

# **Progettazione del Software**

**Giuseppe De Giacomo, Paolo Liberatore,  
Massimo Mecella**

## **Esercitazione 1**



# Macchina Virtuale: Virtual Box

- La macchina virtuale è un software che attraverso un processo di virtualizzazione crea un ambiente virtuale che emula il comportamento di una macchina fisica
- Sui PC è a disposizione la macchina virtuale (distro Linux Ubuntu based) che sarà l'ambiente di riferimento per esercitazioni ed esami
- Copia della macchina sarà scaricabile da un'apposita pagina (comunicata tramite il Forum)



# Sommario

- Programma Java
- Ciclo di compilazione in Java
- IDE
  - Esempio Eclipse
  - Esempio NetBeans
- Dispiegamento di programmi in Java (jar file)
- Classpath

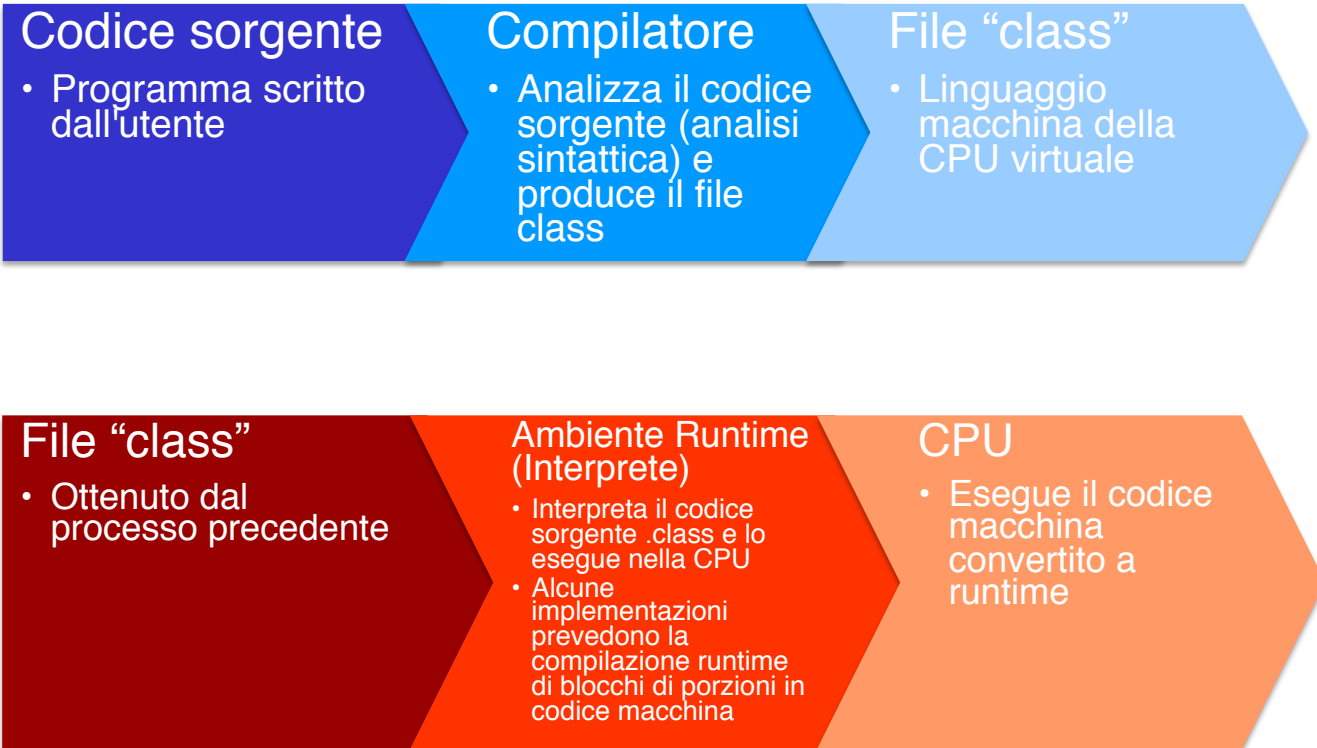


# Programma Java

- Viene scritto il **codice sorgente** in accordo alla sintassi propria del linguaggio
- Il codice sorgente viene analizzato dal compilatore del linguaggio il quale produce dei file con estensione `.class` contenente bytecode.
- Il bytecode prodotto viene interpretato dall'interprete Java ed il file specificato viene eseguito (deve contenere un metodo `main()` ).  
Eventuali altri file `.class` necessari vengono caricati a runtime (collegamento dinamico)

