

---

## 1.6 Casting

Il **casting** è un meccanismo che consente di convertire i dati da un tipo primitivo a un altro.

Esempio:

```
short c = 10;  
int d = c;
```

Questa conversione può essere fatta senza problemi, perché siamo sicuri che il valore contenuto in `c` potrà essere sicuramente contenuto in `d`, essendo il range del tipo `int` più grande di quello del tipo `short`.

Ma cosa succede se facciamo l'operazione inversa? Ossia cerchiamo di convertire un `int` in un `short`?

Esempio:

```
int a = 10;  
short b = a; → errore di compilazione
```

Il compilatore ci segnala l'errore perché c'è il pericolo di perdita di precisione: infatti un numero intero potrebbe essere più grande di un numero `short`.

Possiamo ignorare questa segnalazione effettuando un cast esplicito, cioè inseriamo una coppia di parentesi, indicando all'interno il tipo dati in cui convertire la variabile "a".

Esempio:

```
int a = 10;  
short b = (short) a;
```

in questo modo il programmatore si sta assumendo il rischio di una eventuale perdita di precisione.

Ma che succede se il valore di “a” è più grande di “b”?

**Esempio:**

```
int a = 100000;  
short b = (short) a;
```

dentro b ci sarà il valore “-31072”, cioè, in questo caso, il valore inferiore del range di short.

In un casting, per esempio, tra double e float invece perdiamo le ultime cifre decimali.

**Esempio:**

```
double a = 0.100000234523;  
float b = (float)a; → dentro “b” ci sarà il valore  
approssimato “0.10000023”
```

---

## 1.7 Operatori

Sulle variabili possiamo eseguire tutta una serie di operazioni. Di seguito le più comuni.

### 1.7.1 Operatori Aritmetici

+	Somma	A+B
-	Sottrazione	A-B
*	Moltiplicazione	A*B
/	Divisione	A/B
%	Modulo	A%B

### Esempio ( addizione )

```
int a=5;
int b=2;
int c;
c=a+b; // C vale 7 ossia A+B=5+2 = 7.
```

N.B. i simboli // servono per indicare i commenti in java, ossia del testo che non viene preso in considerazione dal compilatore ma serve al programmatore per capire il funzionamento del codice.

Se dobbiamo incrementare solo di 1 la variabile possiamo scrivere brevemente:

```
x++; // equivale a x=x+1
```

### Esempio ( sottrazione )

```
int a=5;
int b=2;
int c;
c=a-b; // C vale 3 ossia A-B=5-2 = 3.
```

Se dobbiamo decrementare di uno la variabile possiamo scrivere brevemente:

```
x--; // equivale a x=x-1
```

**Esempio** ( moltiplicazione )

```
int a=5;
int b=2;
int c;
c=a*b; // C vale 10 ossia A*B=5*2 = 10.
```

**Esempio** ( divisione )

```
int a=5;
int b=2;
int c;
c=a/b; // ATTENZIONE C è approssimato a 2 perché C è
un intero
```

**Esempio** ( modulo )

```
int a=5;
int b=2;
int c;
c=a%b; // C = 1 perché il resto della divisione è 1
```

Attenzione: La **divisione tra interi** ritorna un valore intero, perciò se si vuole ottenere un numero reale dalla divisione di due numeri interi x e y, bisogna prima trasformarli in reali:

```
float risultato= (float) x / (float) y;
```

---

## 1.8 Stampa Su Console

In Java, per far stampare, al nostro programma, qualcosa a “video” si utilizza il comando **System.out.println**:

```
System.out.println();
```

Inserendo un testo all’interno delle parentesi tonde, questo testo viene stampato a video (o, per essere un pò più precisi, a **console**)

