

# Fondamenti di informatica T-1 (A – K)

## Esercitazione 5: controllo del flusso

AA 2018/2019

Tutor:

**Lorenzo Rosa**

[lorenzo.rosa@unibo.it](mailto:lorenzo.rosa@unibo.it)

# Esercitazione 5

Introduzione al calcolatore e Java

Linguaggio Java, basi e controllo del flusso

Eclipse ed esercizi di consolidamento

Stringhe ed array

Metodi, classi, oggetti

Ereditarietà e polimorfismo

Collezioni Java e interfacce

Esercizi d'esame

# String

- La classe `String` rappresenta una sequenza di caratteri. Ha una classe ad hoc, vista la sua importanza.
- Permette la concatenazione tramite l'operatore `'+'` che abbiamo visto utilizzando

```
System.out.println("val: " + variabile);
```
- Ha moltissimi **metodi di utilità**, ne vedremo alcuni ma è consigliato dare uno sguardo alla documentazione:  
<https://docs.oracle.com/javase/10/docs/api/java/lang/String.html>
- Si può istanziarne un oggetto in due differenti modi :
  - `String s1 = new String("Stringa uno");`
  - `String s2 = "Stringa due";`

# Immutabilità

Le stringhe, istanze della classe `String`, sono

**costanti ed immutabili,**

ovvero il loro valore non può essere modificato dopo la loro creazione.

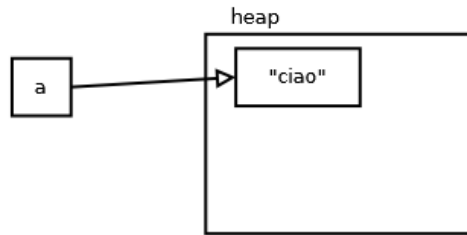


Infatti tutti i metodi che effettuano delle modifiche non modificano l'oggetto su cui è invocato il metodo ma creano e restituiscono una nuova stringa.

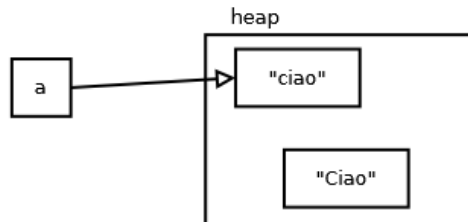
# Immutabilità, esempio

```
1 public class Main
2 {
3     public static void main(String[] argv)
4     {
5         String a = new String("ciao");
6         a.replace('c', 'C');
7         System.out.println(a);
8     }
9 }
```

Prima della replace



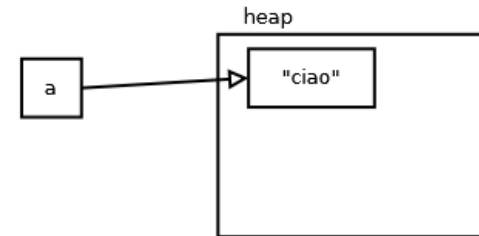
Dopo la replace



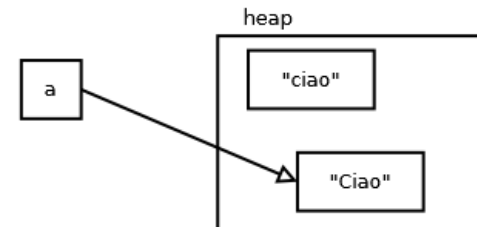
In memoria, oltre alla stringa cui si riferisce *a*, verrà creato un nuovo oggetto tramite il metodo *replace*. Questa stringa non avrà, però, alcuna variabile *reference* associata.

```
1 public class Main
2 {
3     public static void main(String[] argv)
4     {
5         String a = new String("ciao");
6         a = a.replace('c', 'C');
7         System.out.println(a);
8     }
9 }
```

Prima della replace



Dopo la replace



In questo caso si cambia la variabile *reference* e quindi non si modifica la stringa ma se ne crea un'altra (della prima si occuperà il *garbage collector*).