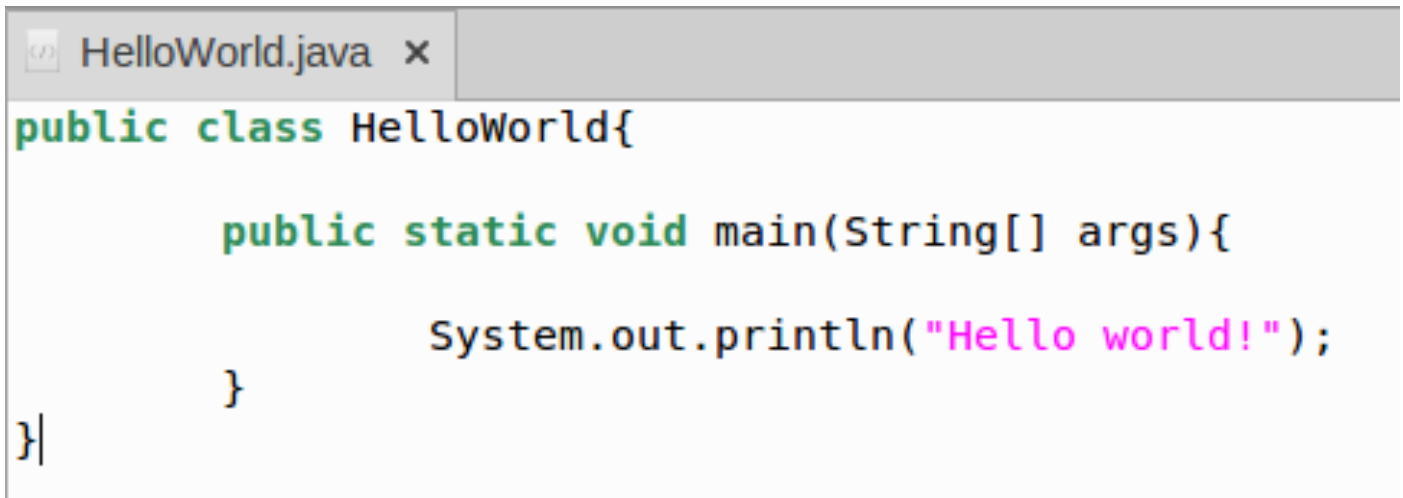


# Programma Java



# Programma Java Esempio

Consideriamo il classico programma HelloWorld:  
Scriviamo il codice sorgente in un file di testo che chiamiamo *'HelloWorld.java'*



```

HelloWorld.java x
public class HelloWorld{

    public static void main(String[] args){

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

    }

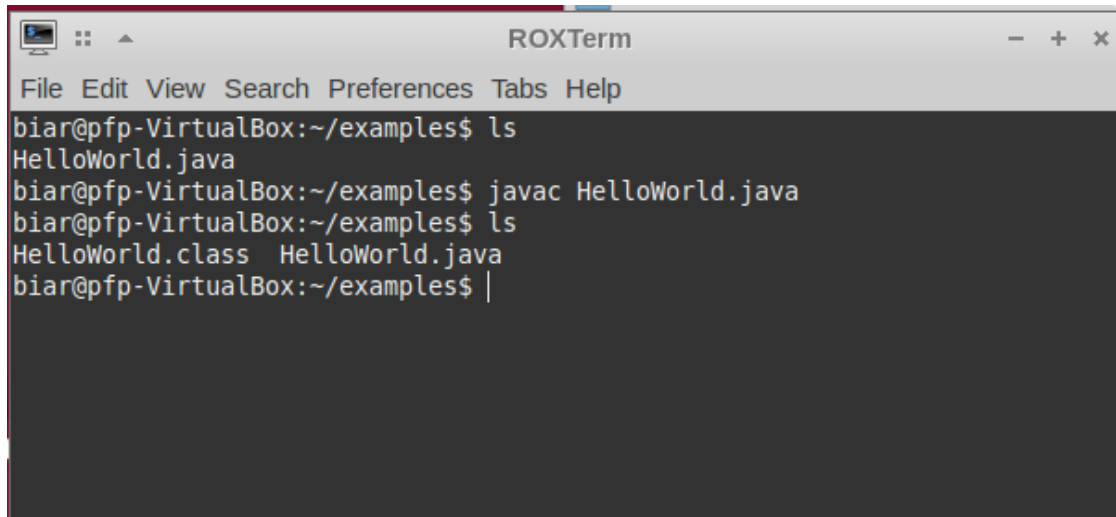
}
```



