

---

Laboratorio di Programmazione I

Gruppo A – Turno 2

**Lezione n. 01: Introduzione al corso**

Alessandro Mazzei

08-10-2021

---

# Informazioni sul corso di Laboratorio

---

Laboratorio di programmazione 1 (A – T2) - 30 ore

- **Docente:** Alessandro Mazzei
- **Lezioni:** 30 ore – 10 lezioni
  - **Orario:** Venerdì ore: 14:00-17:00
  - **Lab.** Turing/Aula virtuale WebEx (link sulla pagina del corso)
- **Orario e luogo di ricevimento:**
  - **In generale:** martedì dalle 15.00 alle 17.00 (email)
- **Email:** [alessandro.mazzei@unito.it](mailto:alessandro.mazzei@unito.it) [mazzei@di.unito.it](mailto:mazzei@di.unito.it)
- **Personal WEB:** <http://www.di.unito.it/~mazzei>
- **Studenti Collaboratori:** Lorenzo Bonannella, Sebastiano Gagliardi

# Alessandro Mazzei

---

- **Ricerca** -> Dipartimento di Informatica
  - Linguistica Computazionale (sintassi, semantica, parsing, dialogo, generazione del linguaggio naturale)
- **Didattica**
  - Triennale Informatica -> Lab. di Prog. A - T2
  - Magistrale Informatica -> Tec. linguaggio naturale
  - Dottorato Informatica -> Conversational Interfaces
- **Compiti amministrativi**
  - Biblioteca, Trasferimenti, Materiali accessibili

# Pagina WEB del corso

## Corso di Studi in Informatica - Università di Torino

Home ► Corsi ► Anno Accademico 21/22 ► Primo anno Laurea DM270 Gestisci i corsi

**NAVIGAZIONE**

- Home
- Dashboard
- Pagine del sito
- I miei corsi
- ▼ Corsi
  - ▼ Anno Accademico 21/22
    - ▼ Primo anno Laurea DM270
      - Programmazione I A LAB-T2 (cognomi A-D, penultima ...)
      - MD21-22
      - Inglese\_I\_2021\_2022
      - ProgI-B-LABT1-21/22-penDispari-Fiandrotti
      - ProgI-C-21/22-Beccuti
      - ProgI-C-LABT1+T2-

Categorie di corso:

Anno Accademico 21/22 / Primo anno Laurea DM270

Cerca corsi:  Vai

Programma	Programmazione I A LAB-T2 (cognomi A-D, penultima cifra matricola PARI), AA 21/22	 
Programma	Matematica Discreta 21-22	  
Programma	Esercitazioni di Inglese	  
Programma	Programmazione 1 B (cognomi E-O, penultima cifra matricola DISPARI) LAB-T1, AA 21/22	 
Programma	Programmazione 1 C (cognomi P-Z), AA 21/22	  
Programma	Programmazione I C LAB-T1+T2 (cognomi P-Z), AA 21/22	  

Moodle, CMS (content management system)

- TEORIA-A-> <https://informatica.i-learn.unito.it/course/view.php?id=2344>
- LAB-AT2 -> <https://informatica.i-learn.unito.it/course/view.php?id=2049>

# Quanto tempo per imparare a programmare?

---

- 24 ore, 7 giorni, 21 giorni, 5 anni, o 10 anni?

# Quanto tempo per imparare a programmare?

- 24 ore, 7 giorni, 21 giorni, 5 anni, o 10 anni?
- *Perché vanno tutti di fretta?*



<http://norvig.com/21-days.html>

<https://web.archive.org/web/20130521001215/http://www.culturahacker.it/documenti/teach.html>

- *Programmare male è facile. Gli idioti possono impararlo in 21 giorni, anche se sono stupidi.*
- *Coders at Work Reflections on the Craft of Programming*  
<https://www.apress.com/it/book/9781430219484>

# Indice

---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!

# Indice

---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!



# Strutture didattiche

- **Ateneo:** [www.unito.it](http://www.unito.it)
  - servizi dell' Ateneo
  - credenziali di Ateneo, fornite all'immatricolazione
  - [nome.cognome@edu.unito.it](mailto:nome.cognome@edu.unito.it)
  - e-learn
  - iscrizioni esami
  - ...
- **Dipartimento:** [www.educ.di.unito.it](http://www.educ.di.unito.it)
  - servizi del dipartimento
  - credenziali dipartimentali (login)
  - [cognome.nome@educ.di.unito.it](mailto:cognome.nome@educ.di.unito.it)
  - ...

# Galateo di comunicazione

- **Come contattare il docente:** lo studente per contattare il docente deve usare la propria casella postale di ateneo [nome.cognome@edu.unito.it](mailto:nome.cognome@edu.unito.it), inoltre deve spedire alla casella di ateneo del docente
- Comunicare con il docente alla mail [alessandro.mazzei@unito.it](mailto:alessandro.mazzei@unito.it) (indicata anche sulla pagina del laboratorio) o con il forum di discussione (sempre sulla pagina del corso).
- È vietato durante le esercitazioni scattare fotografie al docente, girare video, registrare suoni ed in generale avere comportamenti non adeguati al contesto (*si può dormire ma non russare*).  
**Legalmente: non si possono distribuire contenuti**

# Prima cosa da fare

---

- Provare subito il funzionamento delle proprie credenziali di Ateneo e di Dipartimento
- Iscrivarsi alla pagina del corso A di teoria e laboratorio su i-learn.