# Alcuni esempi

Scrivere un programma che concateni due stringhe s1 ed s2 e rimpiazzhi tutte le 'i' contenute nella stringa risultante s3 = s1+s2 con delle 'Z' :

```
public static void main(String[] args) {  
  
    String s1 = "L'esame di informatica";  
    String s2 = " è il mio preferito!";  
    String s3 = s1.concat(s2);  
    System.out.println("La stringa s3 prima della modifica e' : "  
+ s3);  
    s3 = s3.replace('i', 'Z');  
    System.out.println("La stringa s3 dopo la modifica e' : " +  
s3);  
  
}
```

# Alcuni esempi

Scrivere un programma che data una stringa di almeno 2 caratteri,  $s_1s_2$ , costruisca la stringa  $s_2s_1$ , cioè speculare rispetto al carattere intermedio.

Esempio : Ciao -> aoCi.    informatica -> maticainfor

```
public static void main(String [] args) {  
  
    java.util.Scanner sc = new java.util.Scanner(System.in);  
    System.out.println("Inserisci una stringa");  
    String parola = sc.nextLine();  
    int lunghezza, m;  
    lunghezza = parola.length();  
    m = lunghezza/2;  
    System.out.print(parola.substring(m, lunghezza));  
    System.out.print(parola.substring(0, m));  
}
```

# Uguaglianza tra stringhe

E' fondamentale ricordare che due stringhe non si comparano tramite l'operatore '==' come visto, ad esempio, per gli interi.

Il codice:

```
String s1 = "Ciao"; String s2 = "Ciao";  
if ( s1 == s2 ) System.out.println("Le due stringhe sono  
uguali.");
```

*Non funziona come atteso perché l'operatore '==' controlla se le variabili referenziano lo stesso oggetto (cioè se è puntata la stessa area di memoria), non se il valore della stringa è lo stesso.*

Se vogliamo sapere se due stringhe sono uguali dobbiamo usare il metodo *equals*

```
if( s1.equals(s2) )  
    System.out.println("Le due stringhe sono uguali.");
```



# Esercizio 1 - Calcolatrice

- Si scriva un programma (classe Calcolatrice) che funzioni da calcolatrice.
- Dati in ingresso come input
  - 2 numeri `double`
  - una di queste stringhe: "+", "-", "\*", "/"

il programma dovrà restituire

- il risultato dell'operazione, oppure
- "undefined" se l'operatore inserito non è uno dei quattro ammessi oppure se è violata una qualche preconditione.

# Esercizio 1 - Calcolatrice

```
C:\Users\maura\Documents\UNIBO\Tutorato\Materiali>java Calcolatrice
Benvenuto nella tua calcolatrice
Inserisci il primo numero: 78
Inserisci il secondo numero: 90
Inserisci l'operatore: +
Risultato di 78.0 + 90.0 = 168.0
```

- Per leggere una stringa:

```
String stringa = tastiera.next();
```

- Come si confrontano le stringhe?

- Le stringhe sono oggetti, in Java "==" tra oggetti ha un significato diverso da quello qui necessario.
- Quindi, si usa: `stringa.equals( altrastringa );`

# Esercizio 1 – Soluzione (1/2)

```
import java.util.Scanner;

public class Calcolatrice {

    public static void main(String args[]) {

        Scanner tastiera = new Scanner(System.in);

        //Lettura da input

        System.out.println("Benvenuto nella tua calcolatrice");

        System.out.print("Inserisci il primo numero: ");

        double op1 = tastiera.nextDouble();

        System.out.print("Inserisci il secondo numero: ");

        double op2 = tastiera.nextDouble();

        System.out.print("Inserisci l'operatore: ");

        String operatore = tastiera.next();
    }
}
```

