



Programmazione Object Oriented

Introduzione

Dr. Dario Di Nucci - 23/09/2022

Obiettivi ed ambiente di lavoro

Imparare facendo

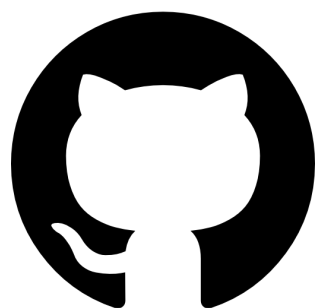
Investiremo 24 ore implementando sistemi software in Java.

Imparare tramite esempi

Vi mostrerò Java in pratica partendo dalle basi fino ai concetti più avanzati.



JUnit



Calendario del laboratorio

In ogni laboratorio completeremo degli esercizi relativi alle diverse caratteristiche del linguaggio di Object Oriented Programming e Java analizzate a lezione.

	Martedì (ore: 9-11 Aula: F/1)	Giovedì (ore: 9-11 Aula: F/1)	Venerdì (ore: 9-13 aula: P/13)
Settimana 1 (20-23 Settembre)	Introduzione al corso Programmazione OO	Utilizzare gli Oggetti	Lab: Introduzione Lab: Scrittura primi programmi in Java
Settimana 2 (27-30 Settembre)	Utilizzare le Classi	Realizzare Classi	Lab: Creare classi ed oggetti
Settimana 3 (4-6 Ottobre)	Tipi di Dati Fondamentali	Decisioni/Iterazioni	
Settimana 4 (11-14 Ottobre)	Vettori e Array	Progettazione di Classi	Lab: Creare classi
Settimana 5 (18-21 Ottobre)	Progettazione di Classi	Interfacce e Polimorfismo	Lab: Classi con array
Settimana 6 (25-27 Ottobre)	Interfacce e Polimorfismo	Ereditarietà	Lab: Packages e variabili statiche
Settimana 7 (3-4 Novembre)		Ereditarietà	Lab: Interfacce
Settimana 8 (8-11 Novembre)	Testing e Debugging	Collezioni Programmazione Generica	Lab: Interfacce
Settimana 9 (15-18 Novembre)	Gestione delle Eccezioni	IO e Flussi	Lab: Ereditarietà
Settimana 10 (22-25 Novembre)	Programmazione Grafica	Gestione degli Eventi	Lab: Eccezioni
Settimana 11 (29 Nov - 2 Dic)	Interfacce Grafiche Utente	Lambda Expressions	Lab: Esercizi su eventi e interfacce grafiche
Settimana 12 (6-9 Dicembre)	Reflections		Lab: Esercizi su eventi e interfacce grafiche
Settimana 13 (13-16 Dicembre)	Reflections	Simulazione prova	Lab: Esercitazione finale

Cosa mi aspetto da voi?

- Svolgere gli esercizi del laboratorio.
- Studiare lezione per lezione ed applicare i concetti teorici man mano che vengono introdotti.
- Avete dubbi?

Contattatemi via mail

Fermatemi alla fine della
lezione

Venite a ricevimento

Come contattarmi?

Riceverete le comunicazioni ufficiali via e-learning platform.

Potete contattarmi via mail: ddinucci+oop@unisa.it

Ricevimento

Giovedì	15:00 - 18:00	Stanza L11. Ex sala del consiglio del Dipartimento di Informatica, vicino al laboratorio Alan Turing.
	In alternativa al ricevimento in presenza, si prega di scrivere al docente per fissare un appuntamento su MS Teams.	



È l'ora del PadLet!



Break

Cosa impariamo oggi?

Come configurare il nostro ambiente di sviluppo integrato?

Come utilizzare oggetti di base in Java?

Cosa ci serve?

- L'ultima versione di IntelliJ IDEA:

<https://www.jetbrains.com/idea/>

La community edition va bene. Nel caso vogliate la versione completa, dovete richiedere una licenza studente.

<https://www.jetbrains.com/community/education/#students>

La licenza è gratuita (basta utilizzare la mail unisa).

- Il plugin EduTools di IntelliJ (lo trovate nel marketplace)

<https://plugins.jetbrains.com/plugin/10081-edutools>

- Oracle JDK 17

<https://www.oracle.com/java/technologies/downloads/#java17>

- La documentazione di JDK è scaricabile qui

<https://www.oracle.com/java/technologies/javase-jdk17-doc-downloads.html>

oppure disponibile online qui

<https://docs.oracle.com/en/java/javase/17/>

La classe String

Una stringa è una sequenza finita di caratteri

In Java non esiste il tipo primitivo stringa ma esiste una classe predefinita: **le stringhe sono oggetti!**

Le operazioni su stringhe avvengono tramite metodi della classe `String`.

In Java una stringa è una sequenza finita di caratteri racchiusa tra virgolette, come "buongiorno" (le virgolette non fanno parte della stringa).