# Accedere alle macchine del laboratorio

---

- Per il **laboratorio Turing**:

Accesso con il login di Ateneo (lo stesso che usate per accedere a [www.unito.it](http://www.unito.it)) e per la posta elettronica `cognome.nome@edu.unito.it`

- Trovate già l'ambiente Java installato, con un editor di testo per scrivere codice in linguaggio Java.

- Per chi vuole usare il suo portatile a casa:

Procediamo ad installare un ambiente in cui si possa scrivere, compilare ed eseguire programmi in linguaggio Java.

# Indice

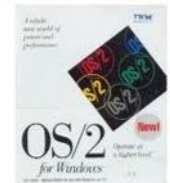
---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!

# Interagire con il computer: al principio era il testo

- Nei primi PC (anni '80), l'utente interagiva con il Sistema Operativo attraverso un'interfaccia testuale

- **Shell** di UNIX, VAX, OS/2:



- **Comand prompt** di MS DOS:



# Interagire con il computer

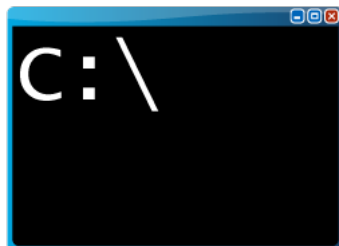
---

- tutte le operazioni avvenivano impartendo comandi in forma testuale;
- anche le applicazioni avevano una interfaccia basata su testo;
- il mouse non esisteva
  - > LISA <https://www.youtube.com/watch?v=m515f5p5e1w>
- la modalità grafica veniva usata solo da particolari applicazioni con output grafico



# Interagire con il computer

- I Sistemi Operativi odierni hanno conservato per l'utente la possibilità di interagire con essi attraverso l'uso di tali interfacce testuali relegandole in finestre del sistema a finestre.
- Principali interpreti di comandi:
  - In **MS Windows**: è conosciuto come il “Prompt dei comandi” (cmd.exe);
  - Nella famiglia **UNIX** (es. Linux, OS X): sono i programmi “shell” (bash, sh, csh ...).



Mac OS





```
amazon:~ depierro$ ls -l
total 8
drwxr-xr-x  8 depierro  staff   272 Jul 10  2014 Books
drwx-----@ 10 depierro  staff   340 Oct  9 09:13 Desktop
drwx-----+ 14 depierro  staff   476 Apr 23 16:12 Documents
drwx-----+ 86 depierro  staff  2924 Jul 23 09:32 Downloads
drwx-----@ 54 depierro  staff  1836 Aug  1  2014 Library
drwx-----+  8 depierro  staff   272 Mar  4  2015 Movies
drwx-----+  4 depierro  staff   136 Mar  4  2015 Music
-rw-r--r--  1 depierro  staff   108 Oct  9 09:46 MyClass.java
drwx-----+ 23 depierro  staff   782 Jun 12 11:56 Pictures
drwxr-xr-x+  5 depierro  staff   170 Feb 20  2014 Public
amazon:~ depierro$ cd /usr/bin
publ class@mintmacbook ~ $ which javac
/usr/bin/javac
} class@mintmacbook ~ $ which java
amazon:/usr/bin/java
MyC class@mintmacbook ~ $ echo $PATH
publ /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games
MyC class@mintmacbook ~ $ ls -l
total 52
MyC drwxr-xr-x  4 class class 4096 feb 18  2014 Applica
} -rw-r--r--  1 class class  144 dic  6  2013 C:\nppd
2 drwxr-xr-x  5 class class 4096 nov  3  2014 Desktop
3 er drwxr-xr-x  2 class class 4096 dic  3  2013 Document
amazon: drwxr-xr-x  2 class class 4096 dic 11  2013 Downloa
drwxr-xr-x  2 class class 4096 mar  1  2014 eps
drwxr-xr-x  2 class class 4096 dic  3  2013 Music
drwx-----  2 class class 4096 feb 18  2014 nets
drwxr-xr-x  2 class class 4096 dic  3  2013 Pictures
drwxr-xr-x  2 class class 4096 mar  1  2014 ps
drwxr-xr-x  2 class class 4096 dic  3  2013 Public
drwxr-xr-x  2 class class 4096 dic  3  2013 Template
drwxr-xr-x  2 class class 4096 dic  3  2013 Videos
class@mintmacbook ~ $
```

```
C:\windows\system32\cmd.exe
C:\Users\depierro>cd \
C:\>cd "Program Files"
C:\Program Files>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 56FF-4E3F

Directory di C:\Program Files

06/10/2014  17:20    <DIR>      -
06/10/2014  17:20    <DIR>      ..
06/10/2014  17:20    <DIR>      Common Files
12/04/2011  06:16    <DIR>      Internet Explorer
10/10/2013  13:15    <DIR>      Java
14/07/2009  06:51    <DIR>      MSBuild
12/11/2013  18:01    <DIR>      NetBeans 7.4
30/12/2013  16:27    <DIR>      Paragon Software
14/07/2009  06:51    <DIR>      Reference Assemblies
09/12/2013  18:15    <DIR>      Samsung
10/10/2013  10:05    <DIR>      TextPad 7
06/10/2014  17:20    <DIR>      VMware
12/04/2011  06:16    <DIR>      Windows Defender
12/04/2011  06:25    <DIR>      Windows Journal
12/04/2011  06:16    <DIR>      Windows Mail
07/10/2013  10:14    <DIR>      Windows NT
12/04/2011  06:16    <DIR>      Windows Photo Viewer
12/04/2011  06:16    <DIR>      Windows Sidebar
               0 File
               18 Directory  55.198.535.680 byte disponibili

C:\Program Files>cd Java
C:\Program Files\Java>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 56FF-4E3F

Directory di C:\Program Files\Java

10/10/2013  13:15    <DIR>      -
10/10/2013  13:15    <DIR>      ..
10/10/2013  10:03    <DIR>      docs
10/10/2013  13:15    <DIR>      jdk1.7.0_40
10/10/2013  09:56    <DIR>      jre7
               0 File
               5 Directory  55.198.535.680 byte disponibili

C:\Program Files\Java>
```

