SOFTWARE

- PROGRAMMA, APPLICAZIONE, APP

SOFTWARE DI SISTEMA

SISTEMA OPERATIVO

- WINDOWS
- ANDROID
- IOS
- MAC OS X
- LINUX

↑

CHI LO FA PARTIRE? IL BOOTLOADER?

CHI FA PARTIRE IL BOOTLOADER? STA NELLA ROM!!

SOFTWARE APPLICATIVO

- OFFICE
- GOOGLE CHROME
- VIDEOGIOCHI

COME SI SCRIVE UN PROGRAMMA

- SEQUENZA DI BYTE "LINGUAGGIO MACCHINA"

```
48 c7 c0 01 00 00 00 48 c7 c7 01 00 00 00 48 c7
c6 00 00 00 00 48 c7 c2 0f 00 00 00 0f 05 48 c7
c0 3c 00 00 00 48 c7 c7 00 00 00 00 0f 05 48 65
6c 6c 6f 2c 20 77 6f 72 6c 64 21 0a
```

byte (in base 16)

PROGRAMMA IN LINGUAGGIO MACCHINA CHE VISUALIZZA LA SCRITTA HELLO WORLD! SULLO SCHERMO

Dipende dalla CPU e dal SISTEMA OPERATIVO

PROCESSORI INTEL/AMD 64 bit

S.O. LINUX

- LINGUAGGIO ASSEMBLY

```
.text

 .global _start

_start:
 mov $1, %rax
 mov $1, %rdi
 mov $msg, %rsi
 mov $len, %rdx
 syscall

 mov $60, %rax
 mov $0, %rdi
 syscall

.data
 msg:
  .string "Hello, world!\n"
 msgend:
  .equ len, msgend - msg
```

```
1                       .text
2
3                       .global _start
4
5                       start:
6 0000 48C7C001                 mov $1, %rax
6      000000
7 0007 48C7C701                 mov $1, %rdi
7      000000
8 000e 48C7C600                 mov $msg, %rsi
8      000000
9 0015 48C7C20F                 mov $len, %rdx
9      000000
10 001c 0F05                    syscall
11
12 001e 48C7C03C                mov $60, %rax
12     000000
13 0025 48C7C700                mov $0, %rdi
13     000000
14 002c 0F05                    syscall
15
16                      .data
17                      msg:
18 0000 48656C6C                .string "Hello, world!\n"
18     6F2C2077
18     6F726C64
18     210A00
19                      msgend:
20                       .equ len, msgend - msg
21
```

ASSEMBLY: versione del linguaggio macchina per esseri umani

dipende dal s.o. e dalla CPU.