# Esercizio 1 – Soluzione (2/2)

```
double result = 0;
if(operatore.equals("+")) {
    result = op1+op2;
    System.out.println("Risultato di "+op1+" + "+op2+" = "+ result);
}
else if(operatore.equals("-")) {
    result = op1-op2;
    System.out.println("Risultato di "+op1+" - "+op2+" = "+ result);
}
else if(operatore.equals("*")) {
    result = op1*op2;
    System.out.println("Risultato di "+op1+" * "+op2+" = "+ result);
}
else if(operatore.equals("/") && op2!=0) {
    result = op1/op2;
    System.out.println("Risultato di "+op1+" / "+op2+" = "+ result);
}
else {
    System.out.println("Undefined");
}}
```

## Esercizio 2 – Calcolo della media

- Si scriva un programma (classe MediaPesata) che calcoli la media pesata degli esami sostenuti dall'utente.
- L'utente, per ogni esame, deve inserire:
  - voto (intero, compreso tra 18 e 31) [31 = 30L]
  - numero crediti (intero, maggiore di zero)
- Supponiamo, per ora, che il numero di esami sostenuti sia noto e pari a **tre esami**.

## Esercizio 2 – Calcolo della media

$$\text{media pesata tre esami} = \frac{\sum_{k=1}^3 (\text{voto}_k * \text{crediti}_k)}{\sum_{k=1}^3 \text{crediti}_k}$$

- Quindi:

```
int totale = 0;
```

```
int totaleCrediti = 0;
```

- Per il primo esame:

*... leggo il voto e i crediti ...*

```
totale = totale + (voto1 * crediti1);
```

```
totaleCrediti = totaleCrediti + crediti1;
```

- E così via...

## Esercizio 2 – Soluzione (1/2)

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

        Scanner tastiera = new Scanner(System.in);

        //Variabili di accumulo
        int totale = 0;
        int totaleCrediti = 0;

        //Primo esame
        System.out.println("Benvenuto nel calcolatore della media");
        System.out.print("Inserisci il voto dell'esame 1: ");
        int voto1 = tastiera.nextInt();
        System.out.print("Inserisci i crediti dell'esame 1: ");
        int crediti1 = tastiera.nextInt();
        totale += (voto1 * crediti1);
        totaleCrediti += crediti1;
    }
}
```

## Esercizio 2 – Soluzione (2/2)

```
//Secondo esame
System.out.print("Inserisci il voto dell'esame 2: ");
int voto2 = tastiera.nextInt();
System.out.print("Inserisci i crediti dell'esame 2: ");
int crediti2 = tastiera.nextInt();
totale += (voto2 * crediti2);
totaleCrediti += crediti2;

//Terzo esame
System.out.print("Inserisci il voto dell'esame 3: ");
int voto3 = tastiera.nextInt();
System.out.print("Inserisci i crediti dell'esame 3: ");
int crediti3 = tastiera.nextInt();
totale += (voto3 * crediti3);
totaleCrediti += crediti3;

//Calcolo finale
double result = (double)totale/(double)totaleCrediti;
System.out.println("MEDIA PESATA: " + result);
```



## Esercizio 2 – Soluzione (2/2)

```
//Secondo esame
System.out.print("Inserisci il voto dell'esame 2: ");
int voto2 = tastiera.nextInt();
System.out.print("Inserisci i crediti dell'esame 2: ");
int crediti2 = tastiera.nextInt();
totale += (voto2 * crediti2);
totaleCrediti += crediti2;
```

Trova le differenze...