### Esempio

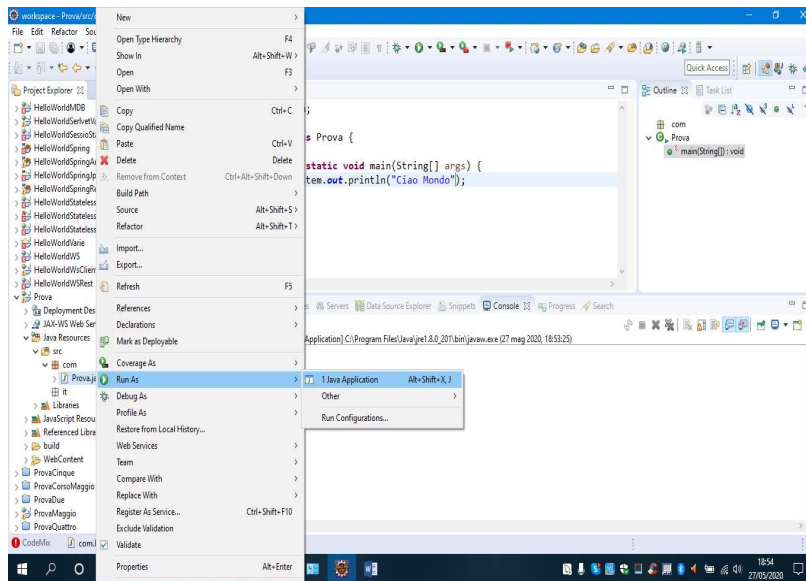
Scriviamo, nel nostro file, questa istruzione:

```
System.out.println("ciao mondo");
```

E eseguiamo il nostro programma.

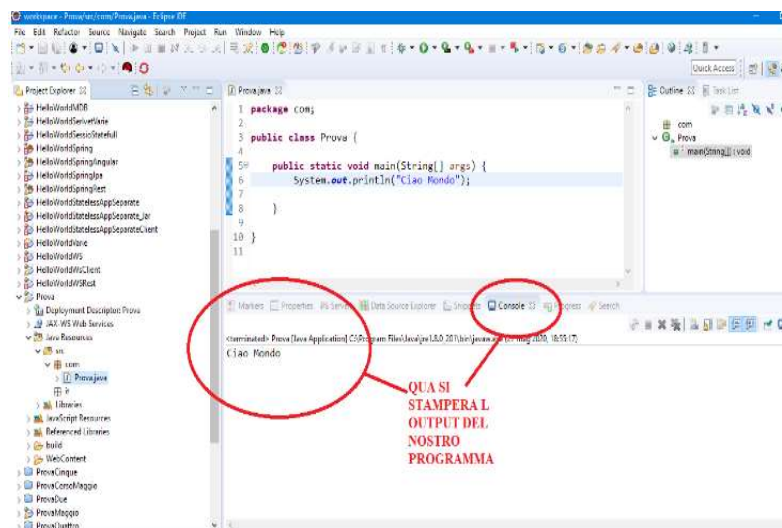
*Ma come si esegue ( o, come spesso si dice, si lancia) il nostro programma?*

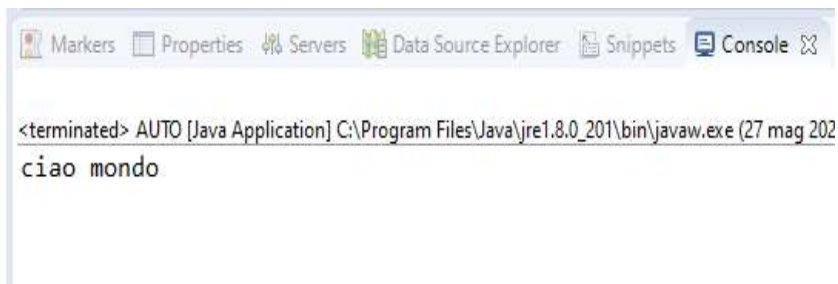
- Selezionare il file (nel nostro caso HelloWorld.java).
- Premere tasto destro.
- Selezionare dal menu runAs->Java Application



Tutto qui. Abbiamo eseguito il nostro primo programma.

L'output prodotto, in questo caso la scritta a video, sarà mostrato nel tab chiamato **Console**:





```
Markers Properties Servers Data Source Explorer Snippets Console  
<terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (27 mag 202  
ciao mondo
```

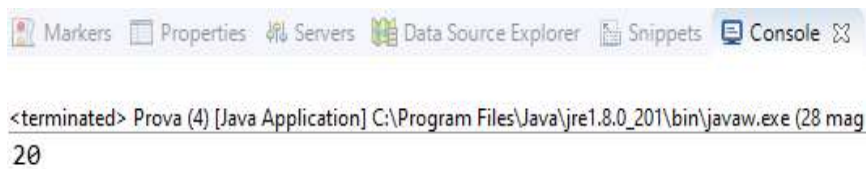
*E se volessimo stampare il valore di una variabile?*

E' sufficiente scrivere la variabile all'interno delle parentesi tonde.

