



Dipartimento di Ingegneria e Scienze
dell'Informazione e Matematica

Università degli Studi dell'Aquila

Esercitazione 3

Corso di Laboratorio di Programmazione

Prof.ssa Monica Nesi - [Dr. Francesco Gallo](#)

<http://people.disim.univaq.it/francesco.gallo/laboratorio.html>

Documentazione

<https://www.oracle.com/technetwork/java/javase/documentation/jdk13-doc-downloads-5672541.html>

Oracle Technology Network / Java / Java SE / Documentation

Overview Downloads Documentation Community Technologies Training

Java SE Development Kit 13 Documentation

Java SE Development Kit 13 Documentation
You must accept the [Java SE Development Kit 13 Documentation License Agreement](#) to download this software.
Thank you for accepting the Java SE Development Kit 13 Documentation License Agreement; you may now download this software.

Product / File Description	File Size	Download
Documentation	48.24 MB	jdk-13_doc-all.zip

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)

Java SE
Java EE
Java ME
Java SE Subscription
Java Embedded
Java Card
Java TV
Community
Java Magazine

Input/Output Java

Input:

- Variabile di tipo **Scanner** che sfrutta la libreria **System.in**

Usando la libreria **System.in** abbiamo in ingresso un buffer con le informazioni sotto forma di byte da convertire.

Scanner si occupa di convertire il buffer in ingresso nel tipo di variabile che vogliamo (`int`, `string`, `double`, ...).

- È una classe offerta dalla libreria standard Java ed è presente nel **package java.util**
- Mette a disposizione un insieme di operazioni che consentono la lettura dell'input fornito dall'utente

Input/Output Java

Esempio:

```
import java.util.*;

public class TestLettura {

    public static void main(String args[]) {

        Scanner input = new Scanner(System.in); //istanzia un oggetto lettore di classe Scanner
        String s = input.next(); //legge una riga di testo e la memorizza nella variabile s
        System.out.println("Ho letto: " + s);

    }

}
```

Input/Output Java

<https://docs.oracle.com/en/java/javase/13/docs/api/java.base/java/util/Scanner.html>

(Alcuni!!!) metodi utili di questa classe:

- `int nextInt()`, legge un numero intero e lo restituisce al chiamante;
- `double nextDouble()`, legge un numero reale e lo restituisce al chiamante;
- `String nextLine()`, legge una linea di testo e la restituisce al chiamante;
- `String next()`, legge un "token" (un blocco di testo, una sottostringa), ossia una sequenza di caratteri contigui senza delimitatori, e lo restituisce al chiamante; questo metodo considera come delimitatori di sottostringhe gli spazi, i caratteri di tabulazione e i caratteri di newline;
- `boolean hasNextInt()`, restituisce vero se il prossimo token può essere interpretato come un numero intero, falso altrimenti;
- `boolean hasNextDouble()`, restituisce vero se il prossimo token può essere interpretato come un numero reale, falso altrimenti;
- `boolean hasNextLine()`, restituisce vero se in input è disponibile una ulteriore linea, falso altrimenti;
- `boolean hasNext()`, restituisce vero se in input è disponibile un ulteriore token, falso altrimenti;
- `Scanner useDelimiter(String)`, modifica il delimitatore dei token, dove la stringa passata come parametro può essere una espressione regolare.

Input/Output Java

([Alcuni!!!](#)) metodi utili di questa classe:

Il tipo di dato inserito deve essere coerente con l'istruzione usata.

Per esempio, se si utilizza il metodo `nextInt()` ma l'utente inserisce una lettera il programma darà un errore. Nel caso sia necessario leggere una stringa, si consiglia l'utilizzo di `nextLine()` per leggere TUTTA la riga inserita dall'utente.