```
//Terzo esame
System.out.print("Inserisci il voto dell'esame 3: ");
int voto3 = tastiera.nextInt();
System.out.print("Inserisci i crediti dell'esame 3: ");
int crediti3 = tastiera.nextInt();
totale += (voto3 * crediti3);
totaleCrediti += crediti3;
```

```
//Calcolo finale
double result = (double)totale/(double)totaleCrediti;
System.out.println("MEDIA PESATA: " + result);
```

## Esercizio 2 – Problemi

1. Il codice **viene ripetuto** praticamente uguale per tre volte e si usa una coppia di variabili diverse per ogni esame.
2. Il numero di esami che si possono inserire deve essere noto a priori.
3. In più, non controllo che il voto sia compreso tra 18 e 31.

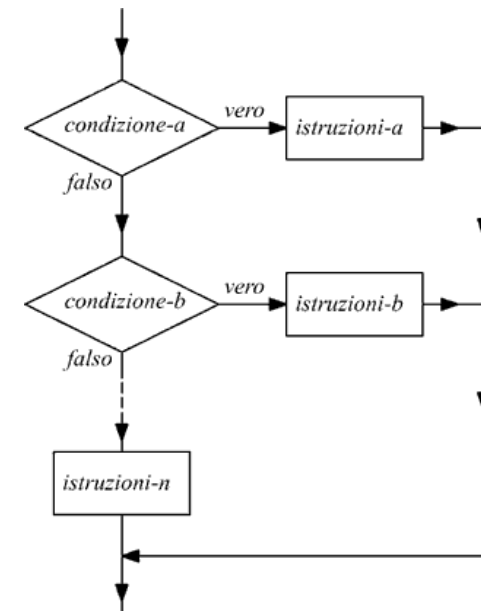
# Controllo del flusso

- Il codice che abbiamo scritto finora esprime una **sequenza** di operazioni, che vengono eseguite una dopo l'altra.
- Con la sola sequenza non possiamo **modellare il flusso** di esecuzione a nostro piacere, ma abbiamo bisogno di almeno altre due strutture di controllo:

➤ la **ripetizione**

➤ l'**alternativa** (scorsa esercitazione)

Java offre opportuni costrutti sintattici che permettono di esprimere queste strutture di controllo.



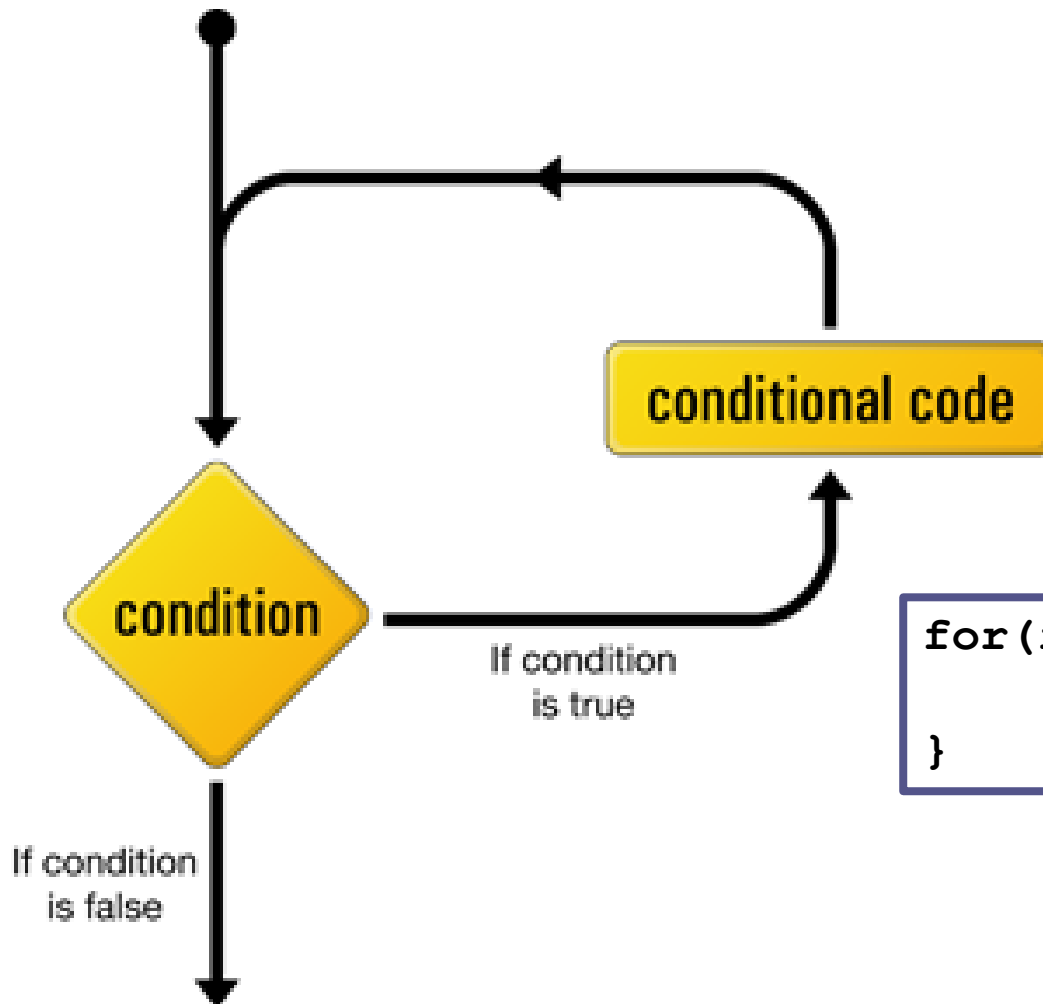
# Iterazione

Tre possibilità:

