Metodologie di Programmazione

- Lezione 2: Hello World!

Lezione 2: Sommario

UNITELMA SAPIENZA
SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

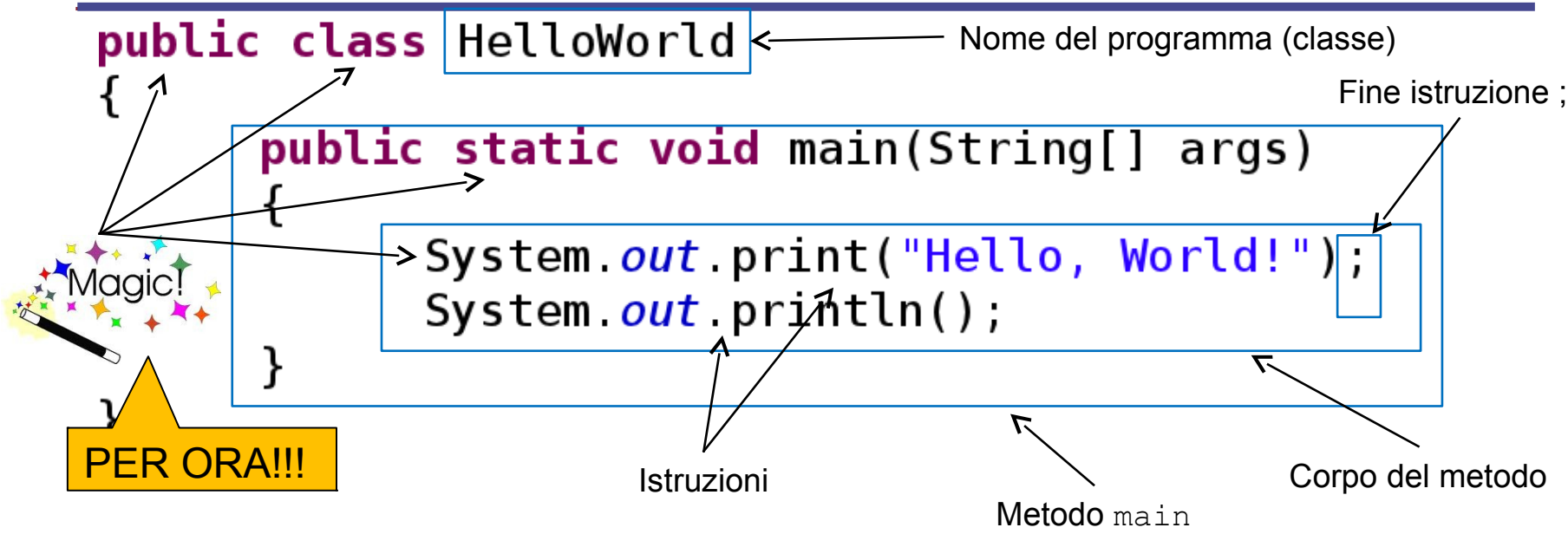
- Un primo programma Java: Hello World!
- Confronto con altri linguaggi di programmazione
- Metodi di base per stampare
- Creazione ed esecuzione del programma mediante ambiente di sviluppo integrato

Hello World!

- Un primo esempio di programma scritto in Java
- Il classico Hello World!

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

Anatomia di Hello World!