SI PUÒ ESEGUIRE

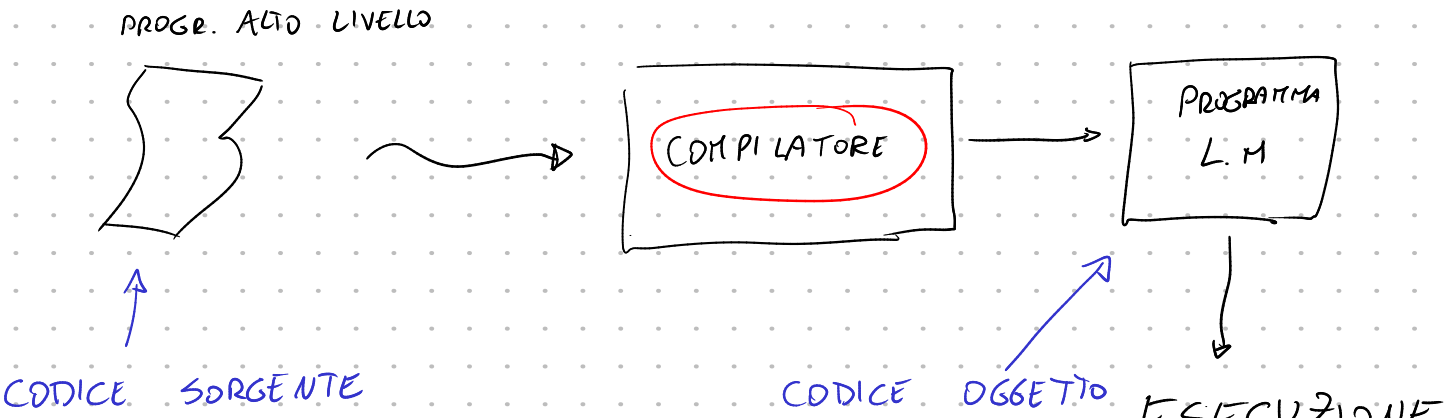
COMMAND LINE INTERFACE / SHELL

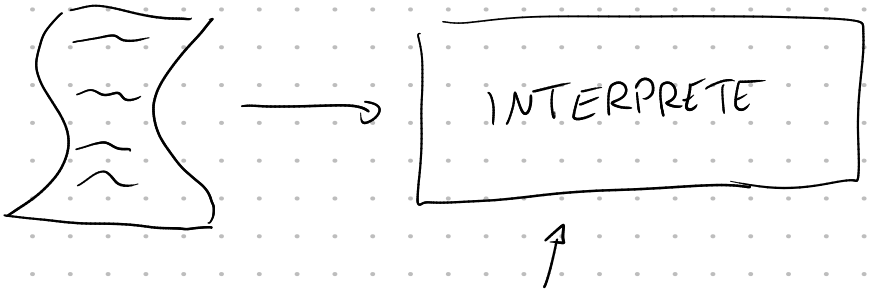
LM e ASSEMBLY: LINGUAGGI A BASSO LIVELLO