```
while( a > 5 ) {  
    ... ;  
}
```

```
do {  
    ... ;  
} while( a > 5 );
```

```
for(int i=0; i < 5; i++){  
    ... ;  
}
```



## Esercizio 3 – Media «risistemata»

- PROBLEMA 1: vogliamo inserire in un ciclo la parte di codice ripetuta e usare una sola coppia di variabili voto/crediti.
- La soluzione è semplice: usare un ciclo!

```
int voto = 0; int crediti = 0;
while ( /* CONDIZIONE */ ) {
    System.out.print("Inserisci il voto dell'esame : ");
    voto = tastiera.nextInt();
    System.out.print("Inserisci i crediti dell'esame : ");
    crediti = tastiera.nextInt();
    totale += (voto * crediti);
    totaleCrediti += crediti;
}
```

## Esercizio 3 – Media «risistemata»

- Qual è la condizione di terminazione del ciclo?
  - Il ciclo deve terminare quando l'utente ha inserito 3 esami

- Devo quindi definire un **indice** che assuma i valori 1,2,3:

```
int index = 1;
```

e che venga incrementato ad ogni iterazione.

- Dunque la condizione di terminazione sarà:

```
index <= 3
```

## Esercizio 3 – Media «risistemata»

- PROBLEMA 1: vogliamo inserire in un ciclo la parte di codice ripetuta e usare una sola coppia di variabili voto/crediti.
- La soluzione è semplice: usare un ciclo!

```
int voto = 0; int crediti = 0;
int index = 1;
while( index <= 3 ) {
    System.out.print("Inserisci il voto dell'esame : ");
    voto = tastiera.nextInt();
    System.out.print("Inserisci i crediti dell'esame : ");
    crediti = tastiera.nextInt();
    totale += (voto * crediti);
    totaleCrediti += crediti;
    index++;
}
```

## Esercizio 3 – Media «risistemata»

- PROBLEMA 2: se il numero di esami da inserire non fosse fissato a priori?

$$\text{media pesata} = \frac{\sum_{k=1}^N (\text{voto}_k * \text{crediti}_k)}{\sum_{k=1}^N \text{crediti}_k}$$

- Dobbiamo prima **chiedere all'utente** quanti voti vuole inserire.
- Useremo la variabile in cui abbiamo memorizzato quel valore per stabilire quante iterazioni fare.



## Esercizio 3 – Media «risistemata»

- PROBLEMA 2: se il numero di esami da inserire non fosse fissato a priori?
- La soluzione è semplice: chiederlo all'utente!

