

1. Laboratorio 1

Esercizio 1.0

Presentazioni dei Tutor Junior:

- [Il tutor junior](#)
- [JDK \(Java Development Kit\)](#)
- [Linux](#)

Esercizio 1.1

Obiettivo: familiarizzare con l'informazione disponibile online.

Avviate il browser Firefox tramite interfaccia grafica e mouse oppure da terminale scrivendo il comando **firefox**. Visitare i seguenti siti di interesse didattico per questo corso:

- Visitate il sito dell'aula Taliercio www.adt.unipd.it : dal menu' di sinistra selezionate "Guide online". Dall'elenco della documentazione disponibile individuate "Java openjdk 11.0.12". Questa documentazione sarà disponibile anche durante l'esame, quindi tenete bene presente come accedervi!
- Visitate il sito moodle del corso Fondamenti di Informatica (A) accessibile da <http://stem.elearning.unipd.it/> . Familiarizzate con le varie sezioni. In particolare consultate nella sezione "Informazioni sul corso" i documenti relativi a "[Calendario delle lezioni e programma svolto](#)" e "Modalità d'esame".
- Visitate il sito dell'università www.unipd.it e familiarizzate con le varie sezioni. In particolare consultate le informazioni relative al COVID-19 : <https://www.unipd.it/segnalazione-positivita-covid-19>

Esercizio 1.2

Obiettivo: Individuare e configurare l'editor

Configurare l'editor prescelto (in aula Taliercio è disponibile gedit) possibilmente con le impostazioni seguenti (tipicamente da menu Preferenze):

- mostrare il numero di riga
 - serve per facilitare la correzione degli errori segnalati dal compilatore
- evidenziare la riga corrente
 - aiuta nella navigazione all'interno del codice
- evidenzia le parentesi corrispondenti
 - aiuta nel verificare la presenza e l'accoppiamento delle parentesi
- ampiezza tabulazione pari a 3 spazi
 - rende il codice ordinato
- inserisci spazi anziché tabulazioni
 - rende più facile l'interscambio di codice sorgente Java

Esercizio 1.3

Obiettivo: Lavorare da riga di comando in Linux (bash shell)

Aprire un terminale e lavorare da riga di comando, eseguendo i comandi riportati nel file [shell linux](#) .

Esercizio 1.4

Usando uno degli editor di testo disponibili sul sistema, scrivere il file **Hello.java**, copiando con precisione quanto segue:

```

Hello.java (~/.javaExamples/lezioni2021/)
1 public class Hello{
2
3     public static void main(String[] args)
4     {
5         // Visualizza un messaggio in output
6         System.out.println("Hello, World!");
7     }
8
9 }

```

Terminata la copiatura, salvare il file e uscire dall'editor. Aprire una shell ed eseguire il comando (di compilazione Java):

javac Hello.java

Se il comando di compilazione non va a buon fine e vengono segnalati errori, usare di nuovo l'editor per correggere il file Hello.java.

Se la compilazione va a buon fine (ovvero non viene segnalato alcun errore), verificare che sia stato creato il file Hello.class. Aprire una seconda shell ed eseguire il comando (di esecuzione Java):

java Hello

Verificare che sulla finestra della shell, subito dopo la riga in cui è avvenuta l'esecuzione del comando, venga visualizzato il messaggio:

Hello, World!

Usare nuovamente l'editor per modificare il messaggio da visualizzare in modo che venga appaia un messaggio di saluto a voi stessi. Salvate, compilate ed eseguite. Nel mio caso verrebbe visualizzato il messaggio:

Hello, Cinzia!

Esercizio 1.5

Utilizzare da terminale il comando `ls` per verificare che i file Hello.java e Hello.class siano presenti. Eliminare il file Hello.class e provare ad eseguire di nuovo il comando di esecuzione Java:

java Hello

verificando di ottenere una segnalazione d'errore.

Generare nuovamente il file Hello.class mediante l'esecuzione del compilatore Java:

javac Hello.java

quindi tentare di nuovo l'esecuzione (con **java Hello**), verificando che venga visualizzato il messaggio di saluto con il vostro nome, nel mio caso:

Hello, Cinzia!

Esercizio 1.6

Scaricare e compilare i file .java nella cartella "Sorgenti con Errori" nella sezione "Attività laboratoriale" (o direttamente a questo [link](#)). Trovare e correggere l'errore. Compilare e verificare che la compilazione non generi errori.

Esercizio 1.7

Scrivere un programma che disegni delle facce stilizzate usando i caratteri della tastiera, ad esempio come queste (e magari un po' migliori!).

NB: evitate il carattere `\` (back-slash) perché darebbe problemi. Vedremo prossimamente perché. Evitate anche il carattere doppio-apice `""`, che verrebbe interpretato come fine stringa. Vedremo prossimamente come fare per usarlo nelle stringhe.

```

//|||||  ###|||###  %%%%%%%
(l O O l) (l O O l)  ((~~~~~))
 ( V )    I 8 I      ))^ ^((
 (<_>)   ( VwwwV )   (( .. ))
<_____> <_____>   ))--((

```