# Programma Java Esempio

Per compilare il programma, posizionarsi da terminale nella directory in cui è contenuto il file ed eseguire il comando:

*javac HelloWorld.java*



```
ROXTerm
File Edit View Search Preferences Tabs Help
biar@pfp-VirtualBox:~/examples$ ls
HelloWorld.java
biar@pfp-VirtualBox:~/examples$ javac HelloWorld.java
biar@pfp-VirtualBox:~/examples$ ls
HelloWorld.class HelloWorld.java
biar@pfp-VirtualBox:~/examples$ |
```

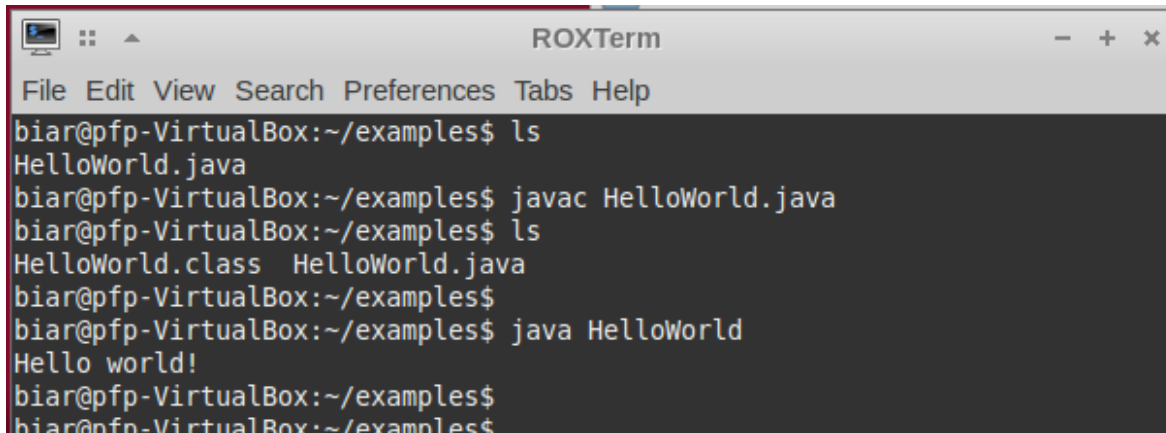
In questo modo viene prodotto il file *HelloWorld.class*



# Programma Java Esempio

Ora si può eseguire il programma con il comando:

*java HelloWorld*



```
ROXTerm
File Edit View Search Preferences Tabs Help
biar@pfp-VirtualBox:~/examples$ ls
HelloWorld.java
biar@pfp-VirtualBox:~/examples$ javac HelloWorld.java
biar@pfp-VirtualBox:~/examples$ ls
HelloWorld.class HelloWorld.java
biar@pfp-VirtualBox:~/examples$
biar@pfp-VirtualBox:~/examples$ java HelloWorld
Hello world!
biar@pfp-VirtualBox:~/examples$
biar@pfp-VirtualBox:~/examples$
```

Nota: per compilare ed eseguire programmi Java devono essere installati JDK e JRE (già installati nella VM del laboratorio)





# IDE

- Per scrivere il programma precedente è stato utilizzato un classico editor di testo (gedit).
- Solitamente per programmi più complessi si usano particolari strumenti più complessi chiamati Integrated Development Environment (IDE). Gli IDE permettono una gestione più completa del codice e delle varie classi, e segnalano in tempo reale eventuali errori di compilazione.
- Esempi di IDE Sono Eclipse e NetBeans (entrambi installati nella VM)