```
System.out.print("Quanti esami inserire? ");  
int N = tastiera.nextInt();
```

```
int index = 1;  
while ( index <= N ) {  
    ...  
    index++;  
}
```

## Esercizio 3 – Media «risistemata»

- PROBLEMA 3: nelle specifiche del problema, erano richiesti, come input:
    - voto (intero, compreso tra 18 e 31) [31 = 30L];
    - numero crediti (intero, maggiore di zero);
- a cui aggiungiamo:
- il numero di esami  $N$  deve essere maggiore di zero.
- 
- Come imporre la terminazione del programma se uno di questi valori non rispetta le specifiche?

## Esercizio 3 – Media «risistemata»

- L'istruzione `System.exit(0);` consente di terminare l'esecuzione del programma quando è insensato proseguire.

```
System.out.print("Quanti esami inserire? ");  
int N = tastiera.nextInt();
```

```
if ( N < 0 ) {  
    System.out.println("Valore non corretto!");  
    System.out.println("Numero esami deve essere positivo");  
    System.exit(0);  
}
```

*...continuo normalmente...*

- Il controllo va ripetuto ogni volta che si legge l'input.

## Esercizio 3 – Soluzione completa (1/3)

```
import java.util.Scanner;

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

        Scanner tastiera = new Scanner(System.in);
        int voto, crediti;

        //Variabili generali
        int totale = 0;
        int totaleCrediti = 0;

        //Leggo input e lo controllo
        System.out.print("Quanti esami inserire? ");
        int N = tastiera.nextInt();
        if(N < 0) {
            System.out.println("Valore non corretto!");
            System.out.println("Numero esami deve essere positivo");
            System.exit(0);
        }
    }
}
```

## Esercizio 3 – Soluzione completa (2/3)

```
int index = 1;
while( index <= N) {
    //Leggo il voto e controllo il valore
    System.out.print("Inserisci il voto dell'esame "+index+": ");
    voto = tastiera.nextInt();
    if(voto < 18 || voto > 31) {
        System.out.println("Valore non corretto!");
        System.out.println("Voto: tra 18 e 31");
        System.exit(0);
    }
    //Leggo i crediti e li controllo
    System.out.print("Inserisci i crediti dell'esame "+index+": ");
    crediti = tastiera.nextInt();
    if (crediti < 0) {
        System.out.println("Valore non corretto!");
        System.out.println("Crediti: maggiore di 0");
        System.exit(0);
    }
    totale += (voto * crediti);
    totaleCrediti += crediti;
    index++;
}
```

## Esercizio 3 – Soluzione completa (3/3)