### Esempio

```
int s=20;  
System.out.println(s);
```

### Output



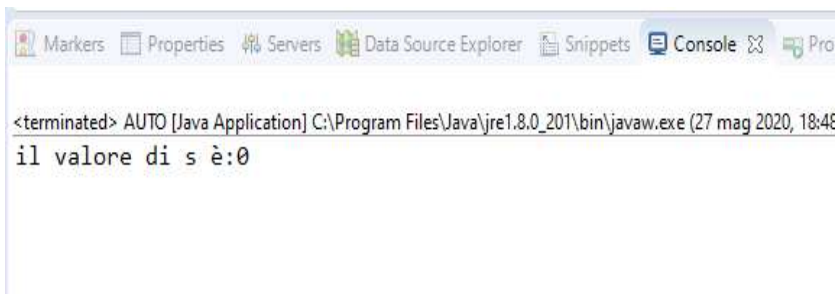
```
Markers Properties Servers Data Source Explorer Snippets Console  
<terminated> Prova (4) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (28 mag  
20
```

### Esempio: stampare un testo e il valore della variabile

Si aggiunge il testo seguito dal simbolo + e il nome della variabile:

```
int s = 0;  
System.out.println("il valore di s è:"+s);
```

output



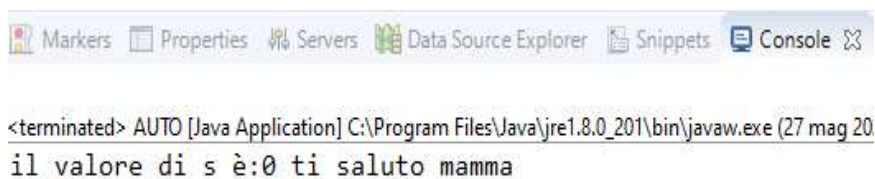
## Esempio: stampare più testo e valore della variabile

Unire il testo tramite l'uso dell'operatore +

```
int s = 0;

System.out.println("il valore di s è:"+ s + " ti saluto
mamma");
```

output



## Esempio: stampare più variabili nella stessa istruzione

```
int s = 0;
int r = 30;

System.out.println("il valore di s è:"+s+ " il valore di r
è:"+r);
```



## Output

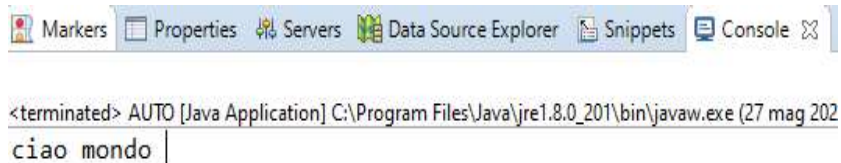


The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, and Console. The console text reads: <terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (27 mag 2021) il valore di s è:0 il valore di r è:30

### Esempio: stampare testo su due righe

```
System.out.print("ciao ");  
System.out.print("mondo ");
```

## Output



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, and Console. The console text reads: <terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (27 mag 2021) ciao mondo

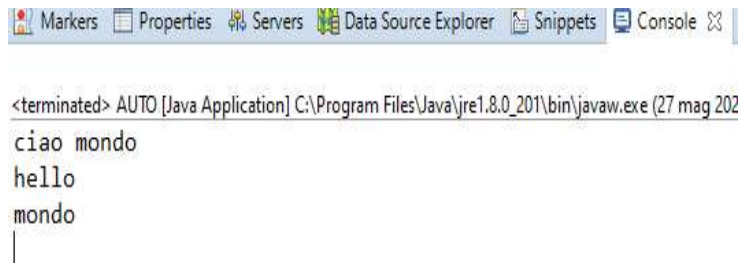
### Esempio: stampare due stringhe sulla stessa linea

Usare l'istruzione `System.out.print`, in questo caso verrà stampato il valore tra le parentesi, senza andare a capolinea a fine stampa.

```
System.out.print("ciao ");  
System.out.println("mondo ");
```

```
System.out.println("hello ");  
System.out.println("mondo ");
```

## Output



The screenshot shows an IDE console window with a toolbar at the top containing icons for Markers, Properties, Servers, Data Source Explorer, Snippets, and Console. The console text reads: <terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (27 mag 202, ciao mondo, hello, mondo, and a cursor on the next line.

