

Fondamenti di Informatica T1

Stringhe

Tutor

Melissa Licciardello

melissa.licciardell2@unibo.it

Stringhe

- **Stringa** : tipo di dato che rappresenta un insieme ordinato di caratteri
- Creazione di una stringa in Java
 - ▶ `String mia_stringa = "Ciao";`
- Lettura di una stringa da tastiera
 - ▶ `String mia_stringa = scanner.next();` → Lettura di ciò che viene immesso da console **FINO AL PRIMO SPAZIO BIANCO**
 - ▶ `String mia_stringa = scanner.nextLine();` → Lettura di una intera linea a console (quindi anche di spazi bianchi)

Caso di studio: `nextLine()` preceduta da `nextInt()` (1)

- Mettiamo il caso di voler leggere da input un intero e poi una riga.
- Codice :

```
int numero = scanner.nextInt();  
String frase = scanner.nextLine();
```

- Se l'input è di questo tipo

42 un po' di parole

non si verifica alcun problema : a *numero* viene assegnato 42 e a *frase* la stringa "un po' di parole"

- Diverso è invece il caso di un input di questo tipo

42

un po' di parole

- **PROBLEMA:** A *numero* viene assegnato 42, ma a *frase* viene assegnata **UNA STRINGA VUOTA**

Caso di studio: `nextLine()` preceduta da `nextInt()` (2)

- La causa di tutto è il carattere speciale `"\n"` (Altrimenti detto "a capo")
 - ▶ Il nostro input riconosce 42 come numero ma, purtroppo, riconosce il nostro `"\n"` come un carattere da salvare (ma, allo stesso tempo, lo riconosce come un carattere di terminazione)
 - ▶ Non viene addirittura richiesto l'inserimento della stringa successiva (in caso invece di *"cut&paste"* viene completamente ignorata)
- **SOLUZIONE** : inserimento di una `nextLine()` in più, con nessuna necessità di salvataggio (non abbiamo bisogno di memorizzare una riga vuota):

```
int numero = scanner.nextInt();
            nextLine();
String frase = scanner.nextLine();
```

Ricordarsi però di separare l'input con l'invio e NON con uno spazio!

Stringhe : operazioni (1)

- Prendiamo come esempio la seguente stringa
 - ▶ *String messaggio = "Ciao!"*;
- **Concatenazione** (dove l'abbiamo già vista?)
 - ▶ *messaggio = messaggio + "Come stai?"*;
- **length()** : Calcolo della lunghezza
 - ▶ *int lunghezza_messaggio = messaggio.length()*;
- **indexOf()** : Indice della prima occorrenza di una data sottostringa
 - ▶ *int indice_prima_occorrenza = messaggio.indexOf("st")*;
- **lastIndexOf()** : Indice dell'ultima occorrenza di una data sottostringa
 - ▶ *int indice_ultima_occorrenza = messaggio.lastIndexOf("st")*;

Stringhe : operazioni (2)

- **charAt()**: Estrazione dell' i-esimo carattere. Nell'esempio → primo carattere
 - ▶ *char prima_lettera = messaggio.charAt(0);*
NB: l'indice del primo carattere è 0, non 1!
- **substring()**: Estrazione di una sottostringa dato un range di indici [indice_iniziale-indice_finale]
 - ▶ *String sottostringa = messaggio.substring(indice_iniziale, indice_finale);*
- **substring()**: Estrazione di una sottostringa dato un indice iniziale
 - ▶ *String sottostringa = messaggio.substring(indice_iniziale);*
- **toUpperCase()** - **toLowerCase()**: Trasformazione maiuscole-minuscole
 - ▶ *String messaggio_maiuscolo = messaggio.toUpperCase();*
 - ▶ *String messaggio_minuscolo = messaggio.toLowerCase();*

Stringhe : operazioni (3)

- **replaceAll()**: Sostituzione di una sottostringa ad ogni occorrenza
 - ▶ *String nuovo_messaggio = messaggio.replaceAll(old,new);*
Utilissima per eliminare gli spazi bianchi. Come? :)
- **compareTo()**: Comparazione di due stringhe
 - ▶ *int ordine = messaggio.compareTo(altro_messaggio);*
- **equals()** - **equalsIgnoreCase()** : Funzioni di uguaglianza
 - ▶ *boolean uguaglianza = messaggio.equals(altro_messaggio);*
Viene controllata l'esatta sequenza di caratteri
 - ▶ *boolean uguaglianza = messaggio.equalsIgnoreCase(altro_messaggio);*
Vengono ignorati spazi, tab e differenze nelle lettere maiuscole/minuscole

