



# LABORATORIO DI PROGRAMMAZIONE

Corso di Laurea Ing.  
Gestionale 21/22

[Ing. Antonio Luca Alfeo](#)

[luca.alfeo@ing.unipi.com](mailto:luca.alfeo@ing.unipi.com)

# Classe java.lang.String (1/4)

- La classe String mette a disposizione dei metodi per effettuare operazioni su stringhe (sequenze di caratteri).
- Un oggetto di tipo String rappresenta una stringa di lunghezza arbitraria.
- Costruzione della stringa:

```
String s = new String ("parola"); // tramite costruttore  
String t = "parola"; // tramite assegnamento di un literal
```

# Classe java.lang.String (2/4)

- Alcuni metodi principali:

- `length()` ritorna la lunghezza (int) della stringa

```
int l = s.length(); // l == 6
```

- `charAt(int i)` ritorna il carattere (char) nella posizione i-esima (il primo carattere occupa la posizione 0)

```
char c = s.charAt(2); // c == 'r'
```

- `equals(String str)` confronta la stringa con un'altra stringa str e ritorna un valore boolean che vale true se le stringhe sono uguali o false altrimenti

```
boolean e = s.equals(t); // e == true
```

# Classe java.lang.String (3/4)

- Alcuni metodi utili:

- `indexOf(char c)` ritorna l'indice (int) della prima occorrenza del carattere c

```
int i = s.indexOf('a');           // i == 1
```

- `indexOf(char c, int i)` ritorna l'indice (int) della prima occorrenza del carattere c partendo dalla posizione i-esima

```
int i = s.indexOf('a', 2);       // i == 5
```

- `lastIndexOf(char c)` ritorna l'indice (int) dell'ultima occorrenza del carattere c

```
int i = s.lastIndexOf('a');     // i == 5
```

- `lastIndexOf(char c, int i)` ritorna l'indice (int) dell'ultima occorrenza del carattere c partendo dalla posizione i-esima e cercando all'indietro

```
int i = s.lastIndexOf('a', 4);  // i == 1
```

# Classe java.lang.String (4/4)

- Alcuni metodi utili:

- `substring(int i)` ritorna la sotto-stringa partendo dalla posizione i-esima

```
String ss = s.substring(3);           // "ola"
```

- `substring(int i, int j)` ritorna la sotto-stringa partendo dalla posizione i-esima fino alla posizione (j-1)-esima.

```
String ss = s.substring(2, 4);       // "ro"
```

# Concatenazione

- L'operatore "+" applicato a due stringhe effettua la concatenazione. Ritorna quindi una terza stringa ottenuta unendo le due stringhe.

```
String a = "ci";  
String b = "ao";  
String c = a + b;           // "ciao"
```

- Se soltanto uno degli operandi dell'operatore "+" è una stringa, l'altro viene convertito in stringa.

```
String a = "b vale ";  
int b = 10;  
String c = a + b;           // "b vale 10"
```

- Attenzione alle precedenze degli operatori!

```
String a = "b + c vale ";  
int b = 10;  
int c = 5;  
String d = a + b + c;       // "b + c vale 105"  
String e = a + (b + c);     // "b + c vale 15"
```

# Uguaglianza tra stringhe

- **Non utilizzare l'operatore di confronto ==** per verificare l'uguaglianza tra due stringhe, perché potrebbe dare risultati inaspettati.
- Utilizzare sempre il metodo **equals()**

```
String a = "ciao";
String b = "ciao";
String c = new String("ciao");

if(a==b){ // true
    System.out.println("a==b");
}

if(a==c){ // false
    System.out.println("a==c");
}

if(a.equals(b)){ // true
    System.out.println("a equals b");
}

if(a.equals(c)){ // true
    System.out.println("a equals c");
}
```

# Input di stringhe (1/2)

- La classe lettore della **libreria fiji** mette a disposizione i seguenti metodi per inserire stringhe:
- **LeggiString()**
  - Legge una stringa di caratteri. La stringa non deve contenere spazi.
- **LeggiLinea()**
  - Legge un'intera linea di testo fino alla pressione del tasto invio. Può contenere spazi.