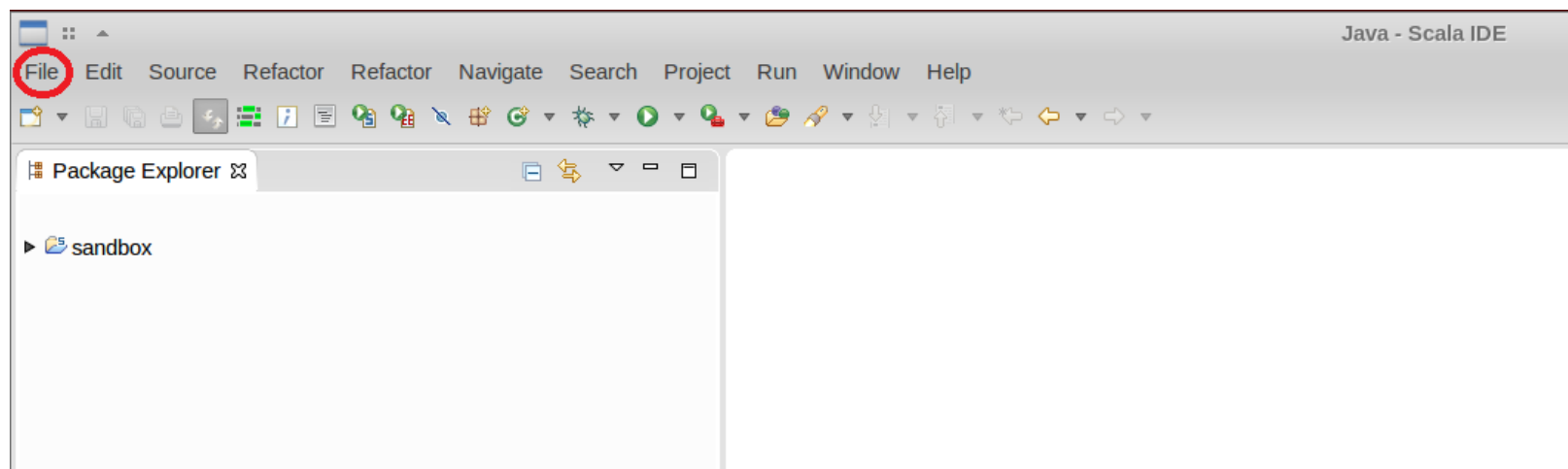
# IDE

- In particolare ogni IDE deve avere più componenti:
- un editor per il codice sorgente;
- un compilatore e un interprete;
- un debugger.



