



# Metodi statici

Dichiarazione e chiamata di metodi statici

# Cos'è un metodo

- Un metodo è un blocco di istruzioni che fornisce una funzionalità e viene identificato con un nome
- Può avere dei parametri ed un valore di ritorno
- Esempi di metodi già visti:
  - `Math.pow(a, b);`
    - I parametri sono due `double` che rappresentano la base e l'esponente dell'operazione di elevamento a potenza. Ritorna un valore `double` contenente il risultato dell'operazione.
  - `System.out.println(str);`
    - Il parametro è una stringa che contiene il messaggio da stampare. Non ritorna nessun valore.
  - `Lettore.in.leggiInt();`
    - Metodo senza parametri. Il valore di ritorno è l'intero letto da tastiera.

# Dichiarazione di un metodo (1/2)

- La fase di dichiarazione serve per definire il metodo, specificando le istruzioni che deve eseguire.
- La dichiarazione va fatta una sola volta, dentro la classe ma fuori dagli altri metodi.
- ```
[modificatori] tipo_di_ritorno nome_del_metodo([parametri]){  
    corpo_del_metodo  
}
```

  - **modificatori**: parole chiave che modificano le caratteristiche del metodo
  - **tipo\_di\_ritorno**: tipo del valore restituito dal metodo (oppure **void** se il metodo non restituisce nessun valore)
  - **nome\_del\_metodo**: identificatore del metodo
  - **parametri**: lista di parametri separati da virgola. I parametri sono espressi specificandone il tipo ed il nome.
  - **corpo\_del\_metodo**: insieme di statement da eseguire alla chiamata del metodo
- L'istruzione `return` si utilizza per uscire dal metodo specificando il valore da ritornare. (Se la funzione non ha valore di ritorno può essere omessa)

# Dichiarazione di un metodo (2/2)

- Esempio: dichiarazione del metodo statico calcFatt che ha come parametro un intero e ritorna un long contenente il fattoriale del parametro.

- **public class** Fattoriale {

```
    public static long calcFatt(int n){
        long ris = 1; // conterrà il risultato
        for(int i=1; i<=n; i++){
            ris *= i;
        }
        return ris;
    }
```

...

```
}
```

- Per evitare errori di programmazione, è bene utilizzare l'istruzione return solo alla fine del corpo del metodo e non in mezzo al codice.

# Chiamata di un metodo

- La chiamata (o invocazione) di un metodo avviene quando si desidera utilizzare il metodo nel programma.

- **import** fiji.io.Lettore;

```
public class Fattoriale {
```

```
    /* Inizio dichiarazione */
```

```
    public static long calcFatt(int n){  
        long ris = 1; // conterrà il risultato  
        for(int i = 1; i <= n; i++){  
            ris *= i;  
        }  
        return ris;  
    }
```

```
    /* Fine dichiarazione */
```

```
    public static void main(String[] args){  
        int num = Lettore.in.leggiInt();  
        long fatt = calcFatt(num); // Chiamata  
        System.out.println("Il fattoriale di " + num +  
            " è " + fatt);  
    }
```

```
}
```

# Passaggio dei parametri

- Attenzione!  
Nel linguaggio java, il passaggio di parametri avviene per valore. Questo significa che i parametri nel metodo sono una copia degli argomenti della funzione che, quindi, non possono essere modificati.
- Esempio:

```
public static void nonScambia(int x, int y){  
    System.out.println("x: " + x + " y: " + y);  
    int tmp = x;  
    x = y;  
    y = tmp;  
    System.out.println("x: " + x + " y: " + y);  
}
```

```
public static void main(String[] args) {  
    int a = 10;  
    int b = 5;  
  
    System.out.println("a: " + a + " b: " + b);  
    nonScambia(a, b);  
    System.out.println("a: " + a + " b: " + b);  
}
```

Output:

```
a: 10 b: 5  
x: 10 y: 5  
x: 5 y: 10  
a: 10 b: 5
```

# Esercizio “StringheUguali”

Scrivere un metodo “uguali” con la seguente intestazione:

```
public static boolean uguali(String s, String t)
```

che confronta due stringhe e ritorna true se sono uguali o false altrimenti.

Successivamente, scrivere il metodo main che legge due stringhe da tastiera , le confronta usando il metodo “uguali” e stampa infine il risultato.

Non usare il metodo equals della classe string, ma effettuare il confronto carattere per carattere.