LINGUAGGI AD ALTO LIVELLO

```
print("Hello World!")
```

} stesso programma ma in PYTHON





ESEGUE DIRETTAMENTE IL PROGRAMMA AD ALTO LIVELLO SENZA GENERARE IL PROGRAMMA IN L.M. CORRISPONDENTE

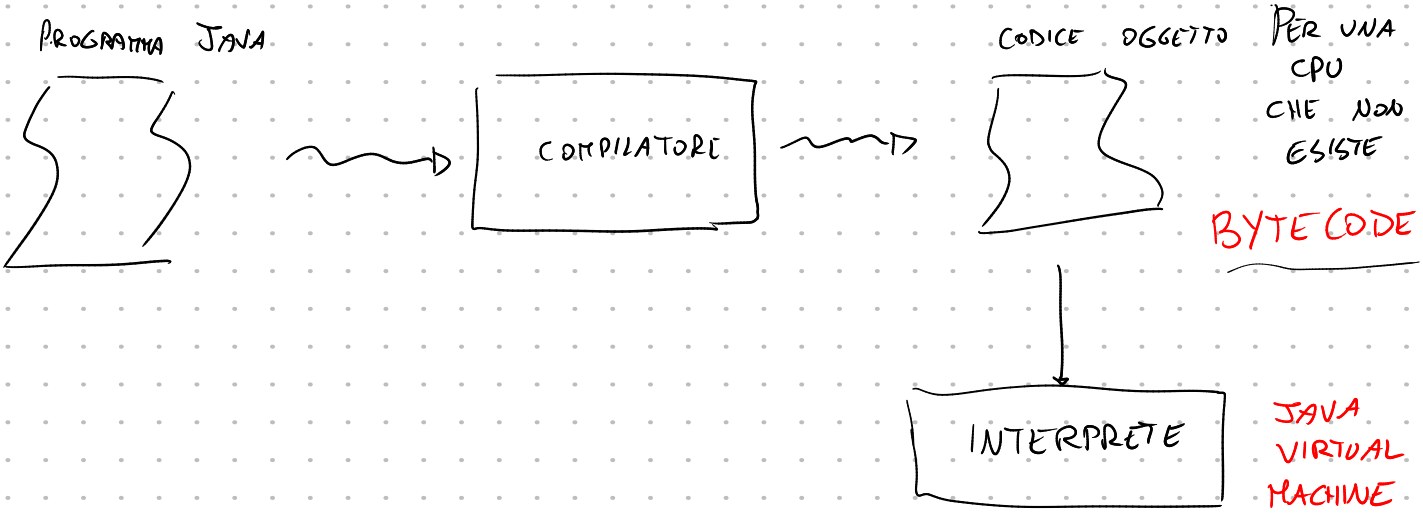
PYTHON È INTERPRETATO

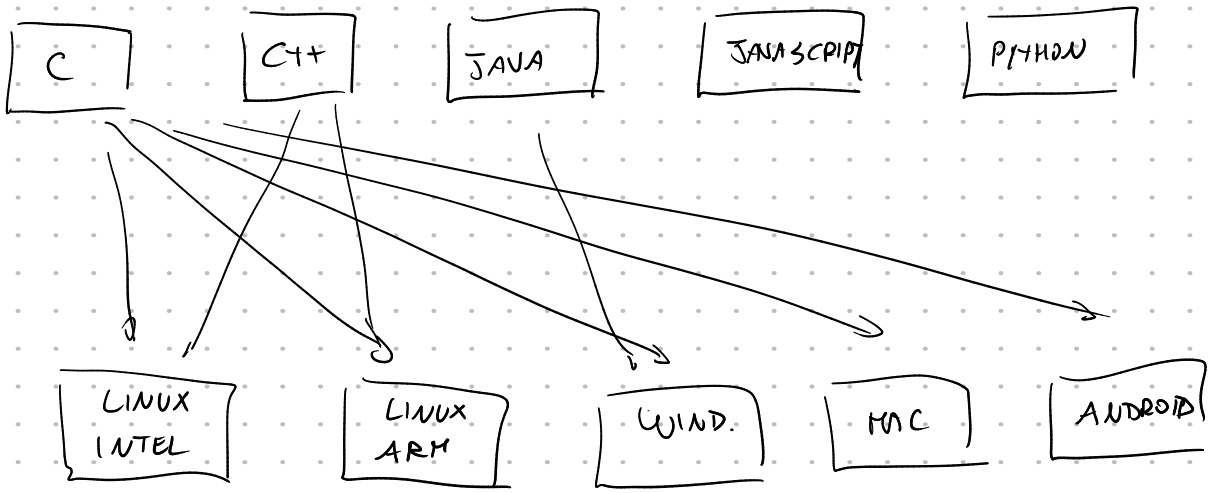
JAVA

- È SEMPLICE (C++, C)
- È MODERNO (rispetto a C)
- NON TROPPO SEMPLICE (rispetto a PYTHON)

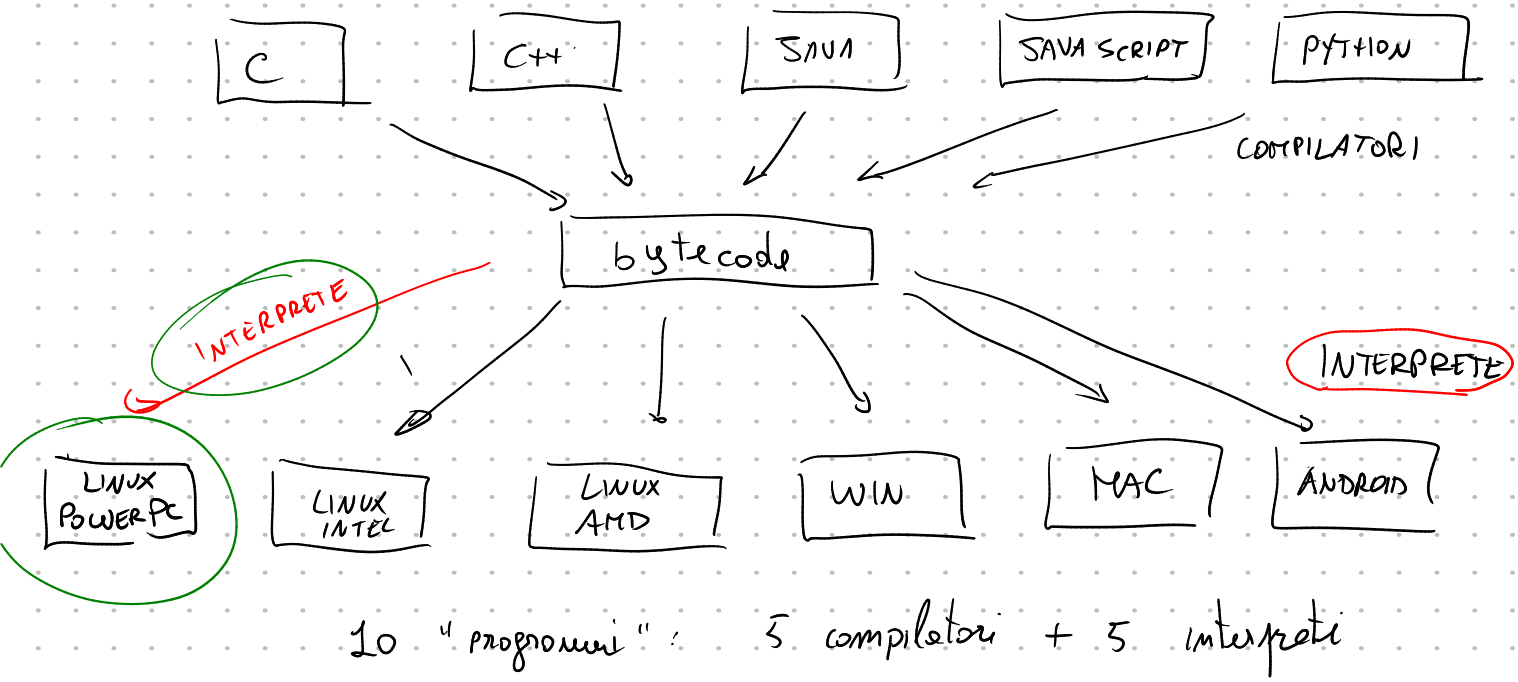
LINGUAGGI STATICI (JAVA, C, C++, C#)