# IDE Esempio: Eclipse

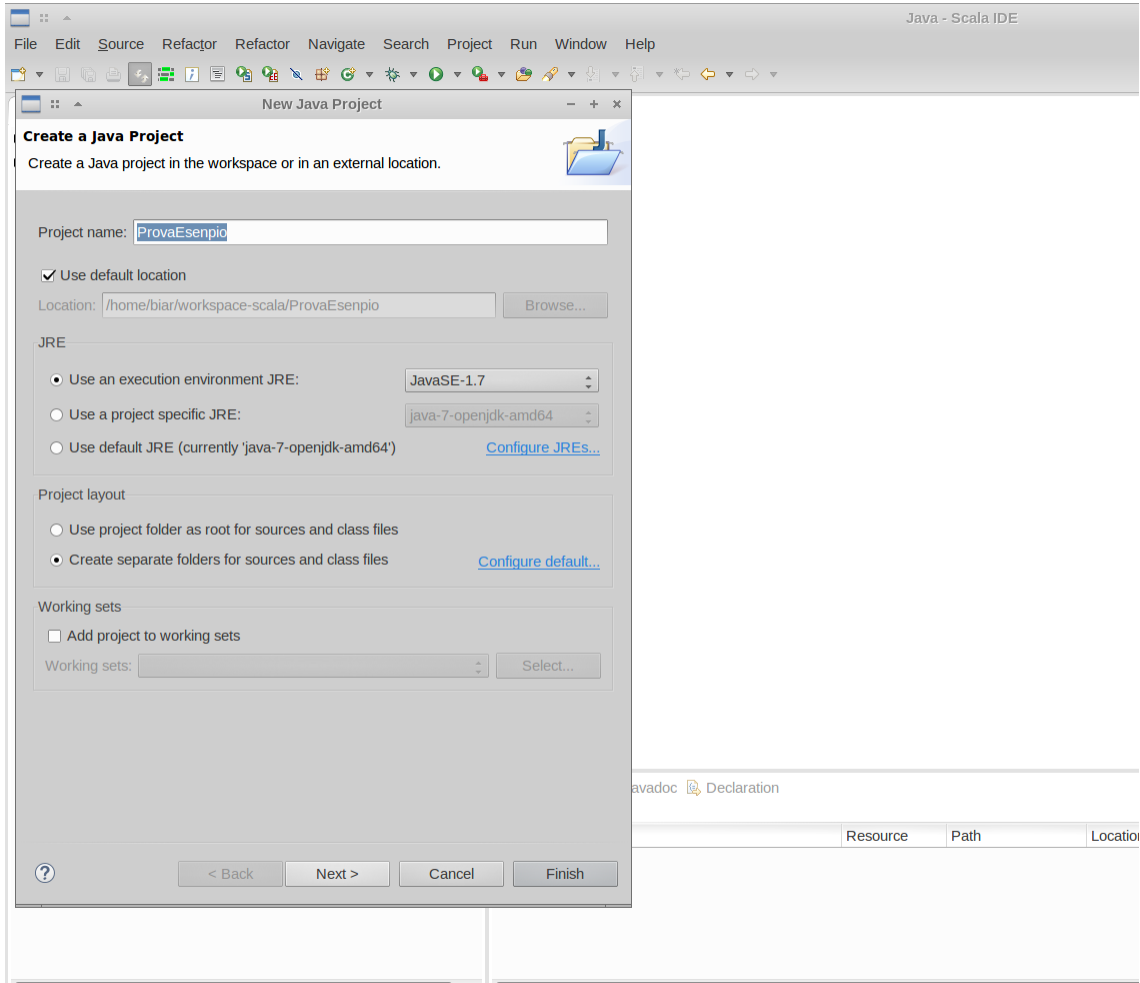
In Eclipse tutti gli elementi di un programma (codice sorgente, file .class, librerie di sistema ecc...) sono contenuti in un Progetto. Per creare un nuovo progetto vuoto:



File → New → Java Project

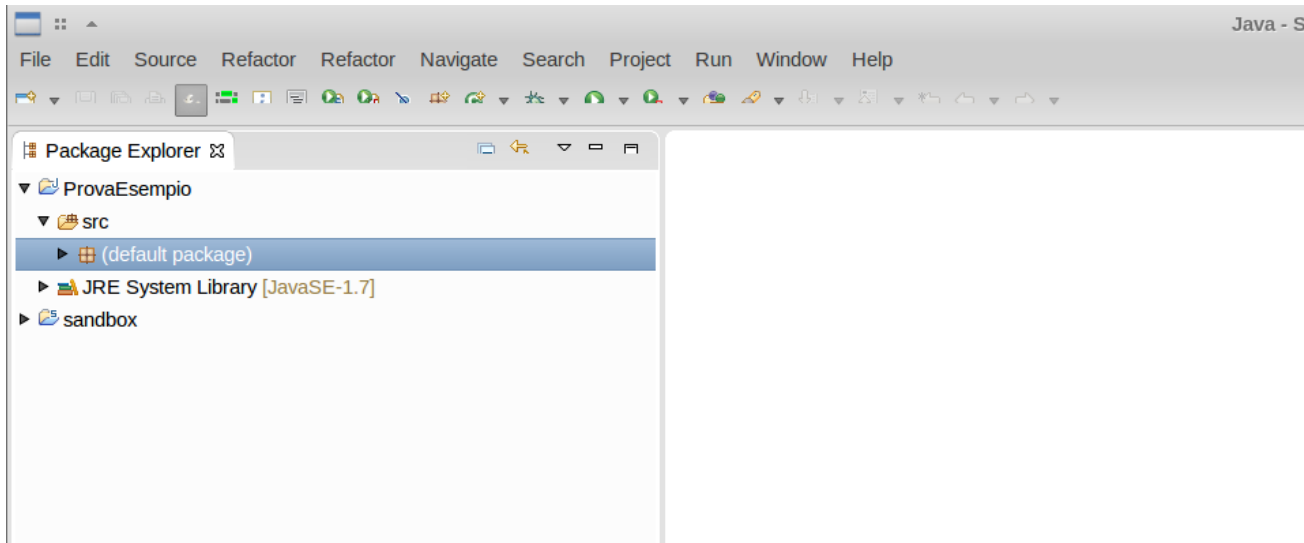


# IDE Esempio: Eclipse



# IDE Esempio: Eclipse

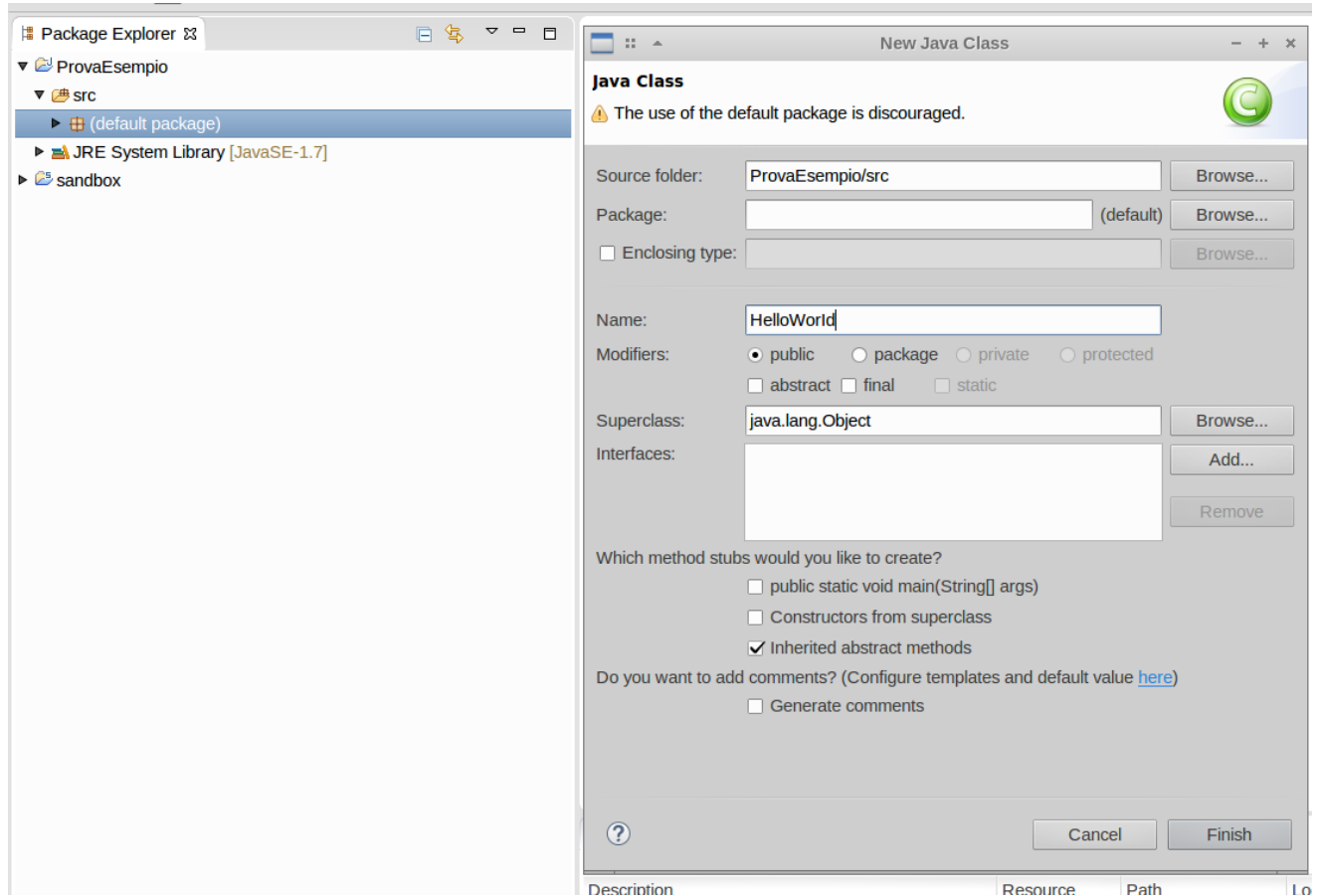
Una volta creato il progetto, possiamo creare classi e packages. Al momento, creiamo le classi nel package di default:



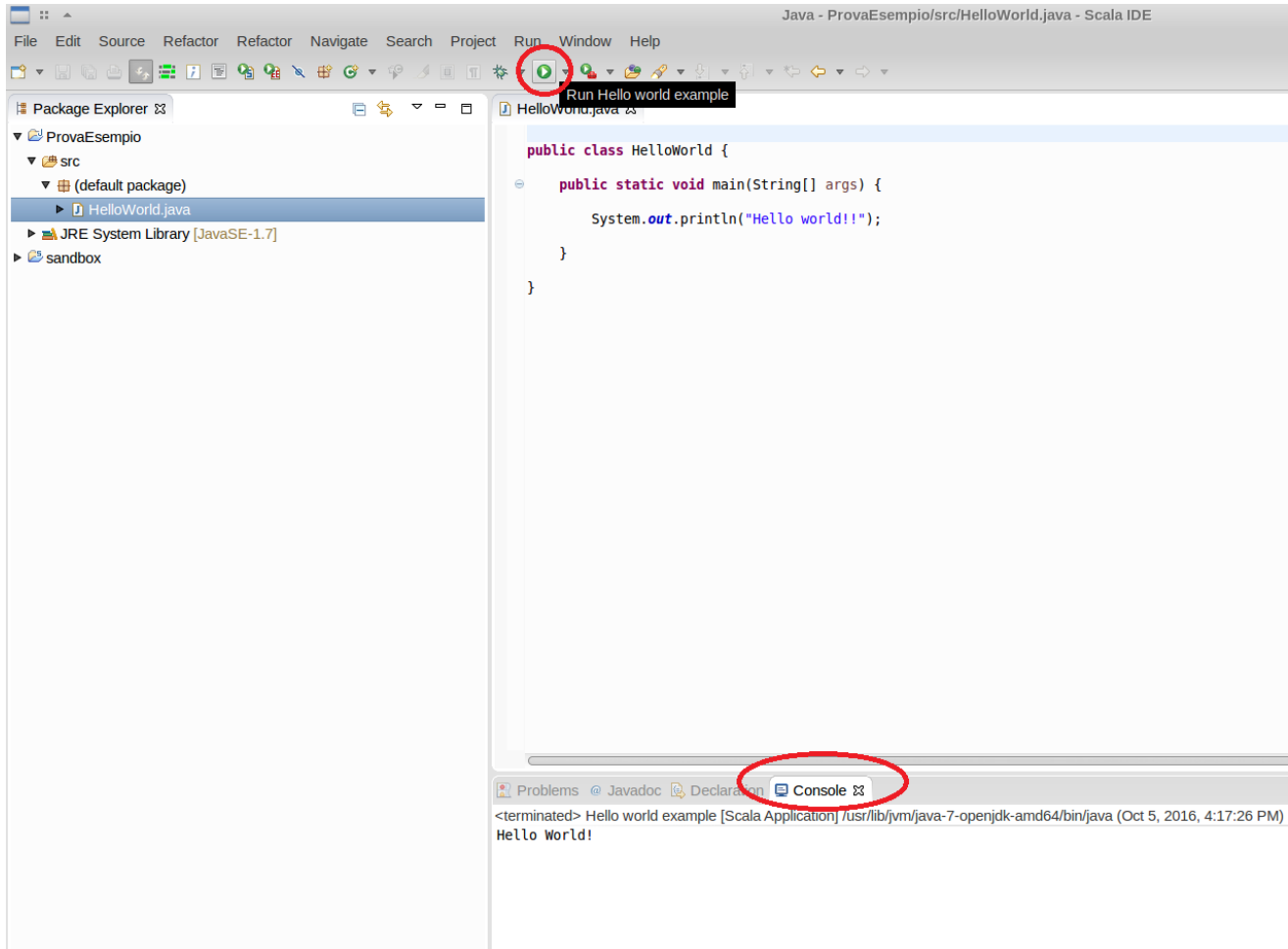
Da ProvaEsempio → src → (default package) col tasto destro selezionare New → Class