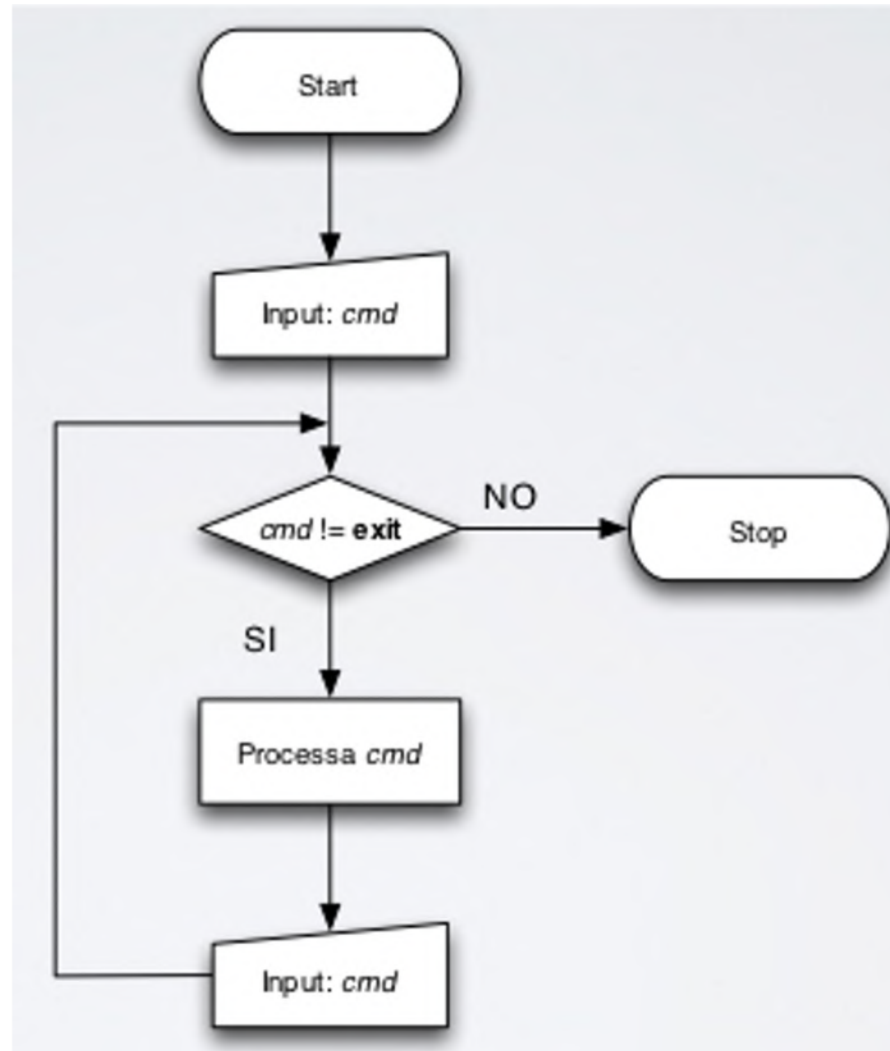
# Interagire con il computer con una CLI

---

*Command Line Interface* (CLI) come interprete dei comandi:

- 1) il programma si presenta all'utente con un prompt e un cursore testuale di solito lampeggiante che indica l'attesa di input.
- 2) l'utente inserisce una riga di testo: le regole sintattiche da rispettare per scrivere i comandi dipendono dal tipo di interprete.
- 3) dopo l'invio il programma interpreta ciò che l'utente ha scritto ed elabora il risultato producendo se previsto un output.

# CLI flowchart



# Sintassi di un comando

- In generale un comando è composto da un nome eventualmente seguito dalle sue opzioni e dagli argomenti!
- **nome-comando** [opzioni] [argomenti]
  - > gli argomenti sono i dati sui quali il comando opera;
  - > le opzioni definiscono modi alternativi nei quali un comando può operare

## Windows:

```
> date /t  
> systeminfo  
> echo ciao, come stai
```

## UNIX:

```
$ date  
$ uname -a  
$ echo "ciao, come stai"
```

*P.S. nella shell UNIX, le opzioni sono introdotte da un trattino '-', in Windows da '/'*

# Comandi built-in

---

- Alcuni comandi, detti **built-in**, sono implementati ed elaborati direttamente dall'interprete, altri sono programmi esterni al programma interprete e sono memorizzati su file, localizzati in particolari cartelle, che l'interprete si limita a caricare in memoria e ad eseguire.
- la variabile d'ambiente **PATH**

# File Naming

---

- In generale molti comandi lavorano su dati memorizzati in file, gli argomenti del comando devono indicare i nomi dei file sui quali operare, esempio:

```
$ rm <file>
```

```
$ cp <file su chiavetta USB> <lettore  
portatile>
```

- C'è bisogno di una maniera che usi una forma testuale per indicare un file nel sistema di cartelle:
  - > Il **pathname**, o nome del percorso, di un file;
- Non è sufficiente scrivere semplicemente il nome del file!