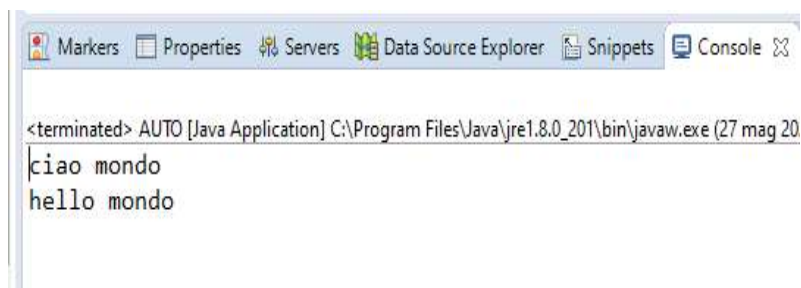
```
<terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (27 mag 202
ciao mondo
hello
mondo
|
```

## Esempio

```
System.out.print("ciao ");
System.out.println("mondo ");
```

```
System.out.print("hello ");
System.out.println("mondo ");
```

## Output



The screenshot shows an IDE console window with a toolbar at the top containing icons for Markers, Properties, Servers, Data Source Explorer, Snippets, and Console. The console text reads: <terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (27 mag 20, ciao mondo, and hello mondo.

```
<terminated> AUTO [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (27 mag 20
ciao mondo
hello mondo
```

---

## 1.9 Costrutto if else

Sintassi:

```
if(condizione){  
    istruzione 1  
}  
else  
{  
    istruzione 2  
}
```

Esegue il blocco istruzione1 (che può essere composto da una o più istruzioni) se la condizione tra parentesi tonde assume il valore booleano true, altrimenti esegue il blocco istruzione2.

Attenzione: le condizioni devono essere espressioni di tipo boolean (cioè delle condizioni che possono essere o true o false).

**Esempio:**

```
int a = 1;  
int b = 3;  
  
// OK! Sintassi corretta!  
if(a == b) { // con == si indica l uguaglianza ossia se a  
sarà uguale a b ritornerà true  
    ...  
}  
  
// NO! Sintassi errata: 'a' è un 'int' e non un 'boolean'  
if(a) {  
}  
  
// mentre è valida la scrittura
```

```

boolean c = ...;
if(c) {
    ...
}
//Il ramo else può anche essere omissso.

```

Nel caso in cui il blocco if sia composto solo da una istruzione le parentesi graffe possono essere omesse.

### Esempio

I due if sono equivalenti:

```

if (condizione) {
    System.out.println("Condizione verificata");
}

if (condizione)
    System.out.println("Condizione verificata");

```

All'interno di un blocco di istruzioni, si possono anche inserire altri if:

```

if (condizioneUno) {
    if (condizioneDue)
        System.out.println("Entrambe verificate");
    else
        System.out.println("Verificata solo
condizioneUno");
}

```

## Esempio

```
// variabili che rappresentano voti conseguiti
int votoScritto=7;
int votoOrale=5;

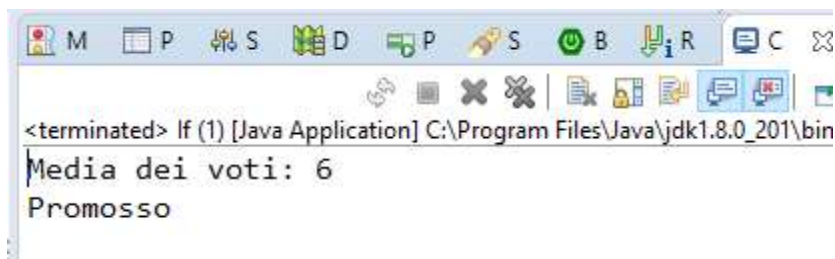
// totale dei voti
int totale=votoScritto+votoOrale;

// calcolo della media
int media=totale/2;

// stampa della media
System.out.println("Media dei voti: "+media);

if (media>=6)
    System.out.println("Promosso");
else
    System.out.println("Bocciato");
}
```