```
//Calcolo finale  
  
double result = (double)totale/(double)totaleCrediti;  
System.out.println("MEDIA PESATA: "+result);  
}
```

## Esercizio 4 – Scarafaggi (1/2)

- Realizzare un programma che calcoli quanto impiega una popolazione di scarafaggi a riempire completamente una casa, dal pavimento al soffitto.
- Dati:
  - Tasso di crescita popolazione = 0.95 (ovvero, 95%);
  - Volume in  $m^3$  di uno scarafaggio =  $0.00000076 m^3$
- L'utente dovrà inserire:
  - Volume della casa in  $m^3$  (`double`);
  - Numero di scarafaggi presenti (`int`).
- Calcolare il numero di settimane (`int`) necessarie a riempire tutto il volume della casa.

## Esercizio 4 – Scarafaggi (2/2)

- A partire dalla popolazione iniziale, bisognerà...
  1. considerare 1 settimana = 1 iterazione
  2. verificare se lo spazio occupato dagli scarafaggi è maggiore, minore o uguale a quello della casa
  3. per ogni iterazione, calcolare quanti sono i nuovi scarafaggi (utilizzando il tasso di crescita in base alla popolazione corrente) e quanto spazio occupano.
  4. Tornare al passo 2.



# Esercizio 5 – Echo di stringhe

- Realizzare un programma che legga ciclicamente da input una stringa e la stampi a video
- Il programma deve terminare quando l'utente inserisce la stringa "0".

## Esercizio 6 – Logaritmo (1/3)

- Obiettivo: calcolo di un logaritmo in base qualunque.
- Realizzare un programma che chieda all'utente due numeri razionali (`double`), uno detto base  $B$  e l'altro detto valore  $X$  (a cui applicare il logaritmo).
- Se e solo se  $B$  e  $X$  sono entrambi positivi, il programma deve fornire come risposta il logaritmo in base  $B$  di  $X$ ; altrimenti deve stampare un messaggio di errore.

## Esercizio 6 – Logaritmo (2/3)

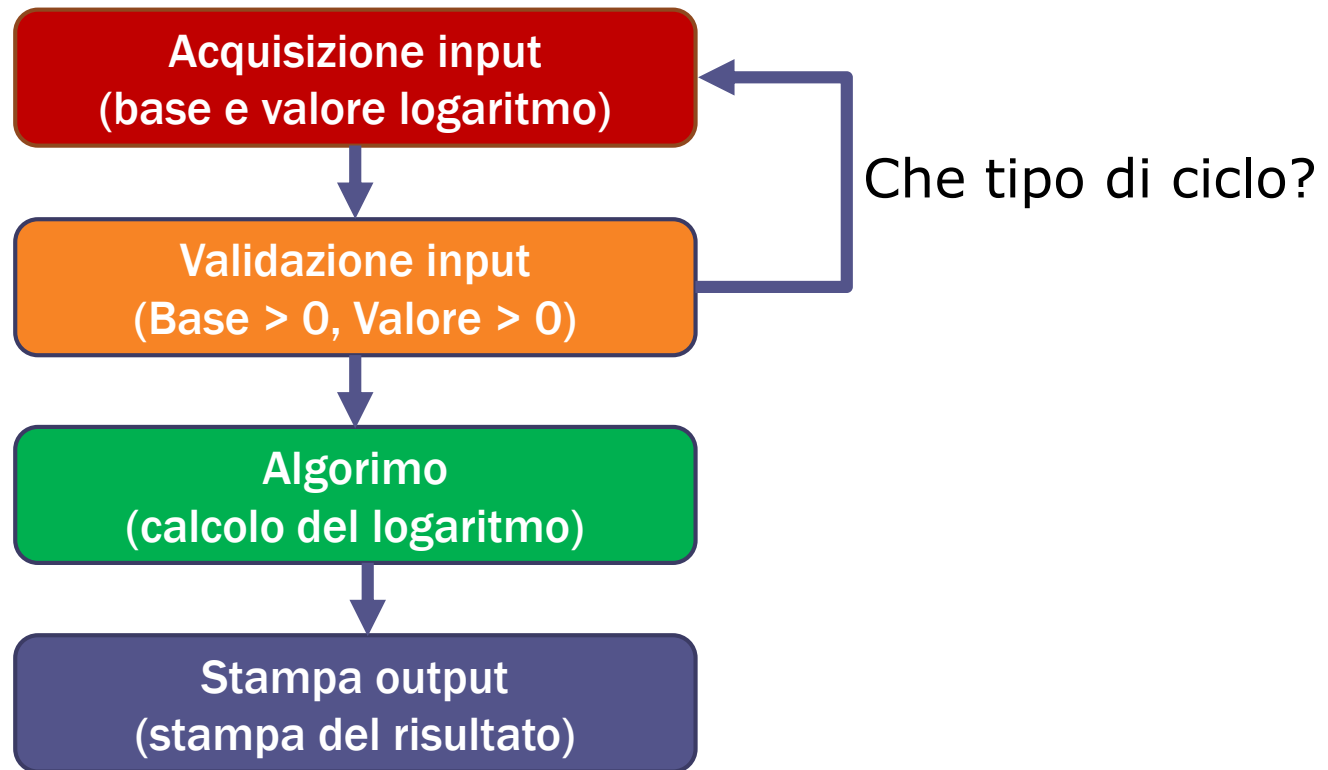
- Si rammenti che, data una funzione logaritmica in base  $k$ , un logaritmo in una base diversa  $B$  è così definito:

$$\log_b(x) = \frac{\log_k(x)}{\log_k(b)}$$

- Posso sfruttare del codice già fatto? Sì! Esiste la libreria *Math* che offre il calcolo del logaritmo naturale.
- Scrivere `Math.log(b)` restituisce un valore `double`, pari al logaritmo naturale di  $b$ .

# Esercizio 6 – Logaritmo (3/3)

- Schema di soluzione:



# Esercizio 7 – SommeProdotti

- Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi terminati dal valore 0 e stampi il prodotto degli interi positivi e la somma dei negativi.

