

4.4.- El tipo de datos real (float, double)

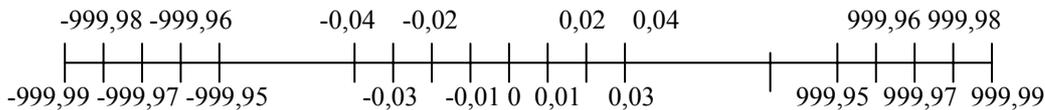
4.4.1.- Dominio de valores

Es el conjunto R.

En realidad, se trata de un subconjunto de R, ya que R es un conjunto infinito y no se pueden representar todos los números reales. Dependerá de cuántos bits se utilicen para su representación. Además ni siquiera se pueden representar todos los números reales que se encuentran entre dos números cualesquiera porque también hay infinitos. Esto no ocurría con los números enteros donde se podían representar todos los números comprendidos en un intervalo.

4.4.2.- Representación interna

¿Cómo representar los números reales? Si se decide representar todos los números reales con un número fijo de decimales. Por ejemplo, se podrían representar todos los números reales con 3 cifras en la parte entera y 2 decimales. A esto se le llama una **representación en coma fija** ya que se utilizan un número fijo de decimales.



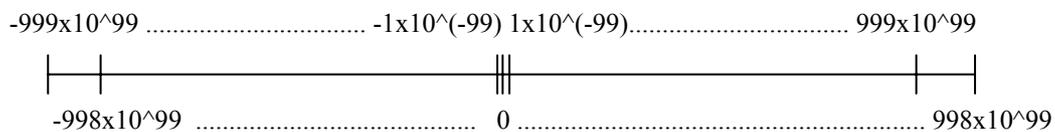
Esta representación no es buena porque no permite representar números muy cercanos al cero lo cual puede producir errores muy grandes:

Ejemplo: Trabajando con los números anteriores (que sólo permiten 2 decimales), al calcular el 1 por mil de 999 (que es 0,999) podría dar que es 0.

$$1/1000 * 999 = 0,001 * 999 \rightarrow 0 * 999 = 0$$

Por ello se utiliza una **representación en coma flotante**, donde se pueden representar números muy pequeños cercanos al cero y números muy grandes. Un número real se representa con dos números enteros: uno que es la mantisa (que contiene los dígitos significativos) y otro que es el exponente (que determina la posición de la coma)

Todo número real se puede representar así: Mantisa * 10 ^ Exponente



Se pueden representar números muy pequeños (cercanos al cero) y números muy grandes, pero la precisión es mucho mayor en números cercanos al cero que en números muy grandes.

Por lo tanto se puede representar cualquier número real con dos números enteros (que por supuesto se pueden representar en complemento a dos).

4.4.3.- Representación externa y definición en Java

En Java hay varios tipos de datos de tipo real que se diferencian en la precisión de cada uno de ellos. El número de bits empleado para cada uno de ellos depende del compilador

tipo `float` con precisión normal → habitualmente 32 bits repartidos entre mantisa y exponente

tipo `double` con doble precisión → habitualmente 64 bits repartidos entre mantisa y exponente

Constantes de tipo `double`:

- Utilizando el punto decimal.

Ejemplos: 23.435, -35.123

- Utilizando la notación científica.

Ejemplo: 234.43e32 que representa el número 234.43×10^{32}
3.43e-15 que representa el número 3.43×10^{-15}

Constantes de tipo `float`:

Son las mismas que las de tipo `double`, pero terminadas en `f`.

Por ejemplo: 23.435f, -35.123f, 234.43e32f, 3.43e-15f

```
float r; double p; /* Declaración de variables de tipo real (simple y doble
                    precisión) */
.....
r=23.435f;
r=234.43e32f; /* Esto es, en r se almacena la constante 234,43 x 10^32 */
p=-3.43e-15; /* Esto es, en r se almacena la constante -3,43 x 10^(-15) */
```

4.4.4.- Operaciones

Aritméticas

Se pueden utilizar los operadores +, -, *, /

NOTA: El operador de división / realiza la división real, siempre y cuando al menos uno de los operadores sea de tipo real (una variable o constante de tipo float o double).

Ejemplo: Si `a` y `b` fueran variables de tipo `int`, y `r` de tipo `float`, entonces la siguiente instrucción dejaría en `r` el cociente entero entre `a` y `b`.

```
r=a/b; /* Asigna a r el cociente entero de dividir a por b */
```

Si se quisiera dejar en `r` la división entera, entonces hay que hacer un `CASTING`.

```
r=(float)a/b; /* Asigna a r la división real entre a y b */
```

```
int a,b;
float r, p;
a=3;
b=2;
r=a/b; /* Dejaría en r el valor 1 ya que es el cociente */
System.out.println(r);
p=a/2; /* Dejaría en p el valor 1 ya que es el cociente */
System.out.println(p);
r=(float)a/b; /* Es necesario hacer un casting para decir que se
quiere que sea real */
System.out.println(r);
p=a/2.0f; /* Si ponemos la constante real entonces también funciona */
System.out.println(p);
```

Comparación

Hay otros operadores de comparación (<, <=, >, >=, == (igual), != (distinto)) cuyos resultados son valores booleanos o lógicos.

Ejemplo: $3.56 < 98.34E15 \rightarrow$ cierto

Funciones matemáticas

Funciones para trabajar con números reales (que se encuentran en la clase Math). Se presenta a continuación un subconjunto de ellas:

Math.abs(x)	devuelve el valor absoluto de un número
Math.sin(x)	devuelve el seno de x siendo x el ángulo en radianes
Math.cos(x)	devuelve el coseno de x siendo x el ángulo en radianes
Math.tan(x)	devuelve la tangente de x siendo x el ángulo en radianes
Math.asin(x)	devuelve en radianes el arco (entre -PI/2 y PI/2) cuyo seno es x (entre -1 y 1)
Math.acos(x)	devuelve en radianes el arco (entre 0 y PI) cuyo coseno es x (entre -1 y 1)
Math.atan(x)	devuelve en radianes el arco (entre -PI/2 y PI/2) cuya tangente es x
Math.exp(x)	devuelve e elevado a x (e^x)
Math.log(x)	devuelve el logaritmo neperiano de x (ln(x)) (error si x es menor o igual que 0)
Math.pow(x,y)	devuelve x elevado a y (error si x=0 e y <=0 o bien si x<0 e y no es entero)
Math.sqrt(x)	devuelve la raíz cuadrada de x
Math.ceil(x)	devuelve el menor entero no menor que x
Math.floor(x)	devuelve el mayor entero no mayor que x
Math.E	devuelve el número E
Math.PI	devuelve el número PI
Math.random()	devuelve un número aleatorio de tipo double

Entrada/Salida de números reales

Entrada: Con la instrucción readDouble se puede leer un número double y asignarlo a una variable de tipo double. Para ello se necesita definir un objeto de la clase IODialog.

Salida: Se consigue con las instrucciones print y println, que ya conocíamos. Se necesita definir un objeto de la clase IODialog, o también se puede realizar directamente usando System.out

Ejemplo: El siguiente programa pregunta un número al usuario y se lo muestra.

```
import acm.io.IODialog;

public class LeerString {
    public static void main(String[] args) {
        IODialog entradaSalida; // Declaración de variables
        double num ;

        entradaSalida = new IODialog(); // Instrucciones

        num = entradaSalida.readDouble("Dime un número: ");
        entradaSalida.println("Leído: "+num);}}}
```