# Soluzione

```
import fiji.io.Lettore;
public class StringheUguali {

    public static boolean uguali(String s, String t){
        boolean eq = true;
        if(s.length() == t.length()){
            for(int i = 0; i < s.length() && eq; i++){
                if(s.charAt(i) != t.charAt(i)){
                    eq = false;
                }
            }
        }else{
            eq = false;
        }
        return eq;
    }

    public static void main(String[] args) {
        String a, b;
        System.out.println("Inserisci due stringhe:");
        a = Lettore.in.leggiString();
        b = Lettore.in.leggiString();
        if(uguali(a, b)){
            System.out.println("Le stringhe sono uguali");
        }else{
            System.out.println("Le stringhe sono diverse");
        }
    }
}
```



# Esercizio “Anagramma”

Scrivere un metodo con la seguente intestazione:

```
public static boolean controllaAnagram(String s, String t)
```

che ritorni true se le stringhe s e t sono una l'anagramma dell'altra, cioè se esiste una permutazione dei caratteri della stringa s che porta ad ottenere esattamente la stringa t.

L'esercizio è svolto introducendo un ulteriore metodo di appoggio con la seguente intestazione:

```
public static int contaOccorrenze (String u, char ch)
```

che restituisce il numero di occorrenze del carattere ch all'interno della stringa u fornita come primo parametro.

Infine scrivere il metodo main dove si utilizza il metodo anagramma su due stringhe lette da tastiera.

# Soluzione (1/3)

```
import fiji.io.Lettore;

public class Anagramma {

    public static int contaOccorrenze(String u, char ch){
        int conteggio = 0;
        for(int i = 0; i < u.length(); i++){
            if(u.charAt(i) == ch){
                conteggio++;
            }
        }
        return conteggio;
    }
}
```

# Soluzione (2/3)

```
public static boolean controllaAnagram(String s, String t){
    boolean anagram = true;

    if (s.length() != t.length()){
        anagram = false;
    }else{
        for(int i = 0; i < s.length() && anagram; i++){
            char l = s.charAt(i);
            int occoS, occoT;
            occoS = contaOccorrenze(s, l);
            occoT = contaOccorrenze(t, l);
            if(occoS != occoT){
                anagram = false;
            }
        }
    }
    return anagram;
}
```

# Soluzione (3/3)

```
public static void main(String[] args) {
    System.out.println("Inserisci due stringhe:");
    String s = Lettore.in.leggiString();
    String t = Lettore.in.leggiString();
    if(controllaAnagram(s, t)){
        System.out.println("Le stringhe sono una" +
            " l'anagramma dell'altra");
    }else{
        System.out.println("Le stringhe non sono una" +
            " l'anagramma dell'altra");
    }
}
}
```

# Esercizio “NumeroParole”

Scrivere un metodo “contaParole” con la seguente intestazione:

```
public static int contaParole(String str)
```

che ritorni il numero di parole presenti nella stringa data.

Una parola è identificata da una sequenza di caratteri consecutivi diversi dallo spazio (carattere ' ') e dal tab (carattere '\t').

Scrivere poi il metodo main che legge due stringhe da tastiera e controlla se hanno lo stesso numero di parole, utilizzando il metodo contaParole precedentemente definito.

Per leggere la stringa da tastiera usare il metodo `Letto.re.in.leggiLinea()`.

# Soluzione (1/2)

```
import fiji.io.Lettore;
public class NumeroParole {

    public static int contaParole(String str) {
        int numPar = 0; // Contatore delle parole trovate

        // Variabile che indica se si sta esaminando
        // una parola (true) o una sequenza di spazi (false)
        boolean parola = false;

        for(int i=0; i<str.length(); i++){
            char ch = str.charAt(i); // carattere corrente
            if(ch != ' ' && ch != '\t'){
                // carattere di una parola
                if(!parola){
                    parola = true; // inizio una nuova parola
                    numPar++;
                }
            }else{
                // carattere spazio
                parola = false;
            }
        }
        return numPar;
    }
}
```

# Soluzione (2/2)

```
public static void main(String[] args) {
    System.out.println("Inserisci due righe di testo");
    String str1 = Lettore.in.leggiLinea();
    String str2 = Lettore.in.leggiLinea();

    int n1 = contaParole(str1);
    int n2 = contaParole(str2);

    if(n1==n2){
        System.out.println("Le righe hanno lo stesso numero" +
            " di parole: " + n1);
    }else{
        System.out.println("Le righe hanno un numero diverso" +
            " di parole: " + n1 + " e " + n2);
    }
}
}
```