LINGUAGGI DINAMICI (PYTHON, JAVASCRIPT)





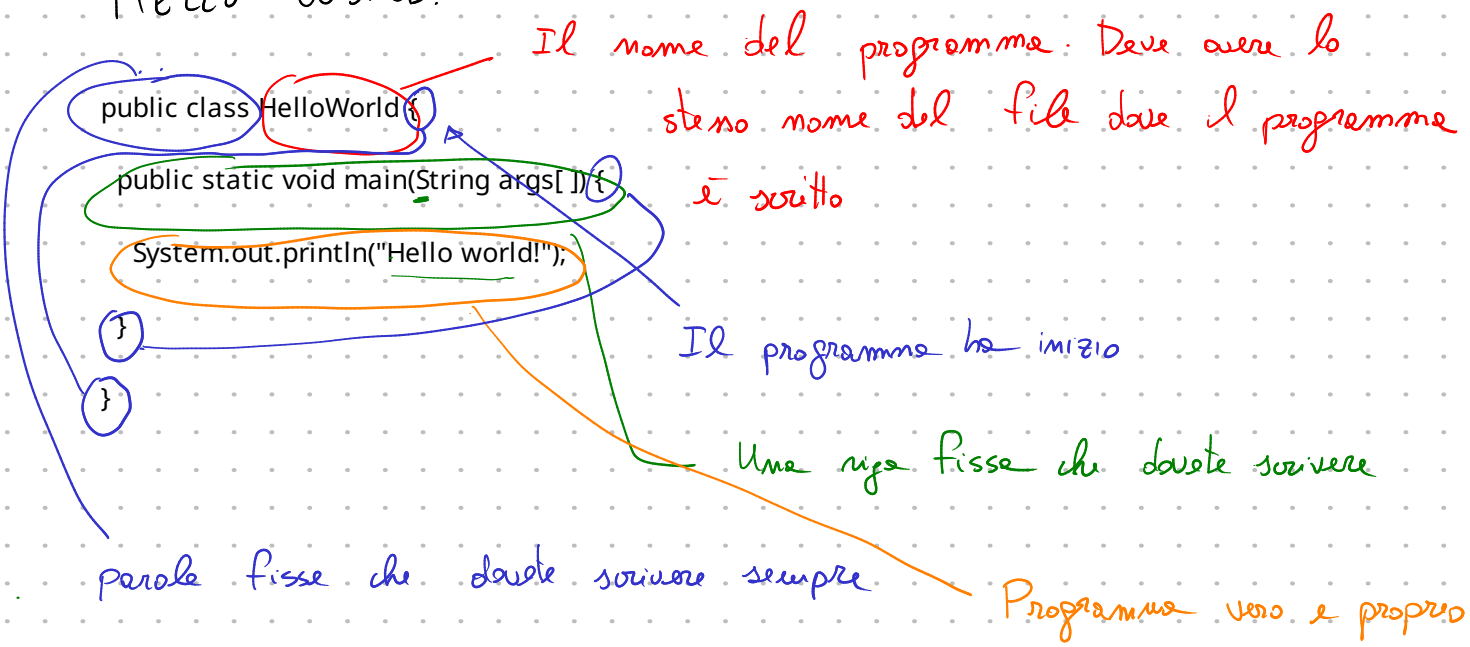
$5 \times 5 = 25$ compilatore



10 "programmi": 5 compilatori + 5 interpreti

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

HELLO WORLD.



SYSTEM.OUT.PRINTLN è il comando che visualizza una STRINGA sullo schermo.

SEQUENZA DI CARATTERI

SYSTEM.OUT.PRINTLN stampa il suo ARGOMENTO (quella cosa che sta tra parentesi tonde)