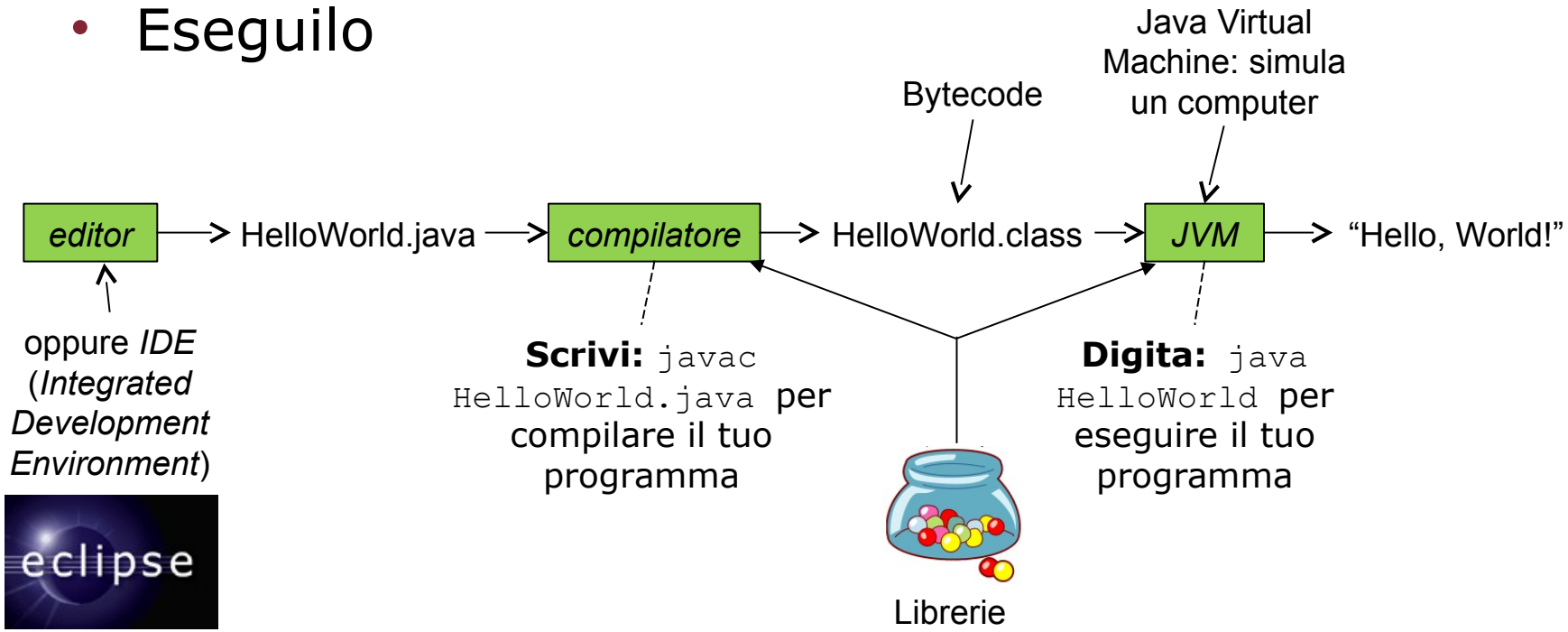
- Il programma (**meglio**: la classe) Java risiede in un file che ha lo stesso nome della classe creata (HelloWorld) più l'estensione .java (HelloWorld.java)
- La prima istruzione chiama **System.out.print** per stampare la stringa "Hello, World!"
- La seconda istruzione chiama **System.out.println** per terminare la linea (= andare a capo)

Sequenza

- Le istruzioni sono raggruppate in un **corpo**, racchiuso da parentesi graffe
- Ogni istruzione termina con un punto e virgola
- **Anche** l'ultima istruzione

