java---表示数字的基本类型

2018年06月15日 20:49:31 qq_36098284 阅读数 1086 更多

数字的基本类型有整数型（4）、浮点型（2）、单字符型（1）、布尔型（2）、其中还包括了强制转换和隐式转换。

1）四种整数类型(byte、short、int、long)：

byte：8 位，用于表示最小数据单位，如文件中数据，-128~127

short：16 位，很少用，-32768 ~ 32767

int：32 位、最常用，-2^31-1~2^31 （21 亿）

long：64 位、次常用

注意事项：

int i=5; // 5 叫直接量（或字面量），即 直接写出的常数。

对于byte/short/char三种类型来说，如果右侧赋值的数值没有超过范围，那么javac编译器将会自动隐含地为我们补上一个(byte)(short)(char)。如果右侧超过了左侧范围，那么直接编译器报错。

整数字面量默认都为 int 类型，所以在定义的 long 型数据后面加 L或 l。

小于 32 位数的变量，都按 int 结果计算。

byte和short可以隐式转换成int，但是long不可以。

```
short result = a + b; // 错误写法！左侧需要是int类型
a,b是变量，变量的值是可能变化的，在编译的时候，编译器javac不确定b2+b3的结果是什么，因此会将结果以int类型进行处理，所以int类型不能赋值给byte类型，因此编译失败。
```

强转符比数学运算符优先级高。

2）两种浮点数类型(float、double)：

float：32 位，后缀 F 或 f，1 位符号位，8 位指数，23 位有效尾数。

double：64 位，最常用，后缀 D 或 d，1 位符号位，11 位指数，52 位有效尾

注意事项：

1. 二 进 制 浮 点 数 ：

1010100010=101010001.0*2=10101000.10*2^10（2次方)=1010100.010*2^11(3次方)= . 1010100010*2^1010(10次方)

尾数：1010100010

指数：1010

基数：2

2.浮点数字面量默认都为 double 类型，所以在定义的 float 型数据后面加F 或 f；double 类型可不写后缀，但在小数计算中一定要写 D 或 X.X

3.float 的精度没有 long 高，有效位数（尾数短）。 float 的范围大于 long指数可以很大。

4. 浮点数是不精确的，不能对浮点数进行精确比较。

3）一种字符类型(char)：

char：16 位，是整数类型，用单引号括起来的 1 个字符（可以是一个中文字符），使用 Unicode 码代表字符，0~2^16-1（65535）。

注意事项：

1. 不能为 0个字符。

2. 转义字符：\n 换行 \r 回车 \t Tab 字符 \" 双引号 \\ 表示一个\

3. 两字符 char 中间用“+”连接，内部先把字符转成 int 类型，再进行加法运算，char 本质就是个数！二进制的，显示的时候，经过“处理”显示为字符。

4）一种布尔类型(boolean)：

true 真

false 假

5）类型转换：

隐式转换： char--> 自动转换：byte-->short-->int-->long-->float-->double

强制转换：①会损失精度，产生误差，小数点以后的数字全部舍弃。②容易超过取值范围。

有一个疑问，为什么作为64位的long可以隐式转换成32位的float？（因为一般你会发现，隐式转换都是位数小转位数大的）

解答：

最为一个常识，我们都知道浮点型在内存中占用的是4个字节的空间，而long型占用的是8个字节的空间。可是为什么4个字节的float型的最大值会大于long型的最大值呢？

我们都知道，float类型的范围是：—3.403E38~3.403E38。而long类型的范围是：-2^63~2^63-1（大概是9*10^18）。

我以前也是简单的记住就算完事了，对于它为什么会这样却没有考虑过。

下面给大家分享一下我现在的理解：

long整型数，在内存中占用8个字节共64位，它表示的数值有2的64次方，平分正负，数值范围是负2的63次方到正2的63次方-1。

而float在内存中占4个字节，共32位，

但是浮点数在内存中是这样的：

V=(-1)^s * M * 2^E

667x190

浮点数的32位不是简单的直接表示大小，而是按照一定的标准分配的。

其中第1位，符号位，即S。

接下来的8位，指数域，即E。

剩下的23位，小数域，即M，M的取值范围为[1，2）或[0，1）。

也就是说，浮点数在内存中的二进制值不是直接转换为十进制数值的，而是按照上述公式计算而来，通过这个公式，虽然只用到了4个字节，但是浮点数却比长整型的最大值要大。

这也就是为什么在数据转换的时候，long类型转换为float类型的根本原因所在！

6）记忆：

8位：Byte（字节型）

16位：short（短整型）、char（字符型）

32位：int（整型）、float（单精度型/浮点型）

64位：long（长整型）、double（双精度型）

最后一个：boolean(布尔类型)

参考：https://www.cnblogs.com/123hll/p/5805040.html

参考：https://blog.csdn.net/hexu8080/article/details/53924178