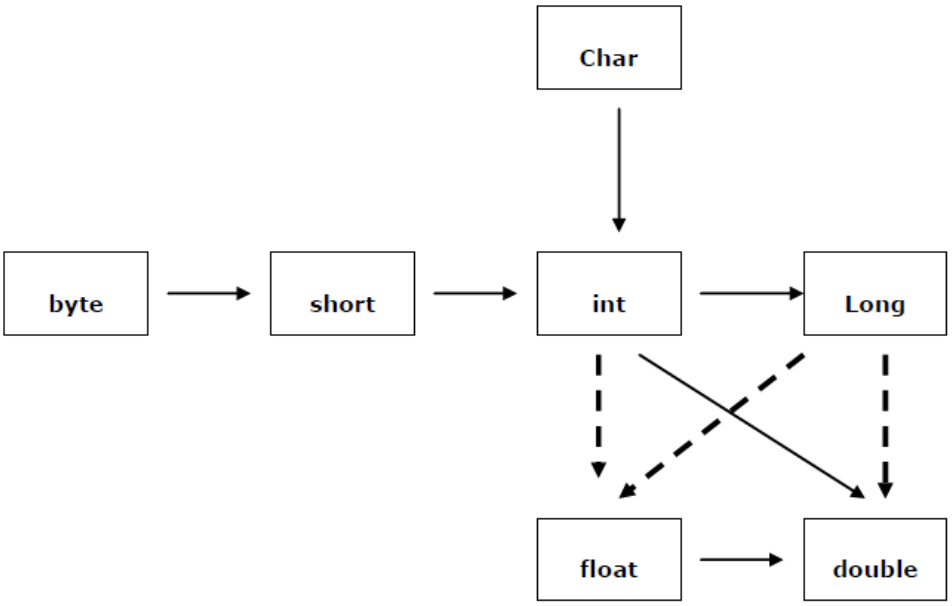
CLASE MATH Y CASTING

Conversiones entre tipos

A la hora de programar, muchas veces nos encontramos con la necesidad de realizar conversiones entre datos de tipo numérico. A continuación, se muestra un cuadro con las conversiones válidas:



Las flechas continuas reflejan conversiones sin pérdidas. Las tres flechas discontinuas reflejan una posible pérdida de precisión. Por ejemplo, un **long** tiene más dígitos de los que puede soportar un **double** con lo que en una conversión de **long** a **double** si el **long** es muy grande, perderá precisión.

Cuando se combinan dos valores mediante un operador, ambos operandos se transforman en un tipo común antes de realizar la operación siguiendo las siguientes pautas:

- **Si alguno de los operandos es de tipo double, el otro se traducirá a double.**
- **En caso contrario, si alguno de los operandos es de tipo float, el otro se traducirá a float.**
- **En caso contrario, si alguno de los operandos es de tipo long, el otro se traducirá en long.**
- **En caso contrario, ambos se traducirían a int.**

Casting

Los valores **int** se traducen a **double** automáticamente cuando es necesario. Pero a veces, hay ocasiones en las que es necesario considerar a un **double** como entero.

En Java es posible realizar estas conversiones numéricas, eso si, perdiendo información, en un proceso llamado casting.

La sintaxis de un casting consiste en dar el tipo de destino entre paréntesis poniendo después el nombre de la variable.

Ejemplo:

```
double x = 9.997;

int j = (int) x;
```

La variable j tiene valor 9 porque al refundir un valor de coma flotante a entero, se descarta la parte fraccionaria.

Si lo que queremos es redondear, tendremos que utilizar el método **round** de la clase Math.

Ejemplo:

```
double x= 9.997;

int j =(int) Math.round(x);
```

Ahora la variable j tiene un valor de 10. En este caso concreto sigue siendo necesario refundir cuando se hace la llamada a **round**. El valor resultante al aplicar el método **round** en este caso es un **long**. Si no queremos perder información estamos obligados a refundir.