Per dichiarare una variabile stringa

```
String vuota = "";           // stringa con valore nullo
String nome = "Luigi";       // stringa con valore Luigi
```

Il metodo length

Una stringa vuota è un oggetto `String` che rappresenta una sequenza vuota di caratteri, cioè una stringa di lunghezza zero. La stringa vuota è denotata dal letterale `""`.

Il metodo `int length()` della classe `String` calcola la lunghezza della stringa, cioè il numero di caratteri che compongono la stringa.

Ad esempio

```
vuota.length();           // vale 0
nome.length();            // vale 5: nome è "Luigi"
"buongiorno".length();    // vale 10
```

Il metodo concat

La concatenazione di stringhe, a partire da due stringhe, permette di ottenere una stringa il cui valore è dato dalla sequenza di caratteri della prima stringa seguita dalla sequenza di caratteri della seconda stringa.

Il metodo **`String concat(String x)`** della classe `String` implementa la concatenazione di stringhe.

`concat` crea e restituisce un nuovo oggetto `String` composto dai caratteri della stringa su cui il metodo viene invocato seguiti dai caratteri della stringa argomento `x`.

Ad esempio

```
String s1="barba", s2="gianni", s3;  
s3 = s1.concat(s2);           // s3 vale "barbagianni"
```

L'operatore +

La concatenazione di stringhe si può effettuare anche tramite l'operatore `+` (inteso come concatenazione).

Ad esempio

```
System.out.println(s1 + s2); // stampa "barbagianni"  
System.out.println("la " + s1 + " di " + s2);
```

L'operatore `+` è sovraccarico (i.e., *overloaded*) perché opera su numeri e stringhe.

Questo è un esempio di overloading predefinito.

Altri metodi della classe String

Metodo	Restituisce	Parametri
<code>toUpperCase</code>	Oggetto String	Nessuno
<code>toLowerCase</code>	Oggetto String	Nessuno
<code>Length</code>	Intero	Nessuno
<code>Trim</code>	Oggetto String	Nessuno
<code>concat</code>	Oggetto String	Un oggetto String
<code>Substring</code>	Oggetto String	Un numero intero
<code>Substring</code>	Oggetto String	Due numeri interi

Per una panoramica sugli altri metodi consultare

<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html>

Posizioni nelle stringhe

- Le posizioni dei caratteri in una stringa sono numerate a partire da 0.

H a m b u r g e r
0 1 2 3 4 5 6 7 8

- Stringhe e sotto-stringhe

```
String hamburger = "hamburger";  
String short_hamburger = hamburger.substring(3);  
String shorter_hamburger = hamburger.substring(3, 7);  
String uppercase_hamburger = hamburger.toUpperCase();  
String my_hamburger = hamburger.concat(" with chips");
```


Esempio

```
public class Esempio_substring_concat {  
    public static void main(String arg[]) {  
        String first = "John";  
        String middle = "Fitzgerald";  
        String last = "Kennedy";  
        String initials;  
        String firstInit, middleInit, lastInit;  
        firstInit = first.substring(0,1);  
        middleInit = middle.substring(0,1);  
        lastInit = last.substring(0,1);  
        initials = firstInit.concat(middleInit);  
        initials = initials.concat(lastInit);  
        System.out.println(initials);  
    }  
}
```



Esercizi

Scrivere un programma che stampi il carattere centrale di una stringa.

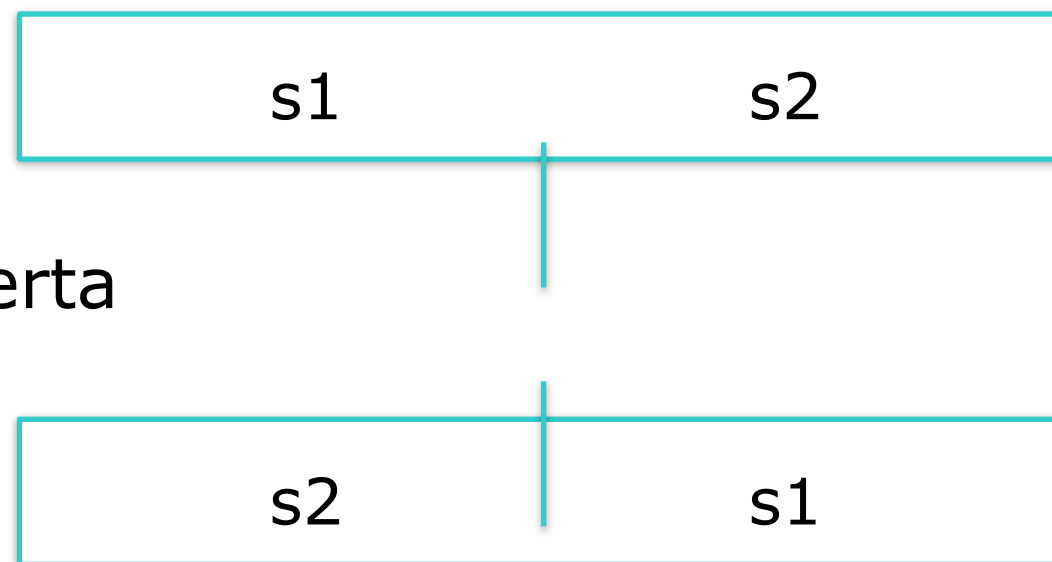
Scrivere un programma che data una stringa di almeno 2 caratteri, ne costruisca un'altra dove il primo carattere è scambiato con l'ultimo, che viene poi stampata a video.