# File Naming

---

- Il **pathname assoluto** di un file è il nome completo di un file nel sistema di cartelle e si forma componendo in ordine i nomi delle cartelle da attraversare a partire dalla cartella radice del sistema per arrivare al file, separando gli elementi costituenti con il carattere '/' (in Unix ) o '\' (in Windows).
- Esempio (Unix e Windows):

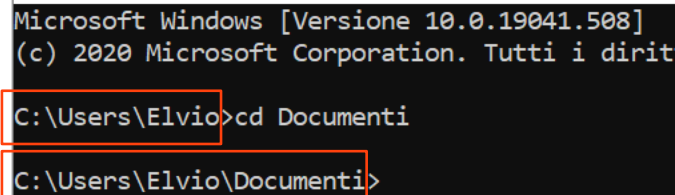
```
/home/mazzei$ rm /home/mazzei/Documenti/todo.txt
```

```
C:\Users\alex> del C:\Users\alex\Documenti\todo.txt
```

rimuove il file dal nome **todo.txt** che si trova nella cartella **Documenti** dell'utente **mazzei**;

# File Naming

- Il pathname assoluto è uno schema molto rigido e non va bene in tutte le situazioni, in alcuni contesti è più utile uno schema di naming relativo.
- Il **pathname relativo** lavora unitamente al concetto di **cartella di lavoro (working directory)**.

A screenshot of a Windows command prompt window. The title bar reads "Microsoft Windows [Versione 10.0.19041.508]". The text inside shows the copyright notice "(c) 2020 Microsoft Corporation. Tutti i diritti sono riservati." followed by two command lines. The first line is "C:\Users\Elvio>cd Documenti" and the second line is "C:\Users\Elvio\Documenti>". Red rectangular boxes highlight the directory paths "C:\Users\Elvio" and "C:\Users\Elvio\Documenti" in the prompts.

```
Microsoft Windows [Versione 10.0.19041.508]
(c) 2020 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Elvio>cd Documenti
C:\Users\Elvio\Documenti>
```

- La working directory viene indicata prima del '>' nel prompt dei comandi (o \$ in unix).



# La working directory

---

- In ogni istante una **specifica cartella**, nel sistema di cartelle e file presenti sul PC, è considerata la **cartella di lavoro (working directory)**.
- I nomi dei comandi e dei file passati come argomenti ai comandi possono essere relativi alla cartella di lavoro, oppure assoluti.
- **Pathname relativo**: è simile al nome assoluto del file ma la cartella iniziale è la cartella di lavoro.
- Esempio, se **/home/mazzei** è la cartella di lavoro, allora il seguente comando:

**rm Documenti/cosedafare.txt**

è equivalente a

**rm /home/mazzei/Documenti/cosedafare.txt**

# Directory di Laboratorio

---

- In DOS/Windows le lettere (seguite dai : ) indicano i dischi
- **C:** disco locale dove trovare i programmi (compilatore e interprete java, notepad)
- **D:** o altri nomi: altri volumi che avete sul vostro PC  
(in laboratorio Turing abbiamo solo C:)

# Alcuni comandi DOS:

[https://it.wikipedia.org/wiki/Comandi\\_di\\_MS-DOS](https://it.wikipedia.org/wiki/Comandi_di_MS-DOS)

- **cd dir**                      *Cambia la working directory in dir*
- **cd ..**                      *Cambia la working directory con la directory superiore*
- **dir**                      *Elenca il contenuto della working directory*
- **dir path**                      *Elenca il contenuto della directory path*
- **mkdir dirA**                      *Crea una nuova directory dirA vuota*
- **rmdir dirA**                      *Rimuove la directory dirA (deve essere vuota)*
- **copy fileA fileB**                      *Copia il contenuto del fileA nel nuovo fileB*
- **copy NUL fileA**                      *Crea un file vuoto di nome fileA*
- **ren fileA fileB**                      *Cambia il nome del fileA in fileB*
- **move pathToA pathToB**                      *Sposta il file pathToA nella nuova posizione pathToB*
- **del fileA**                      *Elimina il fileA*
- **type fileA**                      *Stampa il contenuto del file fileA nel terminale*
- **start notepad++ fileA**                      *Apri il fileA nel programma Notepad++*
- **C: D: X:** *Cambia il disco attivo (radice del filesystem)*

# Alcune directory

- Nel pathame testuale le stringhe “.” e “..” indicano cartelle particolari
- “.” è sinonimo di cartella di lavoro. Un esempio, supponendo che la cartella di lavoro sia **Documenti**, i due comandi sono del tutto equivalenti:

```
$ rm ./verbale.txt
```

```
$ rm verbale.txt
```

- “..” è la cartella che *contiene* la cartella di lavoro: serve per risalire nella gerarchia delle cartelle. Esempio, supponendo che la cartella di lavoro sia la cartella **Documenti** nella home di **mazzei**, e che si voglia copiare in essa il file **pippo.jpg** memorizzato nella cartella **shared/foto/** dell’utente **topolino**, allora si può scrivere

```
$ cp ../../topolino/shared/foto/pippo.jpg .
```