- Ad esempio:

3  
-4  
6  
-2  
0

deve dare come output:

Somma positivi: 9

Prodotto negativi: 8

## Esercizio 8 – MediaPositivi

- Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi terminati dal valore 0 e stampi la media degli interi positivi.

- Ad esempio:

3

-4

6

-2

0

deve dare come output:

Media: 4.5

## Esercizio 9 – Sommatoria

- Si realizzi un programma che, partendo da una base  $a$  ed un limite  $n$ , calcoli la seguente funzione:

$$\sum_{i=0}^n a^i$$

- Realizzare il programma in due modi diversi:
  1. Utilizzando due cicli (uno per la sommatoria, ed uno per la potenza).
  2. Utilizzando un ciclo solo.

# Esercizio 10 – Sommatoria Doppia

- Dato un numero intero  $a$ , definito tramite una variabile, scrivere un programma che calcoli il valore della seguente espressione:

$$\sum_{i=1}^a \sum_{j=1}^i j$$



# Esercizio 11 – CalcoloInteressi (1/2)

- Si progetti un programma che legga un double, rappresentante un ammontare di euro; un tasso d'interesse (in percentuale), ed un numero di anni.
- Il programma deve stampare, per ogni anno, come l'ammontare cresce con gli interessi.
- Si ricordi che l'interesse si calcola con la seguente formula:

$$C_{fin} = C_{in} * \left(1 + \frac{r}{100}\right)^N$$

Dove  $C_{fin}$  è il capitale finale,  $C_{in}$  è quello iniziale,  $r$  è l'interesse, e  $N$  rappresenta il numero di anni in cui si applicano gli interessi.

# Esercizio 11 – CalcoloInteressi (2/2)

- Supponiamo che il capitale iniziale sia di 1000.0 €, con un tasso del 3%, per un periodo di 3 anni. L'output stampato deve avere all'incirca questo aspetto:

**Capitale iniziale: 1000.00€**

**Dopo 1 anno: 1030.00 €**

**Dopo 2 anni: 1060.90 €**

**Dopo 3 anni: 1092.73 €**

- Suggerimento: scomporre il problema in sotto-problemi più semplici!
  - All'inizio quali parametri è necessario chiedere all'utente?
  - Per ogni anno è necessario calcolare il capitale finale
  - Il capitale finale viene calcolato tramite la formula

# Altri esercizi facoltativi

- [http://docs.oracle.com/javase/tutorial/java/nutsandbolts/QuestionsandE/questions\\_flow.html](http://docs.oracle.com/javase/tutorial/java/nutsandbolts/QuestionsandE/questions_flow.html)
- <http://www.indiabix.com/java-programming/flow-control/>

Ma soprattutto vi invito a scrivere più codice possibile!

Rifate a casa gli esercizi fatti a lezione, anche con carta e penna: vi aiuterà moltissimo.