Esercizio

Scrivere un programma che data una stringa di almeno 2 caratteri,

divida la stringa in due parti della stessa dimensione

e poi le inverta



Astrazione sui dati in input

Un modo per dare in input al programma dei valori a tempo di esecuzione è utilizzare l'array passato come parametro al `main`.

Possiamo accedere a questi valori con `args[0]`, `args[1]`, etc.

Gli elementi di un array monodimensionale sono individuati tramite un indice che parte da 0.

Ricordatevi che gli elementi dell'array sono oggetti `String`.



Esercizio

Scrivere un programma che data una stringa inserita a linea di comando restituisce il carattere centrale.

Scrivere un programma che date tre stringhe inserite a linea di comando visualizza la concatenazione delle loro iniziali.

Il metodo `charAt`

Il metodo `char charAt(int)` restituisce il carattere contenuto in una precisa posizione della stringa.

È necessario passare il numero di posizione come argomento della funzione.

La prima posizione ha come numero 0, la seconda 1 e così via.

Riscrivere gli esercizi della precedente slide utilizzando questo metodo.

Il metodo indexOf

Il metodo `int indexOf(String)` restituisce la posizione della prima occorrenza di una data sottostringa in una stringa.

È necessario passare come argomento della funzione la sottostringa che si vuole ricercare.

Quando la ricerca ha esito negativo, la funzione restituisce -1.

Es.

```
"corso-prog".indexOf("ro");
```


Il metodo indexOf

Una seconda versione del metodo effettua la ricerca a partire da una fissata posizione della stringa

È necessario passare come argomenti del metodo

- La sottostringa che si vuole ricercare;
- La posizione a partire da cui effettuare la ricerca.

Es.

```
"corso-prog".indexOf("r", 5);
```



Esercizio

- Scrivere un programma che data una stringa `s` controlla se il primo carattere di `s` è ripetuto nella stringa stampando la posizione in cui è ripetuto o `-1`.



Esercizio

Scrivere un programma che data una stringa scritta tutta in caratteri minuscoli, trasformi il primo carattere in maiuscolo e assegni la stringa risultante ad una variabile di tipo `String`.

Esercizio

Scrivere un programma che scambi tra loro le lettere "e" ed "o" in una stringa usando ripetutamente il metodo `replace`.

Per fare questo esercizio, avete bisogno della classe `StringBuilder`.

Mostrare che la stringa "Hello, World!" si trasforma in "Holle, Werld!"

Esercizio

Scrivere un programma che istanzi un oggetto `Rectangle` e ne stampi le caratteristiche.

Modificare il programma in modo che sposti il rettangolo stampato di `newX` e `newY` ed aggiunga 3 rettangoli in orizzontale così che se i 4 rettangoli stampati fossero disegnati, formerebbero un unico grande rettangolo.

Infine, stamp le caratteristiche del rettangolo finale con area e perimetro.

Cosa facciamo la prossima volta?

	Martedì (ore: 9-11 Aula: F/1)	Giovedì (ore: 9-11 Aula: F/1)	Venerdì (ore: 9-13 aula: P/13)
Settimana 1 (20-23 Settembre)	Introduzione al corso Programmazione OO	Utilizzare gli Oggetti	Lab: Introduzione Lab: Scrittura primi programmi in Java
Settimana 2 (27-30 Settembre)	Utilizzare le Classi	Realizzare Classi	Lab: Creare classi ed oggetti
Settimana 3 (4-6 Ottobre)	Tipi di Dati Fondamentali	Decisioni/Iterazioni	
Settimana 4 (11-14 Ottobre)	Vettori e Array	Progettazione di Classi	Lab: Creare classi
Settimana 5 (18-21 Ottobre)	Progettazione di Classi	Interfacce e Polimorfismo	Lab: Classi con array
Settimana 6 (25-27 Ottobre)	Interfacce e Polimorfismo	Ereditarietà	Lab: Packages e variabili statiche
Settimana 7 (3-4 Novembre)		Ereditarietà	Lab: Interfacce
Settimana 8 (8-11 Novembre)	Testing e Debugging	Collezioni Programmazione Generica	Lab: Interfacce
Settimana 9 (15-18 Novembre)	Gestione delle Eccezioni	IO e Flussi	Lab: Ereditarietà
Settimana 10 (22-25 Novembre)	Programmazione Grafica	Gestione degli Eventi	Lab: Eccezioni
Settimana 11 (29 Nov - 2 Dic)	Interfacce Grafiche Utente	Lambda Expressions	Lab: Esercizi su eventi e interfacce grafiche
Settimana 12	Reflection		Lab: Esercizi su eventi e