

Fondamenti di Informatica T-1

Input/Output, Variabili, Tipi e Stringhe

Tutor:
Angelo Feraudo
angelo.feraudo2@unibo.it

a.a. 2017/2018

TUTOR

Angelo Feraudo

- **Email:** `angelo.feraudo2@unibo.it`
- **Ricevimento:** al **termine** delle esercitazioni o previo **appuntamento per email**

Sito del corso

- `http://www-db.disi.unibo.it/courses/FIT1-LZ/`

VARIABILI

- Una variabile è un'astrazione di una cella di memoria nella quale viene salvato un determinato dato
- È quindi caratterizzata da:
 - ▶ **Tipo** (int, float, char ,...)
 - ▶ **Nome** (per convenzione scritti con la lettera minuscola)
 - ▶ **Valore**
- La dimensione della cella di memoria dipende dal tipo
- **Dichiarazione:** <tipo><nome>;
- **Assegnamento:** <nome> = <valore>;

VARIABILI: Esempio pratico

`int n1;` ← Dichiarazione

`n1 = 10;` ← Assegnamento

```
System.out.println("Una variabile intera: " + n1);
```

TIPI: Breve introduzione

I tipi ammessi in un programma Java sono i seguenti:

- otto tipi primitivi definiti dal linguaggio (*boolean, byte, char, short, int, long, float, double*)
- tipi riferimento (*vedremo quando faremo le classi*)
- void (tipicamente usato per indicare che una funzione non restituisce nulla, ad esempio il **main**)

La differenza tra *tipi primitivi* e *tipi riferimento* la vedremo in seguito.

OPERATORI ARITMETICI

- Somma $\rightarrow a+b$;
- Sottrazione $\rightarrow a-b$;
- Moltiplicazione $\rightarrow a*b$;
- Divisione $\rightarrow a/b$;
- Resto $\rightarrow a\%b$;
- Incremento/decremento:
 - $a = a+n$; $\rightarrow a+=n$; \rightarrow (se $n=1$) $a++$; *incremento postfisso*
 - $a = a-n$; $\rightarrow a-=n$; \rightarrow (se $n=1$) $a--$; *decremento postfisso*

OPERATORI ARITMETICI(2)

Esiste anche la possibilità di *incremento/decremento* **prefisso** realizzabile con la seguente sintassi:

++<variabile>

--<variabile>

Esempio:

```
int c = 5;  
int b = 30 / c++; /* b == 6, c == 6 */  
int d = 6 + --c; /*d == 11,c == 5*/
```

OPERATORI LOGICI e COMPARATIVI

- Uguale → `==`;
(ATTENZIONE: è molto diverso dal "=" dell'assegnamento)
- Diverso → `!=`;
- Maggiore → `>`;
- Maggiore Uguale → `>=`;
- Minore → `<`;
- Minore Uguale → `<=`;
- NOT logico → `!`;
- AND logico → `&&`;
- OR logico → `||`;

LA CLASSE MATH

La classe **Math** è una particolare classe che contiene numerosi metodi statici (capiremo in seguito cosa sono esattamente) e costanti utili per i calcoli matematici.

Per una descrizione completa delle funzionalità, visitare la pagina: <https://docs.oracle.com/javase/9/docs/api/java/lang/Math.html>.

Alcune esempi di funzioni utili:

- `int valore assoluto = Math.abs(x)`; → Restituisce il valore assoluto di `x`
- `double quadrato = Math.pow(x,2)`; → Restituisce il quadrato di `x`
- `double radice = Math.sqrt(x)`; → Restituisce la radice quadrata di `x`

INPUT da TASTIERA: la CLASSE SCANNER

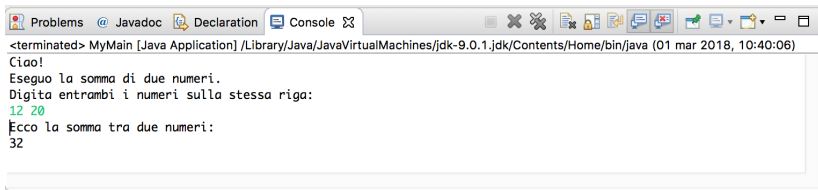
Per leggere dati da tastiera si utilizza la classe **Scanner** che si trova nel package `java.util`