# Esercizio sul file e directory

---

- Creare nuova cartella PROVA e (con Notepad++) un nuovo file Miaprova
- Rinominare il file in MiaProva1
- Fare una copia di MiaProva1 (usando pathname assoluto e poi relativo)
- Cancellare la copia
- Risalire alla cartella genitore di PROVA
- Passare al disco C: (o Z:, se disponibile)
- Rientrare nella cartella PROVA

# Comandi esterni

---

- quando il nome del comando inserito da un utente non corrisponde ad alcun comando **built-in** riconosciuto dall'interprete allora questo assume che il comando sia implementato da un programma esterno;
- l'interprete interpreta il nome-comando come nome (assoluto o relativo) del file contenente il programma da caricare in memoria e da eseguire;
- quando il nome del file del programma è relativo l'interprete lo cerca in un insieme di cartelle predefinite dalla variabile d'ambiente **PATH**

# Esercizio su variabili d'ambiente

---

- verificare in MS DOS il valore della variabile PATH:

> echo %PATH% (Windows)

\$ echo \$PATH (Unix)

- le variabili d'ambiente hanno di solito nomi composti da caratteri in maiuscolo;
- In Windows il valore di una variabile d'ambiente viene specificato racchiudendo il nome tra una coppia di caratteri %, in Unix si antepone il \$

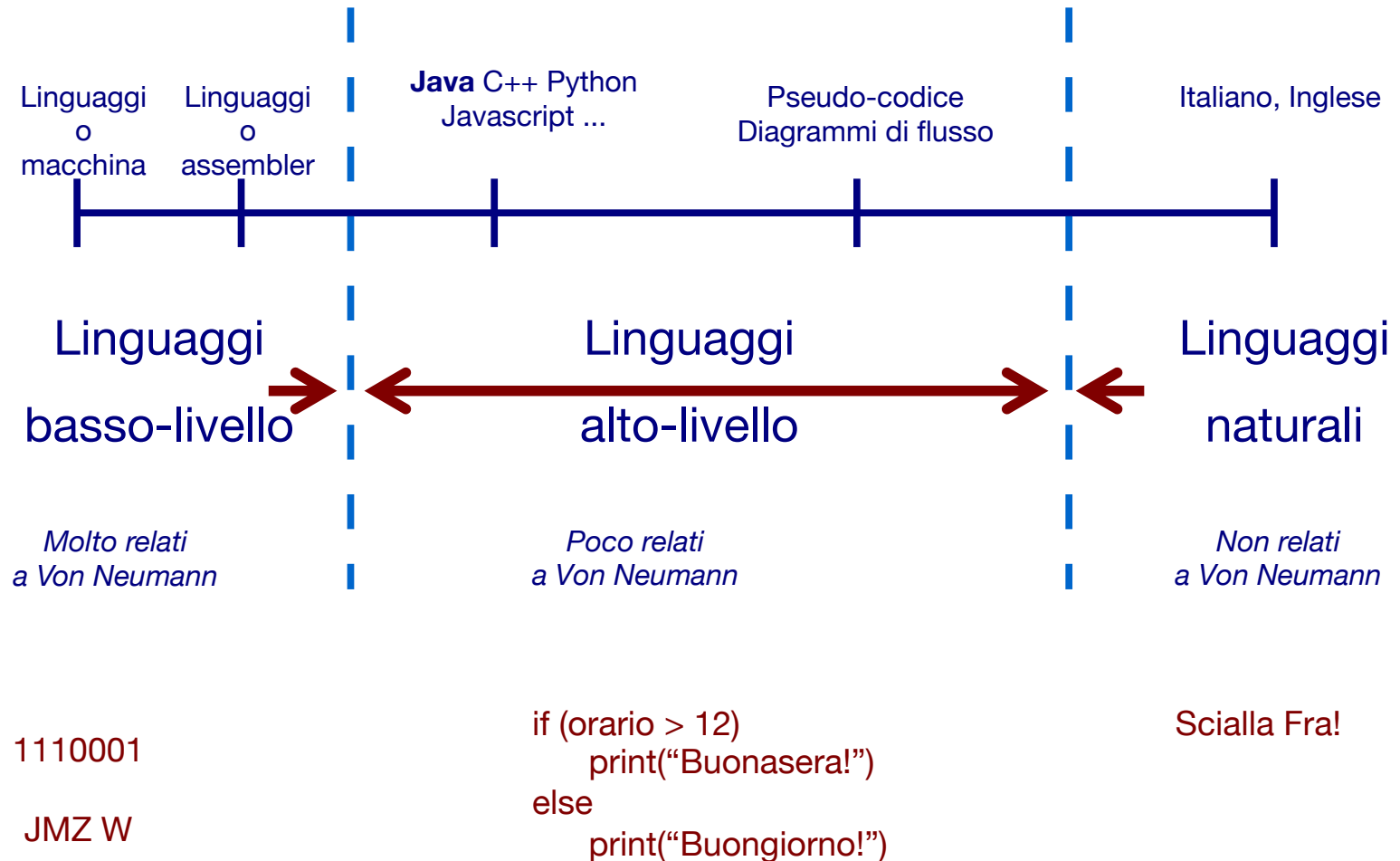
# Indice

---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!



# Macchine, macchine virtuali virtuali e Babele



## Sistema Operativo

C, Python, ...