- `int nextInt()`, legge un numero intero e lo restituisce al chiamante;
- `double nextDouble()`, legge un numero reale e lo restituisce al chiamante;
- `String nextLine()`, legge una linea di testo e la restituisce al chiamante;
- `String next()`, legge un "token" (un blocco di testo, una sottostringa), ossia una sequenza di caratteri contigui senza delimitatori, e lo restituisce al chiamante; questo metodo considera come delimitatori di sottostringhe gli spazi, i caratteri di tabulazione e i caratteri di newline;
- `boolean hasNextInt()`, restituisce vero se il prossimo token può essere interpretato come un numero intero, falso altrimenti;
- `boolean hasNextDouble()`, restituisce vero se il prossimo token può essere interpretato come un numero reale, falso altrimenti;
- `boolean hasNextLine()`, restituisce vero se in input è disponibile una ulteriore linea, falso altrimenti;
- `boolean hasNext()`, restituisce vero se in input è disponibile un ulteriore token, falso altrimenti;
- `Scanner useDelimiter(String)`, modifica il delimitatore dei token, dove la stringa passata come parametro può essere una espressione regolare.

Input/Output Java

Esempio:

```
import java.util.Scanner;

public class TestLettural {
    public static void main(String[] args) {

        int i, j;

        Scanner scanner = new Scanner(System.in);

        System.out.print("Primo numero: ");
        i = scanner.nextInt();
        System.out.print("Secondo numero: ");
        j = scanner.nextInt();
        System.out.print("La somma di "+i+" + "+j+" è: ");
        System.out.println(i + j);

        scanner.close();

    }
}
```

Input/Output Java

Esempio 2:

```
import java.util.Scanner;

class TestLettura2 {
    public static void main(String[] argv) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Inserire nome e cognome: ");
        String nomeCognome = scanner.next();

        System.out.print("Introduci città: ");
        String citta = scanner.next();

        System.out.println("Nome e cognome: " + nomeCognome);
        System.out.println("Città: " + citta);

        scanner.close();
    }
}
```

Input/Output Java

Esempio 2:

```
import java.util.Scanner;

class TestLettura {
    public static void main(String[] argv) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Inserire nome e cognome: ");
        String nomeCognome = scanner.next();

        System.out.print("Introduci città: ");
        String citta = scanner.next();

        System.out.println("Nome e cognome: " + nomeCognome);
        System.out.println("Città: " + citta);

        scanner.close();
    }
}
```

Il codice si comporta in modo corretto?

Input/Output Java

Esempio 2:

```
import java.util.Scanner;

class TestLettura {
    public static void main(String[] argv) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Inserire nome e cognome: ");
        String nomeCognome = scanner.next();

        System.out.print("Introduci città: ");
        String citta = scanner.next();

        System.out.println("Nome e cognome: " + nomeCognome);
        System.out.println("Città: " + citta);

        scanner.close();
    }
}
```

Il codice si comporta in modo corretto?

NO

Input/Output Java

Esempio 2:

Il codice si comporta in modo corretto? NO

- Nel buffer dell'oggetto `Scanner` il dato "Francesco Gallo" non viene trattato come un unico testo ma viene scomposto in due elementi "Francesco" e "Gallo" in virtù dello spazio bianco, che viene considerato un separatore dal parser dello Scanner.
- Quando scorriamo la lista degli elementi presenti in `Scanner` con il metodo `next()` per assegnarne il valore a delle variabili, il programma trova la prima volta il buffer vuoto e per quello attende l'input da parte dell'utente, scompone "Francesco Gallo" in "Francesco" e "Gallo" e assegna il valore "Francesco" a `nomeCognome`.
- Quando si invoca per la seconda volta il metodo `next()` il programma GIÀ TROVA un elemento nel buffer (che è "Gallo") quindi lo assegna direttamente alla variabile `citta` senza aspettare l'input dall'utente, andando poi a stampare i valori contenuti nelle due variabili.

Input/Output Java

Esempio 2:

Soluzione: usare il metodo `nextLine()`

```
import java.util.Scanner;

class TestLettura3 {
    public static void main(String[] argv) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Inserire nome e cognome: ");
        String nomeCognome = scanner.nextLine();

        System.out.print("Introduci città: ");
        String citta = scanner.nextLine();

        System.out.println("Nome e cognome: " + nomeCognome);
        System.out.println("Città: " + citta);

        scanner.close();
    }
}
```

Input/Output Java

Esercizio 1:

Scrivere un programma Java che richiede e stampa la propria anagrafica (Nome, Cognome, Indirizzo, Numero Civico, Data di Nascita, Codice Fiscale).

Esercizio 2:

Scrivere un programma Java che richiede e stampa i dati di un libro (Titolo, Autore, Editore, Numero di pagine, Prezzo).