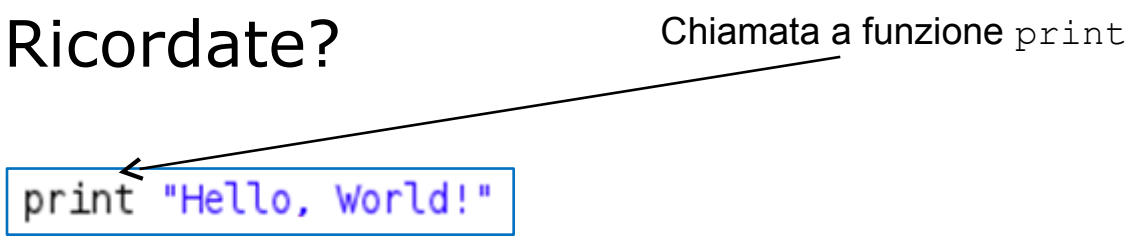
Il primo programma in Java in 3 passi

- Crea un programma
- Compilalo
- Eseguilo



Hello World! In Python

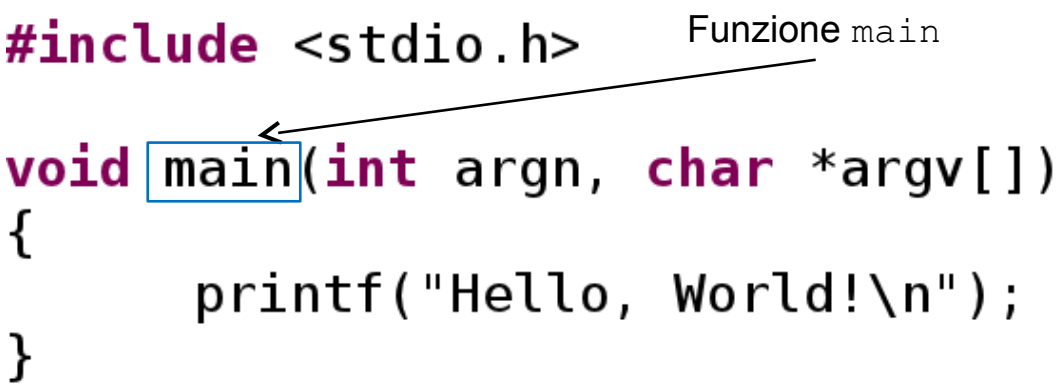
- Ricordate?



- Programma eseguito mediante interprete

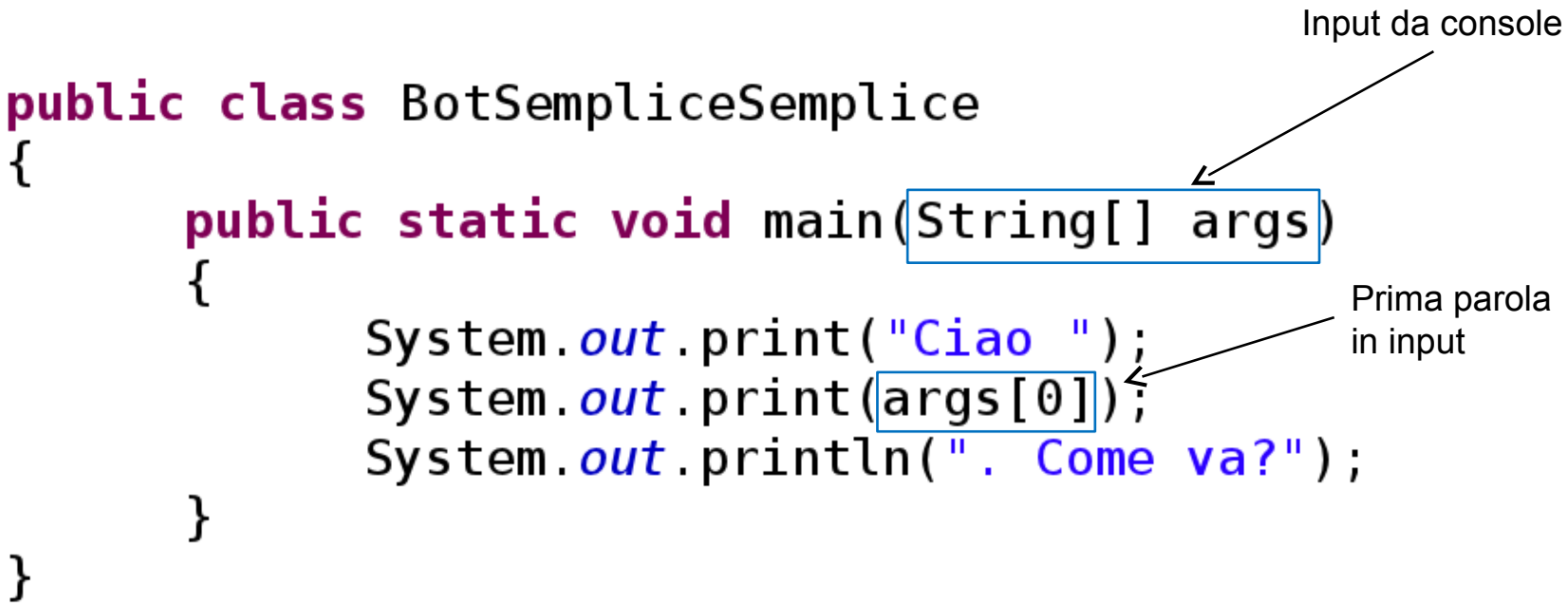
Hello World! in C

- Ricordate?