- Per poterla utilizzare bisogna importarla con l'istruzione **`import java.util.Scanner`** nelle prime righe del programma
- Per creare un'istanza:
`Scanner tastiera = new Scanner(System.in);`
- Lettura di un numero intero:
`int i = tastiera.nextInt();`
- Lettura di un numero reale:
`double d = tastiera.nextDouble();`
- Chiusura tastiera (ricordarsi di farlo quando non serve più):
`tastiera.close();`

CLASSE SCANNER: Esempio(1)

```
import java.util.Scanner; ← Carica la classe Scanner dal  
                           package (libreria) java.util  
  
public class PrimoProgramma { ← Nome della classe, a scelta  
  
    public static void main(String[] args) {  
  
        System.out.println("Ciao!"); ← Invia l'output allo schermo  
        System.out.println("Eseguo la somma di due numeri.");  
        System.out.println("Digita entrambi i numeri sulla stessa riga:");  
  
        int n1, n2; ← Indica che n1 e n2 sono variabili  
                   che contengono interi  
  
        Scanner tastiera = new Scanner(System.in); ← Predispone il programma affinché  
                                                       possa leggere l'input dalla tastiera  
  
        n1 = tastiera.nextInt(); ← Legge un numero intero dalla tastiera  
        n2 = tastiera.nextInt();  
  
        System.out.println("Ecco la somma dei due numeri:");  
        System.out.println(n1 + n2);  
    }  
}
```

CLASSE SCANNER: Esempio(2)










The screenshot shows an IDE console window with the following content:

```
<terminated> MyMain [Java Application] /Library/Java/JavaVirtualMachines/jdk-9.0.1.jdk/Contents/Home/bin/java (01 mar 2018, 10:40:06)
Ciao!
Eseguo la somma di due numeri.
Digita entrambi i numeri sulla stessa riga:
12 20
Ecco la somma tra due numeri:
32
```

ORA TOCCA A VOI

Create un progetto che chiamerete **Esercitazione2** e, al suo interno, create **un package diverso per ogni esercizio**. In ogni package create una classe che conterrà il codice del vostro esercizio.

La struttura del vostro progetto dovrà essere così:

- ▼  Esercitazione2
 - ▶  JRE System Library [JavaSE-9]
 - ▼  src
 - ▼  esercizio1
 - ▶  Esercizio1.java
 - ▼  esercizio2
 - ▶  Esercizio2.java

Esercizio 1

Scrivere un programma che:

- Chieda all'utente di inserire l'età
- Stampi a video l'età incrementata di uno

Esempio di output

```
Inserisci la tua età:
```

```
23
```

```
Tra un anno avrai: 24 anni
```

Esercizio 1: Soluzione

```
public static void main(String []args) {  
  
    int eta = 0;  
    Scanner tastiera = new Scanner(System.in);  
  
    System.out.println("Inserisci la tua età:");  
    eta = tastiera.nextInt();  
  
    eta ++;  
    System.out.println("Tra un anno avrai: " + eta + " anni");  
  
    tastiera.close();  
}
```

Esercizio 2

Scrivere un programma che:

- Chieda all'utente di inserire l'anno di nascita
- Chieda all'utente un numero n che rappresenta gli anni
- Stampi a video l'anno in cui festeggerà o ha festeggiato il suo n-esimo compleanno

Esempio di output

Inserisci anno di nascita:

1995

Inserisci numero n:

23

Il tuo n-esimo compleanno sarà nel: 2018

Esercizio 2: Soluzione

```
public class Esercizio2 {  
    public static void main(String []args) {  
        int anno, n, annoFinale;  
        Scanner tastiera = new Scanner(System.in);  
  
        System.out.println("Inserisci anno di nascita: ");  
        anno = tastiera.nextInt();  
  
        System.out.println("Inserisci numero n: ");  
        n = tastiera.nextInt();  
  
        annoFinale = anno + n;  
        System.out.println("Il tuo n-esimo compleanno sarà nel " + annoFinale);  
        tastiera.close();  
    }  
}
```

STRINGHE: Definizione e creazione

- La **Stringa** è un tipo di dato che rappresenta una sequenza ordinata di caratteri
- Le stringhe in Java sono **oggetti**, istanze della classe `String`
- **Creazione di una stringa in Java:**
 - ▶ `String es_stringa1 = "Ciao";`
 - ▶ `String es_stringa2 = new String("Ciao");`

N.B. `es_stringa1` è possibile in quanto il compilatore crea una variabile di tipo `String` ogni volta che incontra una sequenza racchiusa fra doppi apici

STRINGHE: Operazioni(1)

package: `java.lang.String`

- Prendiamo come **esempio** la seguente stringa
 - ▶ `String saluto = "Ciao";`

- **Concatenazione**

- ▶ `saluto = saluto + " a tutti!";`
ora la stringa è diventata: **"Ciao a tutti!"**

- **Calcolo della lunghezza: `length()`**

- ▶ `int lunghezza = saluto.length();`

La lunghezza della nostra stringa in questo caso è **13**

(N.B. CONTA GLI SPAZI TRA LE PAROLE E LA PUNTEGGIATURA!)

