

# Programmazione 3 e Laboratorio di Programmazione 3

**Primo approccio  
alla programmazione  
Object Oriented**

Proff. Angelo Ciaramella – Emanuel Di Nardo

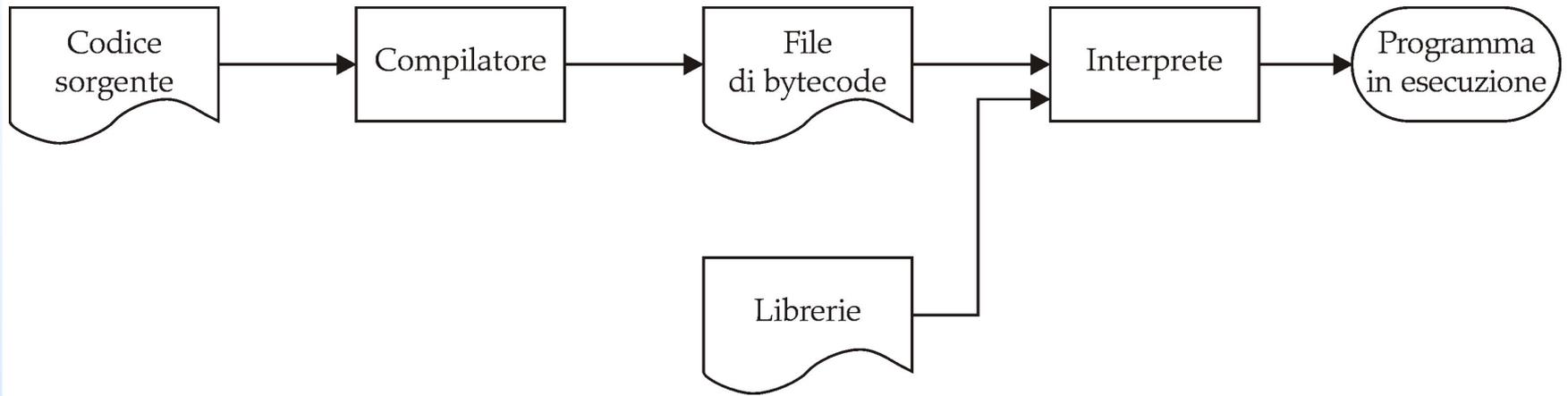
# Primo approccio al codice

---

- Diamo uno sguardo alla classica applicazione “Hello World!”
  - familiarizzare con la **sintassi**
  - concetto di **classe** e di **metodo**
- Vedremo da **linea di comando**
  - come compilare
  - come mandare in esecuzione



# Compilatore-Interprete



Uso del compilatore e dell'interpret in Java



# Linea di comando

- Tramite un **editor di testo** scriviamo il file

```
file.java
```

- **Compiliamo**

```
javac file.java
```

- Se è andato a buon fine otteniamo

- `file.class`

- **Eseguiamo** con il comando

```
java file
```



# Editor Visual Studio Code



The screenshot shows the Visual Studio Code editor interface. The title bar at the top reads "HelloWorld.java". The editor window displays the following Java code:

```
1  public class HelloWorld {  
    Run | Debug  
2      public static void main(String[] args) {  
3          System.out.println(x: "Hello World!");  
4      }  
5  }  
6
```

The status bar at the bottom of the editor shows the following information: "Spazi: 4", "UTF-8", "LF", "Java", "9m", and "Flow".

# Creazione file

---

- Creiamo un file

```
HelloWorld.java
```

- Ciascun file sorgente
  - Deve corrispondere al nome del file che contiene la classe
  - Può contenere al massimo una classe pubblica



# Esecuzione

- Eseguiamo il file

```
java HelloWorld
```

- Risultato linea di comando

```
Hello World!
```

- Risultato **Visual Studio Code**

```
• → java HelloWorld  
Hello World!
```



# Caratteristiche

- Java **distingue** fra maiuscolo e minuscolo (`MAIN` o `PrintLn` errati)
- Java ha una disposizione a formato libero, ma è meglio impaginare i programmi in modo che siano facili da leggere
- `//` e `/* */` permettono di inserire dei commenti per rendere il codice più comprensibile
- Usare le **sequenze di escape** nel testo (esempio `\` , `\n`) per inserire caratteri particolari nelle stringhe
- Ciascun enunciato deve terminare con un punto e virgola (`;`)



# Struttura del file HelloWorld.java

---

nuova classe

```
public class HelloWorld {  
    ...  
}
```



# Struttura del file HelloWorld.java

La classe è pubblica

```
public class HelloWorld {  
    ...  
}
```

Ciascun file sorgente può contenere *al massimo una classe pubblica* il cui *nome corrisponde al nome del file* che contiene la classe



# "Hello World"

Il **metodo** main non agisce su un oggetto (sempre **static**)

**argomenti** della linea di comando

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

ciascuna applicazione deve avere un metodo **main**



# "Hello World"

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

informazione

classe

oggetto nella  
classe System

metodo