

# TIPI DI DATI FONDAMENTALI

## TABELLA RIASSUNTIVA

Tipo	Descrizione	Dimension
int	Tipo intero con intervallo $-2147483648 \dots 2147483647$ (circa 2 miliardi)	4 byte
byte	Tipo che descrive un singolo byte, con intervallo $-128 \dots 127$	1 byte
short	Tipo intero “corto”, con intervallo $-32768 \dots 32767$	2 byte
long	Tipo intero “lungo”, con intervallo $-9223372036854775808 \dots 9223372036854775807$	8 byte
double	Tipo in virgola mobile a doppia precisione, con intervallo circa $\pm 10^{308}$ e circa 15 cifre decimali significative	8 byte
float	Tipo in virgola mobile a singola precisione, con intervallo circa $\pm 10^{38}$ e circa 7 cifre decimali significative	4 byte
char	Tipo che rappresenta caratteri codificati secondo lo schema Unicode (Argomenti avanzati 4.5)	2 byte
boolean	Tipo per i due valori logici <code>true</code> e <code>false</code> (Capitolo 6)	1 bit

## TIPI NUMERICI

### BYTE

- Numer intero 8 bit

```
byte x = 1;
```

### SHORT

- Numero intero 16 bit

```
short x = 10;
```

### INT

- Numero intero 32 bit;

```
int x = -100000;
```

## **LONG**

- Numero intero 64 bit

```
long x = 3000000000L;
```

## **FLOAT**

- Numero virgola mobile 32 bit

```
float x = 30.2f;
```

## **DOUBLE**

- Numero virgola mobile 64 bit

```
double x = 102.23972;
```

### **VALORI MINIMI E MASSIMI**

byte	Byte.MIN_VALUE	Byte.MAX_VALUE
short	Short.MIN_VALUE	Short.MAX_VALUE
int	Integer.MIN_VALUE	Integer.MAX_VALUE
long	Long.MIN_VALUE	Long.MAX_VALUE
float	Float.MIN_VALUE	Float.MAX_VALUE
double	Double.MIN_VALUE	Double.MAX_VALUE

### **Attenzione**

- Java non segnala overflow -> restituisce solo risultato errato
- 1/0 (interi) -> ArithmeticException
- 1/0 (decimali) -> infinito
- 0/0 (decimali) -> NaN

## ERRORI DI ARROTONDAMENTO

### MANCANZA DI PRECISIONE DEI DOUBLE

- Nel tipo double vi possono essere errori di mancanza di precisione in numeri molto grandi
  - numero limitato di cifre significative
  - può sbagliare anche di più unità
- È importante considerare la precisione di ogni tipo prima di fare calcoli

### NUMERI BINARI PERIODICI

- Numeri decimali esatti
  - non hanno tutti rappresentazione esatta in binario
- Esempio: 4.35 -> NO codifica esatta in binario

## ASSEGNAZIONI CON CONVERSIONE

- Ammesse senza conversioni
  - assegnazioni di tipi compatibili
  - int -> long
  - float -> double

### TYPE CASTING

- Assegnazioni con possibile perdita di precisione

```
double doubleVar = 2.99;  
int intVar = (int) doubleVar;  
// intVar -> 2
```

### ARROTONDAMENTO

```
double doubleVar = 2.99;  
int intVar = (int) Math.round(doubleVar);  
// intVar -> 3
```

### ERRORI DI APPROSSIMAZIONE IN OPERAZIONI ARITMETICHE

- Divisioni tra numeri interi danno risultato intero

```
System.out.println(7 / 4) // 1  
System.out.println(7.0 / 4.0) // 1.75
```

## PACKAGE JAVA.MATH.\*

- Classi:
  - BigInteger
  - BigDecimal
- Mi permette di avere un qualunque livello di precisione

## CLASSE MATH

- Non serve importarla
- Presenta metodi statici e costanti utili

```
Math.pow(x, y);  
Math.sqrt(x);  
Math.log(x);  
Math.sin(x);  
Math.E;  
Math.PI;
```

## ABBREVIAZIONI PER ASSEGNAZIONI E ARITMETICA

```
counter += 5;  
counter *= 2;  
counter++;  
counter--;
```

## BOOLEAN

- Può assumere solo due valori: `true` e `false`
- Sono chiamati anche "flag"