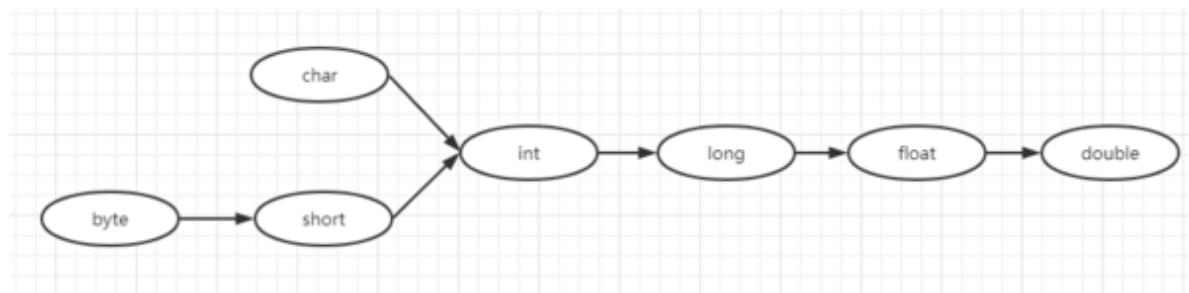


基本数据类型之间的转换

自动类型转换



长度短的类型按上图的顺序可以自动转换为长度大的类型

整数转浮点数时，会在添加一位0的小数位

char不能自动转byte和short

强制类型转换

当长度大的类型转为长度小的类型时，往往要强制类型转换

浮点数强制转整数时，会丢弃小数部分

长整数转短整数时，因为高位溢出，新数的最高位可能正好是1，所以有可能变负数

数字的字符串可以通过Integer.parseInt()方法转换为int

```
String str = "123";
int parseInt = Integer.parseInt(str);
System.out.println("parseInt(): " + parseInt);
```

每一种基本类型都对应一种包装类型/封装类型：

基本类型	包装类型
boolean	Boolean
char	Character
int	Integer
byte	Byte
short	Short
long	Long
float	Float
double	Bouble

基本类型直接赋值给包装类型，称为自动**装箱**；包装类型之间赋值给基本类型，称为自动**拆箱**

表达式自动类型提升

- 表达式中出现的整数数字默认按int类型处理
- byte, short, char类型出现在表达式中自动提升为int
- 一个表达式中的数据类型会提升为最大那个数据的类型

```
//表达式自动类型提升
short sh1 = 5;
//sh1 = sh1 + 1; //表达式中出现的整数默认按int处理
int int8 = 20;
long long8 = 200;
long long9 = sh1+int8+long8;
int int9 = (int) (sh1+int8+long8); //表达式中最大的类型是long, 所以计算结果也是
long类型
```

直接量（字面量）

代码中出现的基本类型的数值或字符串，称为直接量

true和false这两个直接量只能赋给布尔类型变量

null可以赋给所有的包装类型，不能赋值给基本类型

运算符

运算符是一些特殊的符号，用于数值计算（加减乘除等），赋值和比较。运算符可把多个操作数连接为一个表达式。

运算符分为以下几种类型：

- 算术运算符
- 赋值运算符
- 比较运算符
- 逻辑运算符
- 位运算符
- 其它运算符

算术运算符（***）

java中的算术运算符包括：+，-，*，/，%，++，--

+在Java里面除了加法作用以为，还可以用于连接字符串

/整数相除，结果是整数，如果结果有小数，舍弃小数部分，除数不能为0，否则运行时报异常；有一个操作数是浮点数，结果就是浮点数，可以被0除，不会报异常，但结果是无穷大（Infinity）

%取模（取余数），整数相除除数也不能为0；浮点数取模，结果也是浮点数，如果除数是0，结果为NaN（Not a Number）

++ 自加1，只有一个操作数，称为单目运算。++放在前面，先自加1，再参与表达式计算；++放在后面，先参与表达式计算，再自加1

-- 自减1，逻辑同++

赋值运算符

=是赋值运算，所用就是把一个直接量，变量，表达式的值赋给另一个变量。

赋值运算符有返回值，所以可用于连续赋值

```
int a = 8;  
int b;  
int c = b = a;
```

扩展的赋值运算符

赋值运算符可以与算术运算符中的+, -, *, /, %组合成新的赋值运算符。

sum += num ==> sum = sum + num

+=

-=

*=

/=

%=

比较运算符

比较两个变量或直接量的大小，并且返回boolean值

- > 大于
- < 小于
- == 等于
- != 不等于
- >= 大于等于
- <= 小于等于

逻辑运算符 (***)

- && 短路与
- || 短路或
- & 非短路与
- | 非短路或
- ! 非
- ^ 异或

位运算符

Java支持7种位运算符：

- & 按位与
- | 按位或
- ~ 按位非
- ^ 按位异或
- << 左位移
- >> 右位移
- >>> 无符号右移

三目运算符 (***)

(条件表达式) ? num1 : num2

根据表达式的结果 (boolean) , 如果为真, 返回num1(冒号前面的值); 如果为假, 返回num2(冒号后面的值)

运算符优先级 (**)

() > 单目 > 算术 > 位移 > 逻辑 > 赋值

运算符说明	Java 运算符
分隔符	. [] () {} , ;
单目运算符	++ -- ~ !
强制类型转换运算符	(type)
乘法/除法/求余	* / %
加法/减法	+ -
移位运算符	<< >> >>>
关系运算符	< <= >= > instanceof
等价运算符	== !=
按位与	&
按位异或	^
按位或	
条件与	&&
条件或	
三目运算符	?:
赋值	= += -= *= /= &= = ^= %= <<= >>= >>>=

流程控制语句

分支结构

- if (条件) {

}

如果添加为true, 大括号里面的语句块就会被执行

- if(条件){
 //条件为真执行代码块
}else{
 //条件为假执行的代码块
}

- if(条件){

}else if(条件){

```
}else{
```

```
}
```

今日作业

1. 如下代码，请问代码是否能正常编译执行？如果不行是因为什么原因？

```
byte b1 = 2;  
byte b2 = 3;  
byte b3 = b1 + b2;
```

2. 输入时间（单位秒），计算自由落体下落距离。 $s = 1/2 * g * t * t$, $g = 9.8$ 米/秒
3. 输入两个整数，找出最大数并打印。（用三目运算）
4. 输入一个整数，判断是否为正数？负数？还是0？
5. 输入年份，判断是否是闰年。（能被4整除，但不能被100整除；能被400整除）
6. 输入年份和月份，判断当月有多少天。（7月以前单月是31天，双月是30天；8月以后双月31天，单月是30天；闰年2月29天，平年2月28天）
7. 输入一个学生的成绩（100分制），给出该生等级。（90及以上A；80及以上B；70及以上C；60及以上及格；60以下不及格）