在一起运算,最终返回一个布尔结果。

全部结果 > 屏幕尺寸: 6.95英寸及以上 × 运行内存: 8GB ×

"手机

符号	介绍	说明
&	逻辑与	必须都是true,结果才是true;只要有一个是false,结果一定是false。
1	逻辑或	只要有一个为true、结果就是true
!	逻辑非	你真我假、你假我真。 !true=false 、 !false= true
۸	逻辑异或	如果两个条件都是false或者都是true则结果是false。两个条件不同结果是true。



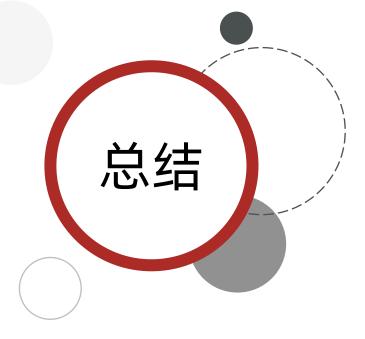
#### 短路逻辑运算符

符号	介绍	说明
&&	短路与	判断结果与"&"一样。过程是 <b>左边为</b> false, 右边则不执行。
П	短路或	判断结果与" "一样。过程是 <b>左边为 true, 右边则不执行。</b>

注意:逻辑与 "&" 、逻辑或 "|": 无论左边是 false还是 true,右边都要执行。



#### 1. 逻辑运算符有哪些,有什么特点?



- 逻辑与 "&": 有一个为false、结果是false
- 短路与&&: 一个为false、结果是false。前一个为false,后一个条件不执行了
- 逻辑或 "I":有一个为true、结果是true
- 短路或||: 一个为true、结果是true。前一个为true,后一个条件不执行了
- 逻辑非 "!": !false=true、!true=false
- 逻辑异或 "^":相同是false、不同是true。
- 注意:实际开发中、常用的逻辑运算符还是: && 、 || 、!



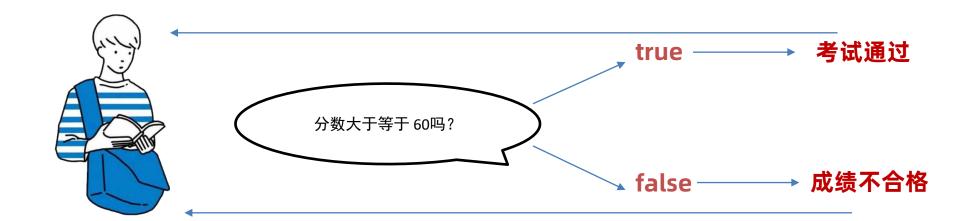
- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



#### 三元运算符介绍

● 格式: 条件表达式? 值1:值2;

● 执行流程:首先计算关系表达式的值,如果值为true,返回值1,如果为false,返回值2。





# 1 案例

## 求三个整数的最大值

需求: 定义三个整数, 找出最大值并打印在控制台。

#### 分析:

- ① 用三元运算符获取前两个整数的最大值,并用临时变量保存起来。
  - num1 > num2 ? num1 : num2;
- ② 用三元运算符, 让临时最大值, 和第三个整数, 进行比较, 并记录结果。
  - temp > num3 ? temp : num3;
- ③ 输出结果



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识: 键盘录入技术



#### 运算符优先级

● 在表达式中,哪个运算符先执行后执行是要看优先级的,例如"\*、/"的优先级高于"+、-"。

优先级	运算符
1	0
2	!, -, ++,
3	*、/、%
4	+、-
5	<<, >>, >>>
6	< 、 <= 、 >、 >=、 instanceof
7	== 、!=
8	&
9	۸
10	
11	&&
12	II
13	?:
14	=、 += 、 -= 、 *= 、 /=、 %=、 &=、





## 运算符优先级

// 目标: 看看Java是否存在优先级, 大家以后需要注意优先级问题

System.out.println(10 > 3 || 10 > 3 && 10 < 3); // true

System.out.println( (10 > 3 || 10 > 3 ) && 10 < 3); // false

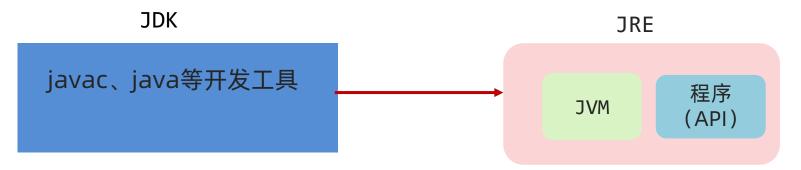


- > 类型转换
- > 运算符
- > 案例知识: 键盘录入技术



#### 需求:

● 请完成Java程序与用户交互,比如录入用户输入的名称、年龄,怎么办?。



#### API(Application Programming Interface,应用程序编程接口)

- Java写好的程序(功能代码), 咱们可以直接调用。
- Oracle 也为Java写好的程序提供了相应的 API文档(技术使用说明书)。

#### 下载API文档:

http://www.oracle.com/technetwork/java/javase/downloads/index.html



#### 键盘录入功能实现的三个步骤:

①:导包:告诉程序去JDK的哪个包中找扫描器技术。

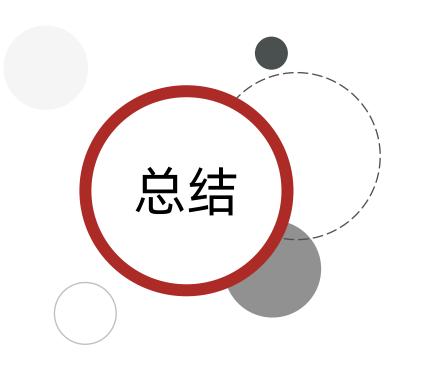
②: 写一行代码代表得到键盘扫描器对象。

③:等待接收用户输入数据。

#### 注意:

- System、String在JDK中的Java.lang包下
- lang包不需要我们导包,是默认的包。

```
package com.itheima.scanner;
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       System.out.println("请输入您的年龄:");
       int age = sc.nextInt();
       System.out.println("年龄是: " + age);
       System.out.println("请输入您的名称:");
        String name = sc.next();
       System.out.println("欢迎: " + name);
```



#### 1. API是什么?

- Application Programming Interface,应用程序编程接口。
- Java写好的程序,咱们可以直接调用。
- 2. 键盘录入的开发步骤
  - 导包: import java.util.Scanner;
  - 抄写代码得到扫描器对象: Scanner sc = new Scanner(System.in)
  - 抄写代码等待接收用户输入的数据: sc.nextInt() 、sc.next()



传智教育旗下高端IT教育品牌