## Sistema Operativo

Java Virtual Machine (**JVM**)

Java

# Java ... 1,2,3,4,5,6,7,8, ..., 17

---

- MultiParadigma: imperative, concurrent, *class-based*, *object-oriented*, *functional*
- 1991: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)#History](https://en.wikipedia.org/wiki/Java_(programming_language)#History)
- Java Virtual Machine
  - *write once, run anywhere* (WORA)
  - linguaggi host
  - Orientato agli oggetti, strutturato in classi
- **Noi lo useremo in questo corso come un linguaggio imperativo semplice (c-style)**

# Installare Java

- Installare **JDK SE: Java Development Kit Standard Edition**. Alla pagina del corso su e-learn è pubblicato il link diretto al sito web del produttore Oracle per il download: <https://www.oracle.com/java/technologies/downloads/>

ORACLE

Products Industries Resources Support Events Developer

View Accounts Contact Sales

Java downloads Tools and resources Java archive

Java > Technical Details >

## Java Downloads

Looking for other Java downloads? Open JDK Early Access Builds Java for Consumers

### Java 17 available now

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions License.

JDK 17 will receive updates under these terms, until at least September 2024.

Learn about Java SE Subscription

### Java SE Development Kit 17 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

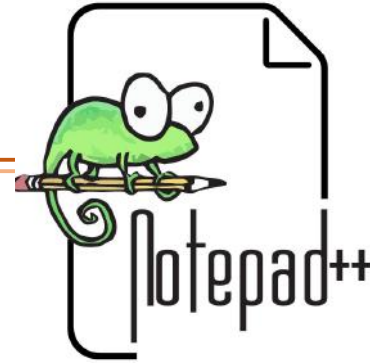
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Documentation Download

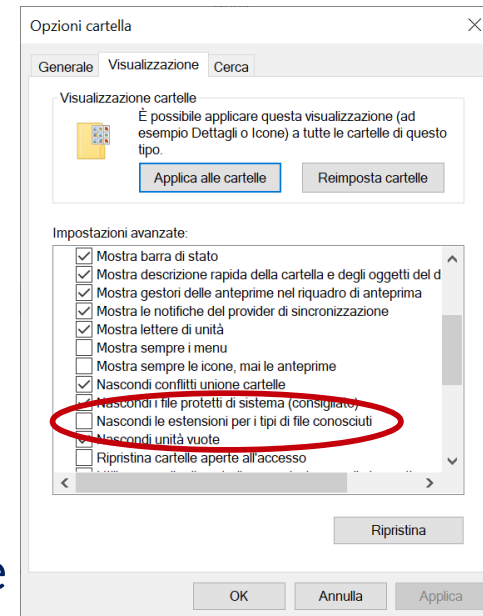
Linux macOS Windows

Product/file description	File size	Download
Arm 64 Compressed Archive	170.95 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.tar.gz</a> (sha256 <a href="#">[2]</a> )
Arm 64 RPM Package	153.12 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.rpm">https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.rpm</a> (sha256 <a href="#">[2]</a> )
x64 Compressed Archive	172.19 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz">https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz</a> (sha256 <a href="#">[2]</a> )

# Installare Notepad++



- Un editor in generale potrebbe aggiungere estensione .txt a files (mentre il codice Java deve essere scritto in file che terminano con estensione .java)
- Un editor come NotePad è inadatto alla scrittura di codice. Usiamo un editor con funzioni più avanzate per la programmazione, come **Notepad++**. (Ci sono molti altri editor possibili).
- **Suggerimento:** sotto Windows disabilitare l'opzione che nasconde le estensioni dei nomi dei file per i file di formato “conosciuto”. In questo modo verifichiamo che i file di codice che andremo a scrivere abbiano effettivamente estensione .java



# Dove si installa JDK?

- Installazione di JDK SE: l'installazione della JDK copia nel file-system della macchina i file (strumenti, librerie, ...) necessari al programmatore per sviluppare applicazioni Java ed imposta alcune configurazioni nel sistema;
- Il luogo dove questi file sono posti è **dipendente dal sistema in uso** (Windows, Linux, Mac). In Windows la radice dell'installazione è di solito:

C:\Programmi\Java\JDK-*numeroVersione*

- a partire dalla radice dell'installazione gli strumenti necessari al programmatore Java sono localizzati sotto la cartella "bin" (bin è l'abbreviazione di *binaries=executable*)

# Java Tools

---

Tra gli strumenti in “bin” noi useremo:

- **javac** è il compilatore di programmi scritti in linguaggio Java verso programmi scritti nel linguaggio macchina (**bytecode**) della JavaVirtual Machine (JVM);!
- **java** è chiamato l'interprete Java, in realtà esso crea una istanza della JVM e sopra vi esegue un programma in linguaggio macchina;
- **javap** è il disassembler cioè traduce un programma in linguaggio macchina JVM in un programma in linguaggio assembler;
- **javadoc** è uno strumento per generare automaticamente la documentazione dei propri programmi;