# Input di stringhe (2/2)

- La classe lettore grafico della libreria unipi mette a disposizione il seguente metodo per inserire stringhe:
- **LeggiTesto(messaggio)**
  - Legge un'intera linea di testo fino alla pressione del tasto invio. Può contenere spazi.

# Esempio “StringaNumerica”

Il programma legge una stringa da tastiera e controlla se il testo inserito è composto solo da numeri o se sono presenti anche delle lettere. Stampa a video il risultato, “È un numero” oppure “Non è un numero”.

# Soluzione (1/2)

```
import fiji.io.Lettore;

public class StringaNumerica {

    public static void main(String[] args) {

        System.out.println("Inserisci la stringa");
        String txt = Lettore.in.leggiString();

        int indice = 0;           // indice del carattere in esame
        boolean numero = true;   // contiene la risposta

        // se il primo carattere è il segno, va tutto bene e
        // parto dal secondo valore.
        if(txt.charAt(0) == '+' || txt.charAt(0) == '-'){
            indice++;
        }
    }
}
```

# Soluzione (2/2)

```
// scorro la stringa controllando la validità
// di ogni carattere
for(; indice < txt.length() && numero; indice++){
    char c = txt.charAt(indice);
    // I caratteri '0', '1', ... '9' sono ordinati,
    // quindi per controllare che il carattere c non sia
    // una cifra si può usare (c<'0' || c>'9')
    if(c<'0' || c>'9'){
        numero = false;
    }
}

if(numero){
    System.out.println("È un numero");
}else{
    System.out.println("Non è un numero");
}
}
}
```

# Esercizio “ConversioneData”

Il programma legge una stringa da tastiera interpretandola come una data nel formato GG/MM/AAAA, ne estrae i vari campi (giorno, mese ed anno) convertendoli in formato numerico ed infine stampa a video i risultati.

Il mese viene stampato anche in formato testuale:  
1 (Gennaio) - 2 (Febbraio) - 3 (Marzo) - 4 (Aprile)  
5 (Maggio) - 6 (Giugno) - 7 (Luglio) - 8 (Agosto)  
9 (Settembre) - 10 (Ottobre) - 11 (Novembre)  
12 (Dicembre)

# Soluzione (1/4)

```
import unipi.io.*;

public class ConversioneData {

    public static void main(String[] args) {

        // Input della Data nel formato GG/MM/AAAA
        String data = LettoreGr.in.leggiTesto(
            "Inserisci la data nel formato GG/MM/AAAA");

        // Conversione Giorno
        int gg_idx = data.indexOf('/');
        String gg_str = data.substring(0, gg_idx);
        int gg = Integer.parseInt(gg_str);

        // Conversione Mese
        int mm_idx = data.lastIndexOf('/');
        String mm_str = data.substring(gg_idx + 1, mm_idx);
        int mm = Integer.parseInt(mm_str);
```

# Soluzione (2/4)

```
// Conversione Mese Testuale
String mm_c_str;
switch (mm) {
case 1:
    mm_c_str = "Gennaio";
    break;
    case 2:
    mm_c_str = "Febbraio";
    break;
    case 3:
    mm_c_str = "Marzo";
    break;
case 4:
    mm_c_str = "Aprile";
    break;
    case 5:
    mm_c_str = "Maggio";
    break;
    case 6:
    mm_c_str = "Giugno";
    break;
```

# Soluzione (3/4)

```
case 7:
    mm_c_str = "Luglio";
    break;
    case 8:
    mm_c_str = "Agosto";
    break;
    case 9:
    mm_c_str = "Settembre";
    break;
case 10:
    mm_c_str = "Ottobre";
    break;
    case 11:
    mm_c_str = "Novembre";
    break;
    case 12:
    mm_c_str = "Dicembre";
    break;
    default:
    mm_c_str = "caso non previsto";
}
```

# Soluzione (4/4)

```
// Conversione Anno
String aa_str = data.substring(mm_idx + 1);
int aa = Integer.parseInt(aa_str);

// Output dei dati estratti e convertiti.
ScrittoreGr.out.scriviTesto(
    "Giorno: " + gg);
ScrittoreGr.out.scriviTesto(
    "Mese: " + mm + " (" + mm_c_str + ")");
ScrittoreGr.out.scriviTesto(
    "Anno: " + aa);
}
}
```

# Esercizio “Uguali”

Scrivere un programma che legge due stringhe di uguale lunghezza da tastiera, le confronta carattere per carattere e mostra a video l'indice del primo carattere diverso, oppure l'indicazione che sono uguali.