Output



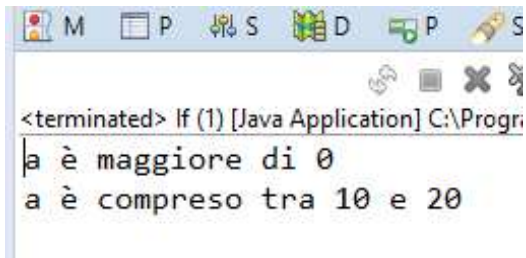
## Esempio

```
int a=10;

if (a>0)
    System.out.println("a è maggiore di 0");

if (a>=10 && a <=20)
    System.out.println("a è compreso tra 10 e 20");
```

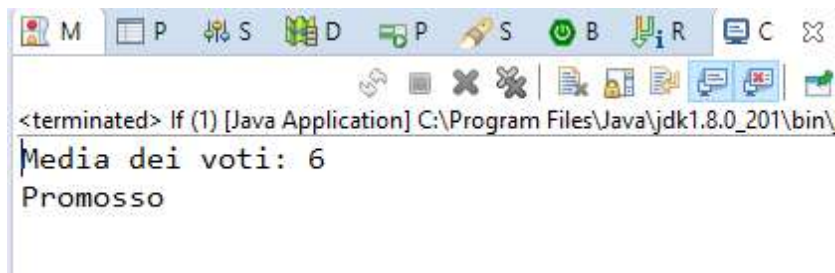
Output



## Esempio

```
public static void main(String[] args) {  
  
    int etaStudiante=20;  
    boolean studenteUniversitario=true;  
  
    if (studenteUniversitario)  
    {  
        System.out.println("Studente maggiorenne, può entrare");  
    }  
    else if (etaStudiante<18)  
    {  
        System.out.println("Studente minorenni, NON può  
entrare");  
    }  
    else  
        System.out.println("Studente maggiorenne, può entrare");  
}
```

Output



```
<terminated> If (1) [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\  
Media dei voti: 6  
Promosso
```

---

## 1.10 Operatori di confronto

Elenco di operatori che si possono usare per confrontare due variabili.

	Descrizione	Cosa restituisce
>	maggiore di	true se x è strettamente maggiore di y, false altrimenti
>=	maggiore o uguale di	true se x è maggiore o uguale di y, false altrimenti
<	minore di	true se x è strettamente minore di y, false altrimenti
<=	minore o uguale di	true se x è minore o uguale di y, false altrimenti
==	uguale	true se x è uguale a y, false altrimenti
!=	diverso	true se x è diverso da y, false altrimenti

## Esempio

```
if(x == 10)// x è uguale a 10  
    System.out.println("Condizione verificata");
```

```
if(x != 10)// x è diverso da 10  
    System.out.println("Condizione verificata");
```

```
if(x != y)// x è diverso da y  
    System.out.println("Condizione verificata");
```

```
if(x == y)// x è uguale a y  
    System.out.println("Condizione verificata");
```

---

## 1.11 Input da Tastiera



A questo punto del libro dobbiamo alzarci le maniche e fare un bel numero di esercizi, ma, per renderli più interessanti, dobbiamo prima spiegare come prendere dei dati in input da tastiera.

Con le conoscenze che si hanno fino ad ora non si possono capire le istruzioni che seguono (e vedremo più avanti), perciò si deve solo copiare e incollare, per ora è sufficiente questo.