- Compila, linka ed esegui

Ricevere un input ed emettere un output

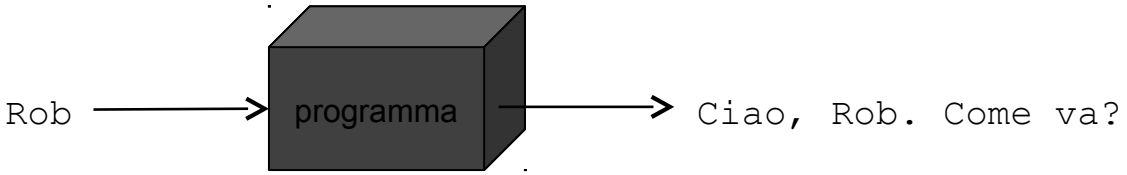



```
public class BotSempliceSemplice
{
	public static void main(String[] args)
	{
		System.out.print("Ciao ");
		System.out.print(args[0]);
		System.out.println(". Come va?");
	}
}
```

- Compila: `javac BotSempliceSemplice.java`
- Esegui: `java BotSempliceSemplice Rob`
- Output: `Ciao Rob. Come va?`

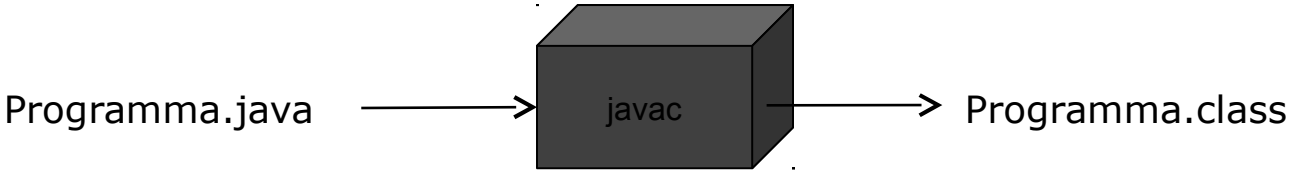
Il vostro programma dal punto di vista dell'utente

- Abbiamo implementato un programma che prende una stringa in input e ne restituisce un'altra in output



Anche javac è un programma

- Javac prende in input una classe .java (una stringa di testo) e ne restituisce la versione "compilata" .class (una stringa di testo) in bytecode



Coding Horror: Anarchia vs. Regole di Stile

- Posso scrivere HelloWorld.java in questo modo?

```
public            class
          HelloWorldAnarchico
{ public static void main(String

            [] args){        System.out.print("Hello, World!");
                                System.out.println();     }
             }
```

- Potenzialmente Sì: non è una questione di spazi e di "a capo"
- Decisamente No! il codice è scritto per essere **riletto e riusato** da altri (e da noi stessi) dopo qualche settimana, mese, anno…

Due stili altrettanto validi

- Stile simmetrico

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

- Stile Kernighan & Ritchie (K&R)

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

printf

- Potete usare System.out.printf() per formattare l'output

formato	descrizione
%b	stampa il valore booleano
%s	stampa una stringa
%d	stampa un intero
%x	stampa il valore intero in esadecimale
%f	stampa il valore floating point
%e	stampa il valore floating point in notazione scientifica
%n	separatore di linea (dipende dal sistema operativo)

Esercizi

- Scrivere una classe Java chiamata PrimoProgramma che stampi a video la stringa "Primo Programma", andando poi a capo
- Scrivere una classe Java chiamata SecondoProgramma che stampi a video la stringa "Hai digitato: " seguita dal primo parametro in input (args[0]). Si utilizzino i metodi print e println
- Svolgere nuovamente il secondo esercizio utilizzando il metodo printf