ATTENZIONE!

indice 0

indice $\text{length}()-1$



- La posizione di un carattere in una stringa si conta a partire da 0 e non da 1

STRINGHE: Operazioni (2)

- Riprendiamo come **esempio** la stringa
 - ▶ `String saluto = "Ciao a tutti!";`
- **Indice della prima occorrenza** di una data sottostringa: `indexOf()`
 - ▶ `int prima_occorrenza = saluto.indexOf("tutti");`
la prima occorrenza di "tutti" è **7**.
N.B. Per contare a partire dall'ultimo carattere della stringa, posso usare `lastIndexOf()`
- **Estrazione dell' i-esimo carattere:** `charAt()`
 - ▶ `char prima_lettera = saluto.charAt(0);`
l'indice del primo carattere è **0** → `prima_lettera = 'C'`

STRINGHE: Operazioni (3)

- **Estrazione di una sottostringa** dati due indici: **substring()**

es: `int iniziale = 7;`

`int finale = 12;`

- ▶ `String sottostringa = saluto.substring(iniziale, finale);`
→ **sottostringa = "tutti"**

N.B. L'indice finale **12** è escluso nella selezione dei caratteri.

- **Trasformazione maiuscole-minuscole:** **toUpperCase()** e **toLowerCase()**

- ▶ `String saluto_maiuscolo = saluto.toUpperCase();`
→ **saluto_maiuscolo = "CIAO A TUTTI!"**

- **Sostituzione di una sottostringa ad ogni occorrenza:** **replaceAll()**

- ▶ `String nuovo_saluto = saluto.replaceAll("a","AAA");`
→ **nuovo_saluto = "ciAAAO AAA tutti!"**

Utilissima per **eliminare gli spazi bianchi**. Come?

STRINGHE: OPERAZIONI(4)

Altre operazioni sono disponibili al seguente link:

`https://docs.oracle.com/javase/9/docs/api/java/lang/String.html`

STRINGHE: Lettura

- Metodo **next()** della classe Scanner:
Scanner tastiera = new Scanner(System.in);
String s1 = tastiera.**next()**;
s1 è il contenuto della console **fino al primo spazio bianco**
- Metodo **nextLine()** della classe Scanner:
String s2 = tastiera.**nextLine()**;
s2 è un'intera riga della console
N.B. **nextLine()** ha un comportamento anomalo se preceduto da una lettura (ad es. **next()** o **nextInt()** o **nextDouble()**...):
tali metodi non "consumano" l'ENTER al termine dell'input e quindi **una successiva nextLine() perderebbe l'acquisizione dell'input raccogliendo una stringa vuota.**

ESERCIZIO 3

- Una libreria utilizza un sistema di catalogazione usando, per ogni libro, dei **codici strutturati nel seguente modo**:
 - ▶ Le **prime due lettere, in maiuscolo**, sono le prime due lettere del nome dell'autore
 - ▶ L'anno di pubblicazione (da acquisire come **INTERO**), **separato da un trattino** dalle precedenti lettere
 - ▶ Le **prime tre lettere del titolo del libro, in maiuscolo**, separate da un trattino dall'anno di pubblicazione
 - ▶ La **prima e l'ultima lettera del nome della casa editrice (in minuscolo e tra parentesi tonde)**
- Scrivere un programma che, inseriti da input il nome dell'autore, l'anno di pubblicazione e il nome della casa editrice, **stampi a video il corrispondente codice**.

ESERCIZIO 3 (ESEMPI e SUGGERIMENTI)

- **Esempio di output:**

Inserire il nome dell'autore: **Antoine de Saint-Exupéry**
Inserire l'anno di pubblicazione: **2000**
Inserire il titolo del libro: **Il Piccolo Principe**
Inserire il nome della casa editrice: **Bompiani**
Il codice prodotto è AN-2000-ILP(bi)

- Le lettere del nome dell'autore, il titolo del libro e il nome della casa editrice **NON devono contenere spazi** al fine di generare il codice. Dovremo eliminarli prima di estrarre i caratteri utili.