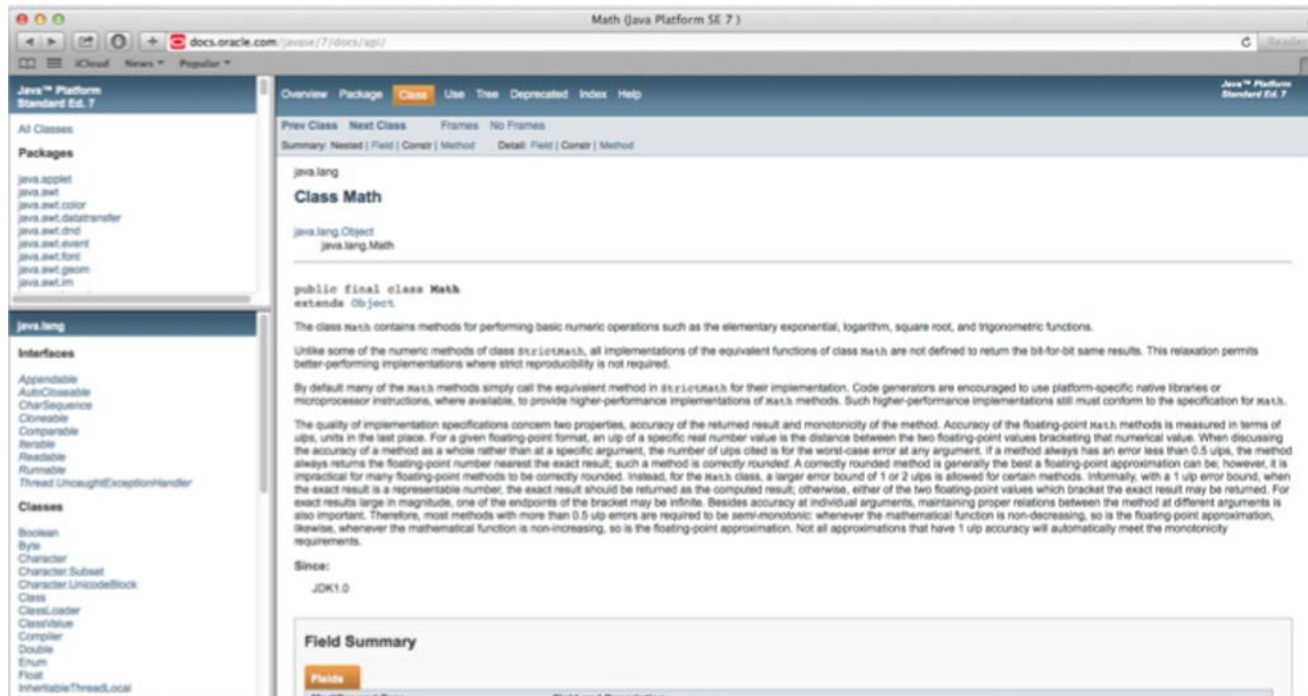
# JavaDoc

---

- Tutta la documentazione delle classi Java disponibili con l'installazione della JDK è scritta usando javadoc: esso è uno strumento che legge un programma Java scritto dall'utente e genera dai commenti inseriti nel sorgente la documentazione usando un formato particolare per il web.
- In Programmazione 1 useremo raramente la documentazione della API Java perché non scriveremo programmi che gestiscono direttamente gli oggetti Java e questo limita l'insieme delle classi usate;
- Tuttavia è utile avere una panoramica sulla struttura di questa documentazione prodotta con javadoc.

# JavaDoc

2 <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>



**Esercizio:** leggere la documentazione della classe `java.lang.Math` disponibile con l'installazione della JDK SE.

# Updating the PATH Environment e JAVA

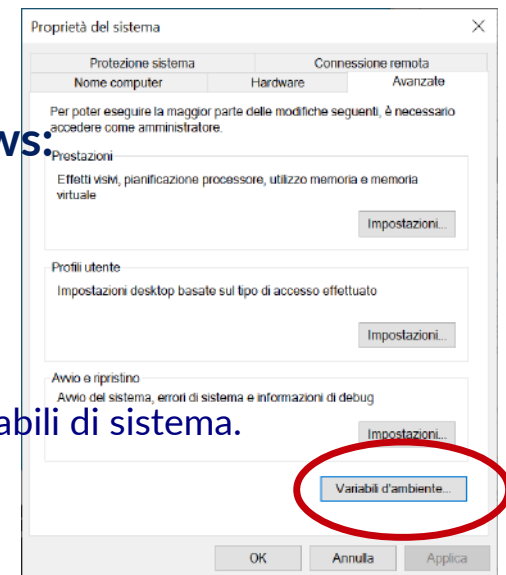
- Se la variabile d'ambiente PATH non viene impostata per puntare anche al JDK, serve specificare ogni volta il path completo a javac/java per compilare ed eseguire:

```
C:\> "C:\Program Files\Java\jdkXXX\bin\javac" MyClass.java
```

- È utile impostare il PATH in modo permanente, così si preserva ai successivi riavvii. Per impostare il PATH in modo permanente, aggiungere il percorso jdkXXX\bin alla variabile PATH (Il percorso da aggiungere assomiglia a C:\Program Files\Java\jdkXXX\bin )

- Per impostare il PATH in modo permanente in Windows.**

- Start > Pannello di Controllo > Sistema.
- Avanzato > Variabili d'Ambiente
- Aggiungere la directory jdkXXX\bin alla variabile PATH nelle variabili di sistema.
- Aggiungere ;. alla fine della linea





# Indice

---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!