La prima cosa da fare è inserire

```
import java.util.Scanner;
```

sotto l'istruzione package com, ottenendo:

```
package com;
```

```
import java.util.Scanner;
```

Dopodichè inserire sotto l'istruzione

```
public static void main(String[] args) {
```

questo

```
Scanner scan = new Scanner(System.in);
```

Il codice completo è:

```
package com;
```

```
import java.util.Scanner;
```

```
public class HelloWorld {
```

```
    public static void main(String[] args) {
```

```
        Scanner scan = new Scanner(System.in);
```

```
}  
}
```

A questo punto, ogni volta che si vuole prendere un numero in input scriviamo:

```
System.out.println("dammi numero in ingresso:");  
int n = scan.nextInt();
```

L'istruzione

```
System.out.println("dammi numero in ingresso:");
```

non è obbligatoria ma stampa un messaggio per far capire all'utente che il programma è in attesa di un numero, mentre

```
int n = scan.nextInt();
```

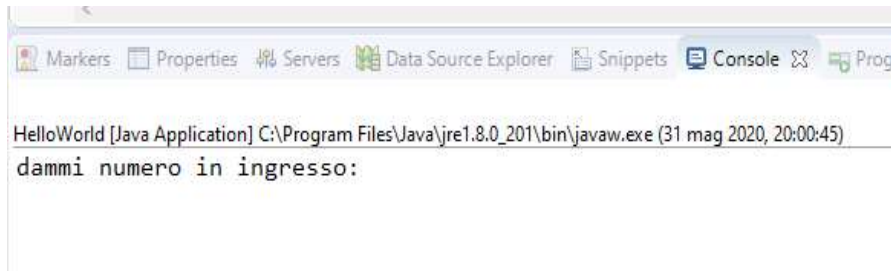
prende un numero inserito dalla tastiera (quindi si scrive il numero e si preme il tasto invio) e lo inserisce nella variabile n.

## Esempio

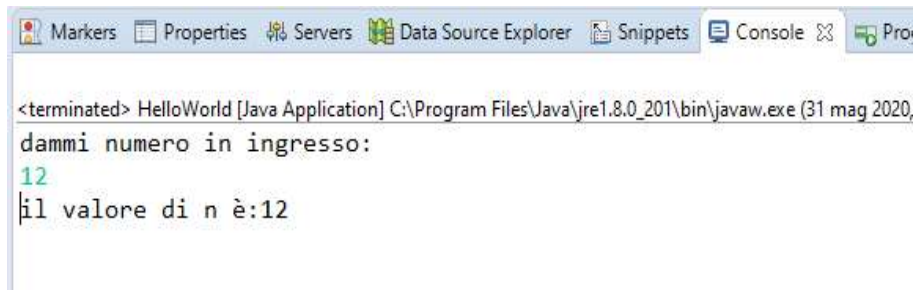
```
package com;  
import java.util.Scanner;  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
  
        System.out.println("dammi numero in ingresso:");  
        int n = scan.nextInt();
```

```
        System.out.println("il valore di n è:"+n);  
    }  
}
```

Lanciando il programma sulla console appare il messaggio



Inseriamo un numero qualsiasi, in questo esempio 12, e premiamo invio.