Esercizio 3:

Scrivere un programma Java che presa in input una stringa qualsiasi (composta da lettere dell'alfabeto e da cifre comprese tra 0 e 9) restituisce in output: una stringa contenente solo le lettere contenute nell'input e un intero che contiene la somma degli interi contenuti nell'input. (Es. Input=Francesco42Gallo, Output=FrancescoGallo, 6)

Input/Output Java

Esercizio 2: Scrivere un programma Java che richiede e stampa i dati di un libro (Titolo, Autore, Editore, Numero di pagine, Prezzo).

```
import java.util.Scanner;

class Libro {

    public static void main (String[] args) {
        String titolo, autore, editore;
        int numPagine;
        float prezzo;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Inserire il titolo");
        titolo = scanner.nextLine();
        System.out.println("Inserire l'autore");
        autore = scanner.nextLine();
        System.out.println("Inserire l'editore");
        editore = scanner.nextLine();
        System.out.println("Inserire il numero di pagine");
        numPagine = scanner.nextInt();
        System.out.println("Inserire il prezzo");
        prezzo = scanner.nextFloat();
        System.out.println();
        System.out.println(titolo);
        System.out.println(autore);
        System.out.println(editore);
        System.out.println(numPagine);
        System.out.println(prezzo);

        scanner.close(); } }
```

Input/Output Java

Input mediante finestre di dialogo:

```
String input = JOptionPane.showInputDialog("Ciao Mondo!")
```

Visualizza una finestra di dialogo e resta in attesa dell'inserimento dei dati da parte dell'utente.

metodo statico della classe `JOptionPane` (package `javax.swing`)

Il valore restituito da una finestra di dialogo è sempre di tipo **stringa**

Se si deve leggere un numero bisogna effettuare una **conversione di tipo**.

Input/Output Java

Esempio 3:

```
import javax.swing.JOptionPane;

class EsempioDialog {

    public static void main(String[] args) {

        String dato = JOptionPane.showInputDialog("Inserire il dato");
        JOptionPane.showMessageDialog(null, "Il dato inserito e' " + dato);
    }
}
```



Output mediante finestre di dialogo

showMessageDialog è un **metodo statico** della classe JOptionPane (package javax.swing)

Input/Output Java

Esercizio 3:

Scrivere un programma Java che richiede due numeri in input e stampa il risultato della somma e il prodotto (mediante finestre di dialogo)

Input/Output Java

Esercizio 3:

Scrivere un programma Java che richiede due numeri in input e stampa il risultato della somma e il prodotto (mediante finestre di dialogo)

```
import javax.swing.JOptionPane;

class Operazioni {
    public static void main(String[] argv) {

        int somma, prodotto;
        int valUno;
        int valDue;

        String valoreUno = JOptionPane.showInputDialog("Inserire il primo valore");
        String valoreDue = JOptionPane.showInputDialog("Inserire il secondo valore");

        valUno = Integer.parseInt(valoreUno);
        valDue = Integer.parseInt(valoreDue);
        somma = valUno + valDue;
        prodotto = valUno * valDue;

        JOptionPane.showMessageDialog(null, "La somma di " + valoreUno + " + " + valoreDue + " è " +
somma + "\n" + "Il prodotto di " + valoreUno + " * " + valoreDue + " è " + prodotto);
    }
}
```

Input/Output Java

Esercizio 3:

Scrivere un programma Java che richiede due numeri in input e stampa il risultato della somma e il prodotto (mediante finestre di dialogo)

```
import javax.swing.JOptionPane;
```

```
class Operazioni {  
    public static void main(String[] argv) {
```

```
        int somma, prodotto;  
        int valUno;  
        int valDue;
```

```
        String valoreUno = JOptionPane.showInputDialog("Inserire il primo valore");  
        String valoreDue = JOptionPane.showInputDialog("Inserire il secondo valore");
```

```
        valUno = Integer.parseInt(valoreUno);  
        valDue = Integer.parseInt(valoreDue);  
        somma = valUno + valDue;  
        prodotto = valUno * valDue;
```

```
        JOptionPane.showMessageDialog(null, "La somma di " + valoreUno + " + " + valoreDue + " è " +  
somma + "\n" + "Il prodotto di " + valoreUno + " * " + valoreDue + " è " + prodotto);  
    }  
}
```

- **int Integer.parseInt**(String s) converte una stringa in un valore int;
- **long Long.parseLong**(String s) converte una stringa in un valore long;
- **double Double.parseDouble**(String s) converte una stringa in un valore double;
- **float Float.parseFloat**(String s) converte una stringa in un valore float.