# Java e CLI

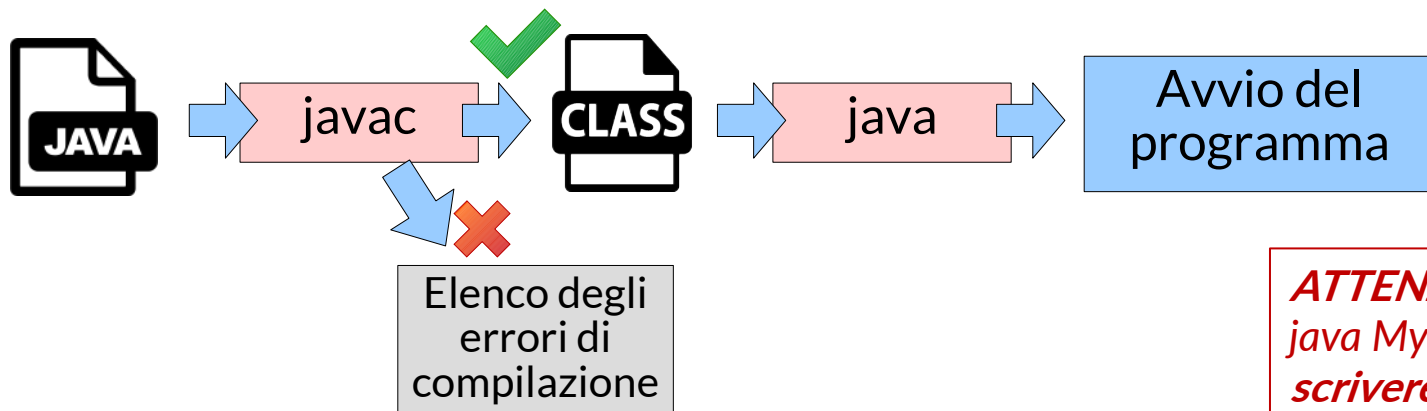
- I file di codice in linguaggio Java hanno estensione **.java** 
- Un file di codice non può essere eseguito direttamente, ma può essere **compilato** (= tradotto in linguaggio JVM, il *bytecode*). I file in bytecode hanno estensione **.class** 
- Un file Java contiene una **classe**. Il nome della classe deve essere lo stesso del file .java
- Esempio: il file **Programma.java** contiene una classe di nome **Programma**, e viene compilato nel file **Programma.class**
- Un file .class può essere eseguito se contiene il metodo **main()**

# Il compilatore e l'interprete

Tutti gli strumenti prima descritti (java, javac, javap, javadoc) **non** sono applicazioni con le quali si interagisce usando un'interfaccia grafica, ma all'origine si usano attraverso un'interfaccia testuale: si invoca lo strumento dal prompt dei comandi:

1. COMPILARE:        **> javac MyClass.java**

2. INTERPRETARE: **> java MyClass**



**ATTENZIONE:**  
*java MyClass **senza**  
scrivere  
l'estensione .class*

# Indice

---

- Preliminari
- Interagire con il computer: la linea di comando (CLI)
- Java e JDK
- Java e CLI
- Hello World!

# HelloWorld [http://www.rosettacode.org/wiki/Hello\\_world/Newbie#Java](http://www.rosettacode.org/wiki/Hello_world/Newbie#Java)

- Creare una working directory PROG1 con dentro una directory Lezione1
- Aprire **NotePad++** e scrivere:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello world!");  
  
    }  
  
}
```

- Salvare il file nella working directory con nome **"HelloWorld.java"**
- ```
> javac HelloWorld.java (produce HelloWorld.class)  
  
> java HelloWorld
```



# Cosa fa l'interprete su una classe?

- L'interprete Java su una classe: **java HelloWorld**
- java cerca nel sistema di file e cartelle un file che ha nome **HelloWorld.class**
  - se non lo trova genera un errore;
  - se lo trova inizia ad eseguire le istruzioni del testo della classe a partire dalla prima del metodo **main**
  - se la classe non contiene tale metodo è generato un errore.
  - La ricerca della classe **HelloWorld** è fatta solo in un insieme di cartelle definite come elenco nella variabile d'ambiente **CLASSPATH**
  - Se **CLASSPATH** non è definita la ricerca è fatta di default solo nella cartella di lavoro
- *Attenzione: si può modificare il classpath con l'opzione **-cp**, oppure cambiando la variabile di ambiente **CLASSPATH**.*

# Leggere e scrivere da CLI 1: Sin.java

Usa la classe Sin.class:

```
public class CiaoNomeSin {  
  
    public static void main(String[] args) {  
  
        System.out.println("Come ti chiami?");  
  
        String nome = Sin.readLine();  
  
        System.out.println("Ciao, " + nome);  
  
    }  
  
}
```

NOTA: Scaricare la classe Sin.java dalla pagina del laboratorio, e compilarla prima di provare questo esercizio.

# Leggere e scrivere da CLI 2: Scanner

```
import java.util.Scanner;

public class CiaoNomeScan {

    public static void main(String[] args) {

        Scanner tastiera = new Scanner(System.in);

        System.out.print("Come ti chiami? ");

        String nome = tastiera.nextLine();

        System.out.println("Ciao, " + nome);

    }

}
```

# Leggere e scrivere con l'interfaccia grafica: Swing

---

Usa la libreria Swing:

```
import javax.swing.JOptionPane;

public class CiaoNomeSwing {

    public static void main(String[] args) {

        String nome = JOptionPane.showInputDialog("Come ti chiami?");

        JOptionPane.showMessageDialog(null, "Ciao, " + nome);

        System.exit(0);

    }}
```