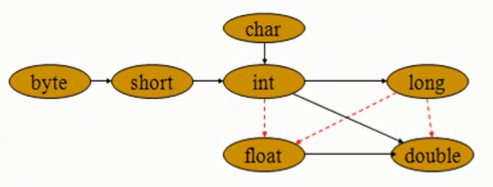
**1.自动类型转换**

**容量小的数据类型可以自动转换为容量大的数据类型；**  
容量指的是表数范围而不是字节数；



图中，**黑色实线箭头表示无数据丢失的自动类型转换，而红色虚线在转换时可能存在数据丢失。**

自动转型**byte<short<char<int<long<float<double**

byte、short、char在运算时会转换成int，结果也是int。

特例：

可以将整型常量直接赋值给**byte、short、char**等类型变量，而不需要进行强制类型转换，只要不超过其表数范围。

**2.强制类型转换**

用于显式的转换一个数值的类型

当将一种类型强制转换为另一种类型而又超出了目标类型的表数范围，就会被截断成为一个完全不同的值，无意义的值！

强制类型转换：(转换类型)转换的变量 例：(char)(a+2)

**short** s1 = 1;  
**int** s2 = s1 + 2;//自动转换  
**short** s3=(**short**) s2;//强转

**3.表达式中类型提升问题**

在混合多种数据类型的表达式中，类型会向容量大的值的数据类型提升

**4.自动类型转换中的值溢出问题**

int money=100000000;

int years=20;

long total=money\*years;

//返回的是负数,因为money和years都是int类型,相乘也为int,但其值超过了int的表数范围,所以溢出。

//错误修改\*完已经溢出了

**long total=(long)(money\*years);**

**//正确修改**

**long total=(long)money\*years;**

//算一个人70年心跳多少次

long times=70\*60\*24\*356\*70;//溢出

//一般我们优先对第一个数进行类型转换(可能前面就已经溢出)

long times=70L\*60\*24\*365\*70;