Nel caso le due stringhe siano di lunghezza diversa il programma deve indicarlo e terminare.

# Soluzione (1/2)

```
import fiji.io.Lettore;

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

        System.out.println("Scrivi due righe di testo");
        String str1 = Lettore.in.leggiLinea();
        String str2 = Lettore.in.leggiLinea();

        if(str1.length() != str2.length()){
            System.out.println("Lunghezza diversa, "
                + str1.length() + " e " + str2.length());
        }
    }
}
```

# Soluzione (2/2)

```
else{
    boolean uguali = true;
    int i;
    for(i = 0; i < str1.length() && uguali ; i++){
        if(str1.charAt(i) != str2.charAt(i)){
            uguali = false;
        }
    }

    if(uguali){
        System.out.println("Le stringhe sono uguali");
    }else{
        System.out.println("L'indice del primo " +
            "carattere diverso è " + (i - 1));
    }
}
}
}
```

# Esercizio “Palindromo”

Scrivere un programma che legge una stringa da tastiera e indica se la stringa inserita è palindroma, cioè se contiene la stessa sequenza di caratteri leggendola da destra e da sinistra.

## Metodi:

- Usando degli indici
- Creazione di una copia invertita della stringa

## Esempi di stringhe palindrome:

- anna
- itopinonavevanonipoti

# Metodo degli indici

```
import fiji.io.Lettore;

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

        System.out.println("Scrivi la stringa");
        String str = Lettore.in.leggiLinea();

        boolean palindromo = true; // indica se la stringa è palindroma
        int idxFw = 0; // indice per scorrere in avanti
        int idxBw = str.length()-1; // indice per scorrere indietro
        int estremo = str.length()/2; // numero confronti

        while(idxFw<estremo && palindromo){
            palindromo = str.charAt(idxFw) == str.charAt(idxBw);
            idxFw++;
            idxBw--;
        }

        if(palindromo){
            System.out.println("La stringa è palindroma");
        }else{
            System.out.println("La stringa non è palindroma");
        }
    }
}
```

# Metodo degli indici - Variante

```
import fiji.io.Lettore;

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

        System.out.println("Scrivi la stringa");
        String str = Lettore.in.leggiLinea();

        boolean palindromo = true;
        for(int i=0; i<str.length()/2 && palindromo; i++){
            if(str.charAt(i) != str.charAt(str.length()-1-i)){
                palindromo = false;
            }
        }

        if(palindromo){
            System.out.println("La stringa è palindroma");
        }else{
            System.out.println("La stringa non è palindroma");
        }
    }
}
```

# Metodo della copia invertita

```
import fiji.io.Lettore;

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

        System.out.println("Scrivi la stringa");
        String str = Lettore.in.leggiLinea();

        String tmpStr = ""; // stringa che conterrà la copia invertita

        for (int i = 0; i < str.length(); i++) {
            tmpStr = str.charAt(i) + tmpStr;
        }

        if(str.equals(tmpStr)){
            System.out.println("La stringa è palindroma");
        }else{
            System.out.println("La stringa non è palindroma");
        }
    }
}
```

# Esercizio “Prefisso”

Scrivere un programma che legge due stringhe da tastiera (s e t) e dice se la stringa s inizia con la stringa t, cioè se t è un prefisso di s.

# Soluzione

```
import fiji.io.Lettore;

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

        System.out.println("Scrivi due stringhe");
        String s = Lettore.in.leggiLinea();
        String t = Lettore.in.leggiLinea();

        boolean prefix = true;
        if(s.length() >= t.length()){
            for (int i = 0; i < t.length() && prefix; i++){
                if(s.charAt(i) != t.charAt(i)){
                    prefix = false;
                }
            }
        }else{
            prefix = false;
        }

        if(prefix){
            System.out.println("La seconda stringa è un prefisso");
        }else{
            System.out.println("La seconda stringa non è un prefisso");
        }
    }
}
```