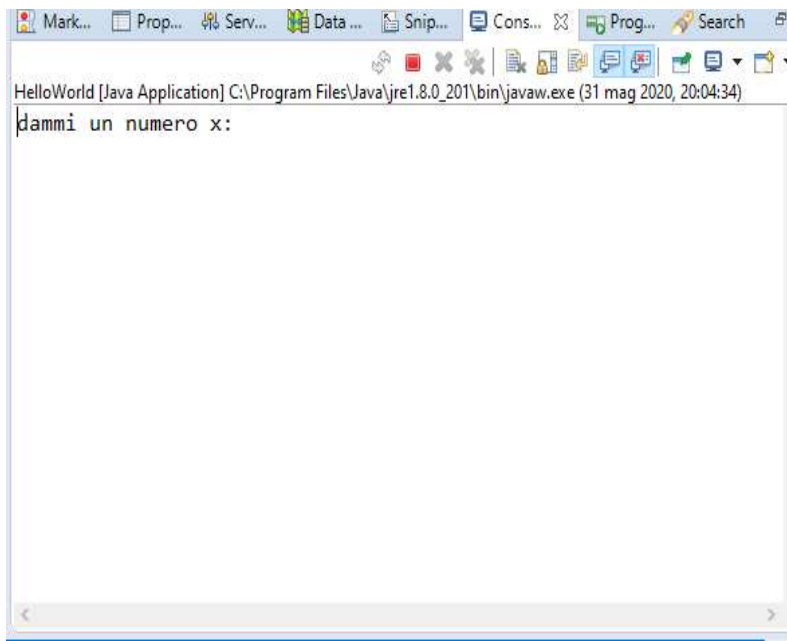
## Esempio

Prendere due numeri in ingresso e sommarli

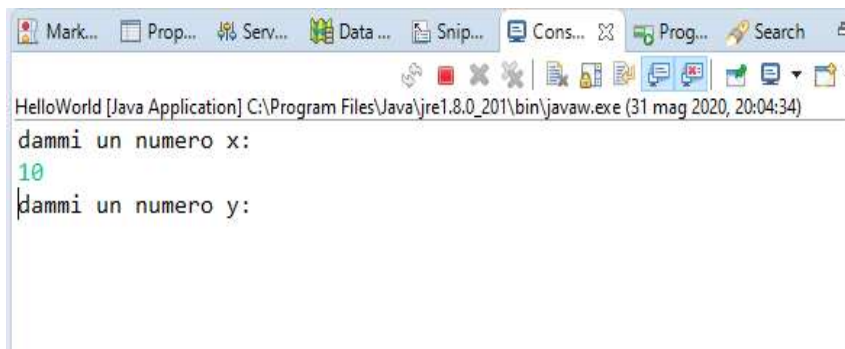
```
package com;  
  
import java.util.Scanner;  
  
public class HelloWorld {
```

```
public static void main(String[] args) {  
    Scanner scan = new Scanner(System.in);  
  
    System.out.println("dammi un numero x:");  
    int x = scan.nextInt();  
  
    System.out.println("dammi un numero y:");  
    int y = scan.nextInt();  
  
    int somma = x+y;  
  
    System.out.println("la somma è:"+somma);  
  
    }  
}
```

## Output



Inserisco ad esempio 10

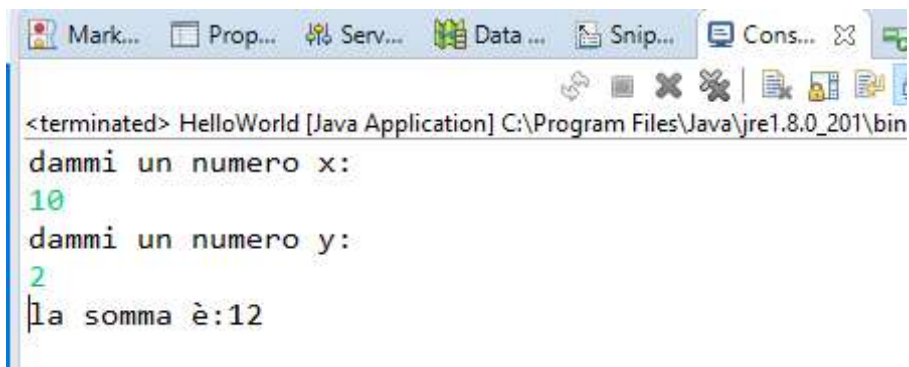


A screenshot of a Java console window titled 'HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (31 mag 2020, 20:04:34)'. The window has a menu bar with 'Mark...', 'Prop...', 'Serv...', 'Data ...', 'Snip...', 'Cons...', 'Prog...', and 'Search'. Below the menu bar is a toolbar with various icons. The main text area contains the following text: 'dammi un numero x:', '10', 'dammi un numero y:'. The text '10' is highlighted in green.

```
dammi un numero x:
10
dammi un numero y:
```

Inserisco, ad esempio, 2

La stampa finale sarà:



A screenshot of a Java console window titled 'HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (31 mag 2020, 20:04:34)'. The window has a menu bar with 'Mark...', 'Prop...', 'Serv...', 'Data ...', 'Snip...', 'Cons...', 'Prog...', and 'Search'. Below the menu bar is a toolbar with various icons. The main text area contains the following text: '<terminated> HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin', 'dammi un numero x:', '10', 'dammi un numero y:', '2', 'la somma è:12'. The text '10' and '2' are highlighted in green.

```
<terminated> HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0_201\bin
dammi un numero x:
10
dammi un numero y:
2
la somma è:12
```