Esercizio 1 (Svolto)

- Scrivere un programma che, inserite due frasi da tastiera, stampi a video una stringa composta nel seguente modo:
 - ▶ I primi dieci caratteri devono essere i primi dieci caratteri della prima frase. Le lettere devono essere tutte maiuscole e agli spazi devono essere sostituiti dei trattini.
 - ▶ Gli ultimi dieci caratteri devono essere gli ultimi dieci caratteri della seconda frase. Le lettere devono essere tutte minuscole e agli spazi devono essere sostituite delle chioccioline.
 - ▶ Le due parti della stringa devono essere separate da un underscore.
- **Problema** : non stiamo facendo alcun tipo di controllo sull'input (impareremo a farlo). Dobbiamo quindi garantire "da utilizzatori" di star inserendo almeno dieci caratteri per ogni frase.

Soluzione primo esercizio

```
package mainpackage;
import java.util.Scanner;
public class EsercizioUnoMain
{
    public static void main(String[] args)
    {
        //Creo la mia istanza di Scanner
        Scanner scanner = new Scanner(System.in);

        //Stampo a video la richiesta di inserire la prima frase
        System.out.println("Inserire la prima frase");
        String prima_frase = scanner.nextLine();

        //Stampo a video la richiesta di inserire la seconda frase
        System.out.println("Inserire la seconda frase");
        String seconda_frase = scanner.nextLine();

        //PRIMA FRASE : metto tutto in lettere maiuscole e sostituisco gli spazi con dei trattini
        prima_frase = prima_frase.toUpperCase();
        prima_frase = prima_frase.replaceAll(" ", "-");

        //SECONDA FRASE : metto tutto in lettere minuscole e sostituisco gli spazi con delle chiocciole
        seconda_frase = seconda_frase.toLowerCase();
        seconda_frase = seconda_frase.replaceAll(" ", "@");

        //Prendo le prime dieci lettere della prima frase
        prima_frase = prima_frase.substring(0, 10);

        //Prendo le ultime dieci lettere della seconda frase e concateno
        seconda_frase = seconda_frase.substring(seconda_frase.length() - 10);
        String frase_finale = prima_frase + "_" + seconda_frase;

        //Stampo a video
        System.out.println("La frase finale è : " + frase_finale);

        //Chiudo lo scanner
        scanner.close();
    }
}
```

1

Esercizio 2

- Scrivere un programma che, inseriti da tastiera nome e cognome (esempio: Pinco Pallino), stampi a video le tre possibili email fornite dall'Unibo:
 - ▶ Email da studente → `pinco.pallino@studio.unibo.it`
 - ▶ Email da professore → `pinco.pallino@unibo.it`
 - ▶ Email da tutor didattico → `pinco.pallin2@unibo.it`
- Nota bene
 - ▶ Le email NON contengono lettere maiuscole
 - ▶ Le email NON contengono spazi bianchi (quindi, in caso di nomi e cognomi composti, vorremmo essere in grado di eliminarli.. come?)
Es : Nome = Maria Grazia
Cognome = De Giovannini
→ `mariagrazia.degiovannini@unibo.it`
 - ▶ L'email da tutor tronca l'ultima lettera e, prima della chiocciola, ha sempre un "2". Tips : utilizzo di `substring()`;

Esercizio 3 (1)

- Un negozio di CD utilizza un sistema di catalogazione in base a dei codici strutturati nel seguente modo:
 - ▶ Le prime tre lettere, in maiuscolo, sono le prime tre lettere del nome dell'artista.
 - ▶ Le successive tre lettere, separate da un trattino dalle precedenti, sono le ultime tre lettere del nome dell'album, sempre in maiuscolo.
 - ▶ L'anno di produzione del disco, separato dal pezzo precedente da un trattino.
 - ▶ Un numero casuale da 0 a 100, separato dal pezzo precedente con un carattere di dollaro.
- Scrivere quindi un programma che, inseriti da input il nome dell'artista, il nome dell'album e l'anno di produzione stampi a video il corrispondente codice.

Esercizio 3 (2)

- Nota bene:

- ▶ Le lettere del nome dell'artista e dell'album che contano al fine di generare il codice NON devono contenere spazi. Dovremo eliminarli prima di estrarre i tre caratteri utili :)
- ▶ Per la generazione del numero casuale utilizzare il seguente codice:

```
int numero_casuale = (int) (Math.random() * 100);
```

- Esempio

- ▶ Artista : A Day to Remember
- ▶ Album : For Those Who Have Heart
- ▶ Anno : 2007
- ▶ Codice generato : ADA-ART-2007\$77 ← Il 77 è un numero *casuale*