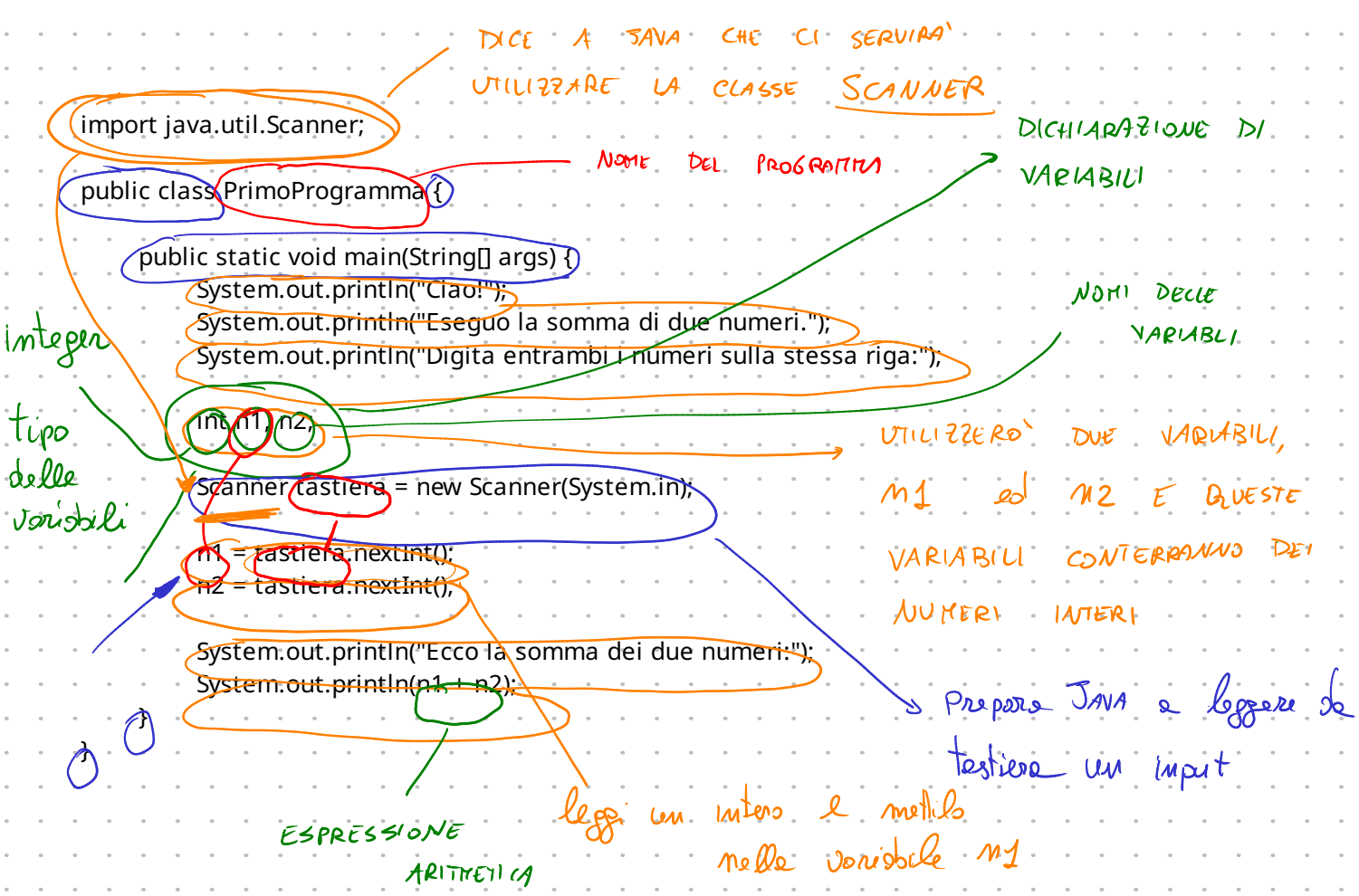
System.out.println (" Hello World! ") ;

DELIMITANO LE STRINGHE CHE SI VOGLIONO STAMPARE

comando

argomento

tutti i comandi terminano con un punto e virgola


DICE A JAVA CHE CI SERVIRÀ UTILIZZARE LA CLASSE SCANNER
import java.util.Scanner;
NOME DEL PROGRAMMA
public class PrimoProgramma {
DICHIARAZIONE DI VARIABILI
public static void main(String[] args) {
System.out.println("Ciao!");
System.out.println("Eseguo la somma di due numeri.");
System.out.println("Digita entrambi i numeri sulla stessa riga:");
NOMI DELLE VARIABILI
integer
tipo delle variabili
int n1, n2;
UTILIZZERÒ DUE VARIABILI, n1 ed n2 E QUESTE VARIABILI CONTERRANNO DEI NUMERI INTERI
Scanner tastiera = new Scanner(System.in);
Prepara JAVA a leggere da tastiera un input
n1 = tastiera.nextInt();
n2 = tastiera.nextInt();
System.out.println("Ecco la somma dei due numeri:");
System.out.println(n1 + n2);
}
}
ESPRESSIONE ARITMETICA
leggi un intero e mettilo nella variabile n1