Input/Output Java

Input da linea di comando:

Esercizio 4:

Scrivere un programma Java che richiede due numeri in input e stampa il risultato della somma e il prodotto (mediante terminale)

Input/Output Java

Input da linea di comando:

Esercizio 4:

Scrivere un programma Java che richiede due numeri in input e stampa il risultato della somma e il prodotto (mediante terminale)

```
class Esercizio4 {  
    public static void main(String[] args) {  
  
        int somma, prodotto;  
        int valUno;  
        int valDue;  
  
        valUno = Integer.parseInt(args[0]);  
        valDue = Integer.parseInt(args[1]);  
  
        somma = valUno + valDue;  
        prodotto = valUno * valDue;  
  
        System.out.println("Somma: "+somma);  
        System.out.println("Prodotto: "+prodotto);  
    }  
}
```

Input/Output Java

Output su terminale - `System.out.print()`:

Escape Sequences

Escape Sequence	Description
<code>\t</code>	Insert a tab in the text at this point.
<code>\b</code>	Insert a backspace in the text at this point.
<code>\n</code>	Insert a newline in the text at this point.
<code>\r</code>	Insert a carriage return in the text at this point.
<code>\f</code>	Insert a formfeed in the text at this point.
<code>\'</code>	Insert a single quote character in the text at this point.
<code>\"</code>	Insert a double quote character in the text at this point.
<code>\\</code>	Insert a backslash character in the text at this point.

Input/Output Java

Esercizio 5:

```
-----  
Email System - Display students  
-----
```

Grade	Last Name	First Name	Student Number	Parent Email
12	Frost	Johnny	313523510	example@hotmail.com

Description

If the user selects the fourth option, all students are printed on the screen in the order in which they were found on the data file or were sorted by the program. The information will be printed in a neatly formatted table so that the user may easily read through the data.

Input/Output Java

Output su terminale - `System.out.printf()` o `System.out.format()`:

`System.out.printf(...)` <==> `System.out.format(...)`

```
class TestFormat {  
  
    public static void main(String[] args) {  
        long n = 461012;  
        System.out.format("%d%n", n);           // --> "461012"  
        System.out.format("%08d%n", n);        // --> "00461012"  
        System.out.format("%+8d%n", n);        // --> " +461012"  
        System.out.format("% ,8d%n", n);       // --> " 461,012"  
        System.out.format("%+,8d%n%n", n);     // --> "+461,012"  
  
        double pi = Math.PI;  
  
        System.out.format("%f%n", pi);         // --> "3.141593"  
        System.out.format("%.3f%n", pi);       // --> "3.142"  
        System.out.format("%10.3f%n", pi);     // --> "      3.142"  
        System.out.format("%-10.3f%n", pi);    // --> "3.142"  
  
    }  
}
```

Input/Output Java

Esercizio 6:

```
-----  
Email System - Display students  
-----  
  
Grade      Last Name  First Name  Student Number  Parent Email  
12         Frost      Johnny     313523510      example@hotmail.com
```

Description

If the user selects the fourth option, all students are printed on the screen in the order in which they were found on the data file or were sorted by the program. The information will be printed in a neatly formatted table so that the user may easily read through the data.

Input/Output Java

Esercizio 7:

Scrivere un programma che prevede un array di 10 numeri interi contenente valori a piacere e ne stampa gli elementi secondo il seguente ordine: il primo, l'ultimo, il secondo, il penultimo, il terzo, il terzultimo, ecc.

Esercizio 8:

Scrivere un programma che prevede un array di 10 numeri interi contenente valori a piacere e stampa **Pari e dispari uguali** se la somma dei numeri in posizioni pari dell'array è uguale alla somma dei numeri in posizioni dispari, altrimenti il programma stampa **Pari e dispari diversi**.