

Operazioni aritmetiche e la classe Math

Docente
Mario Perna
prof.perna.mario@darzo.net

A.S.
2025/2026
Materia
Informatica

Numeri interi

In Java per definire dei numeri interi possiamo utilizzare il tipo primitivo `int` o il tipo riferito `Integer`

```
public class Es5 {  
    public static void main(String[] args) {  
        int intero_primitivo = 5;  
        Integer intero_oggetto = 5;  
  
        System.out.println("Intero (primitivo): " + intero_primitivo);  
        System.out.println("Intero (oggetto): " + intero_oggetto);  
    }  
}
```

Numeri in virgola mobile (32bit)

In Java per definire dei numeri in virgola mobile possiamo utilizzare il tipo primitivo `float` o il tipo riferito all'oggetto `Float` per valori con una precisione fino a 6-7 cifre decimali.

```
public class Es5 {  
  
    public static void main(String[] args) {  
        float float_primitivo = 5.345F;  
        Float float_oggetto = 5.345F;  
  
        System.out.println("Float (primitivo): " + float_primitivo);  
        System.out.println("Float (oggetto): " + float_oggetto);  
    }  
}
```

N.B. Per assegnare un valore di tipo `float` è necessario porre il carattere `F` al termine del numero scelto.

Numeri in virgola mobile (64bit)

In Java per definire dei numeri in virgola mobile possiamo utilizzare il tipo primitivo `double` o il tipo riferito all'oggetto `Double` per valori con una precisione fino a 14-15 cifre decimali.

```
public class Es5 {  
    public static void main(String[] args) {  
        double double_primitivo = 5.345;  
        Double double_oggetto = 5.345;  
  
        System.out.println("Double (primitivo): " + double_primitivo);  
        System.out.println("Double (oggetto): " + double_oggetto);  
    }  
}
```

Operazioni aritmetiche di base

Java supporta l'utilizzo dei seguenti operatori aritmetici di base

Operatore	Risultato
+	Somma di due numeri
-	Sottrazione due numeri
*	Moltiplicazione di due numeri
/	Divisione di due numeri
%	Resto della divisione di due numeri

Operazioni aritmetiche di base

```
public class Es5 {  
  
    public static void main(String[] args) {  
        int a = 8;  
        int b = 5;  
        double c = 3.5;  
        double d = 1.5;  
  
        System.out.println("a+b: " + (a+b));  
        System.out.println("a-b: " + (a-b));  
        System.out.println("a*b: " + (a*b));  
        System.out.println("a/b: " + (a/b));  
        System.out.println("a%b: " + (a%b));  
        System.out.println("-----");  
        System.out.println("c+d: " + (c+d));  
        System.out.println("c-d: " + (c-d));  
        System.out.println("c*d: " + (c*d));  
        System.out.println("c/d: " + (c/d));  
        System.out.println("c%d: " + (c%d));  
    }  
}
```

```
a+b: 13  
a-b: 3  
a*b: 40  
a/b: 1  
a%b: 3  
-----  
c+d: 5.0  
c-d: 2.0  
c*d: 5.25  
c/d: 2.3333333333333335  
c%d: 0.5
```

Precedenza degli operatori

```
public static void main(String[] args) {  
    int a = 8;  
    int b = 5;  
    double c = 3.5;  
    double d = 1.5;  
  
    double risultato = a*b-c/d; // attenzione alle regole di precedenza  
    System.out.println("Risultato: " + risultato);  
  
    risultato = (a*b-c)/d; // posso modificare la precedenza con le parentesi  
    System.out.println("Risultato: " + risultato);  
}
```

Risultato: 37.666666666666664

Risultato: 24.333333333333332

N.B. State sempre attenti alle regole di precedenza degli operatori e al tipo di variabile utilizzata. Nel dubbio un paio di parentesi aggiuntive non fanno mai male e rendono il codice sorgente più esplicito.

Lettura in input

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int x;
    float y;
    double z;

    System.out.print("Inserire x (intero): ");
    x = scanner.nextInt();
    System.out.print("Inserire y (float): ");
    y = scanner.nextFloat(); // bisogna inserire un numero con separatore ,
    System.out.print("Inserire z (double): ");
    z = scanner.nextDouble(); // bisogna inserire un numero con separatore ,

    System.out.println("x vale " + x);
    System.out.println("y vale " + y);
    System.out.println("z vale " + z);
}
```

N.B. i metodi `nextFloat()` e `nextDouble()` accettano in input valori che utilizzano come separatore decimale la `,` e non il `.`

La classe Math

La classe Math fornisce diverse metodi che ci permettono di:

- Calcolare la radice quadrata di un numero (metodo sqrt)
- Calcolare la potenza di un numero (metodo pow)
- Calcolare il valore assoluto di un numero (metodo abs)
- Generare un numero casuale tra 0 e 1 (metodo random)

Inoltre fornisce anche diversi attributi per l'utilizzo di costanti matematiche:

- Il valore del pigreco (attributo PI)
- La costante e (attributo E)
- E tanti altri...

```
public static void main(String[] args) {  
    System.out.println("Il valore assoluto di -5 vale " + Math.abs(-5));  
    System.out.println("La radice quadrata di 16 vale " + Math.sqrt(16));  
    System.out.println("5 elevato alla 2 vale " + Math.pow(5, 2));  
  
    System.out.println("Il valore del pigreco: " + Math.PI);  
    System.out.println("Il valore di e: " + Math.E);  
}
```

Informazioni sull'utilizzo dei materiali didattici

Queste slides si basano su materiali originariamente elaborati dal **Prof. Andrea Melioli**, opportunamente modificati e integrati secondo specifiche esigenze riguardanti la programmazione disciplinare.

L'autore autorizza al **prof. Mario Perna** l'utilizzo, la modifica, la pubblicazione e qualsiasi altra forma di operazione sui materiali a scopo didattico e formativo.

L'uso o la diffusione di questi materiali è **vietato** senza il preventivo contatto con il proprietario del documento, Prof. Mario Perna (prof.mario.perna@darzo.net).