# IDE Esempio: Eclipse

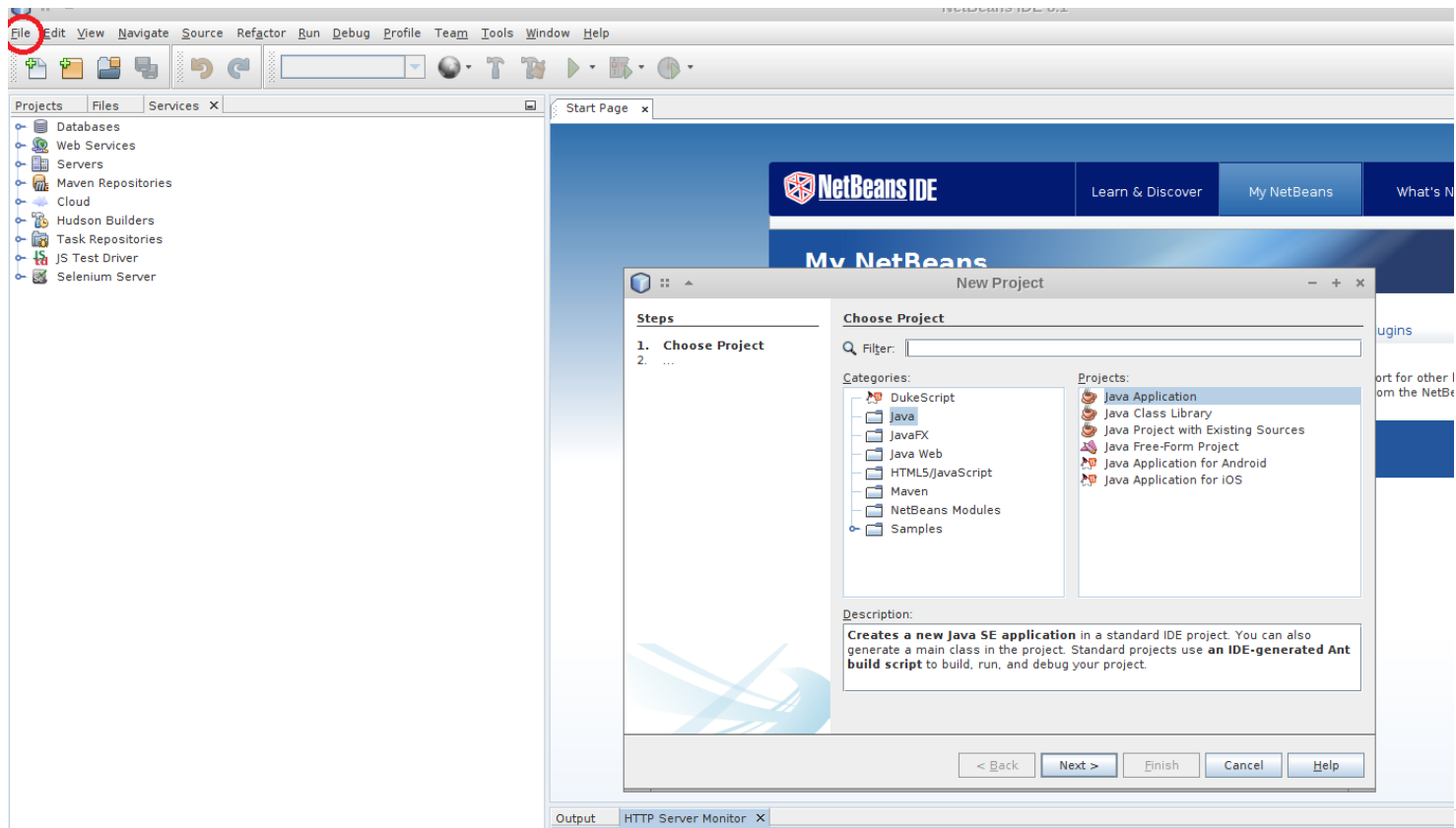


# IDE Esempio: Eclipse



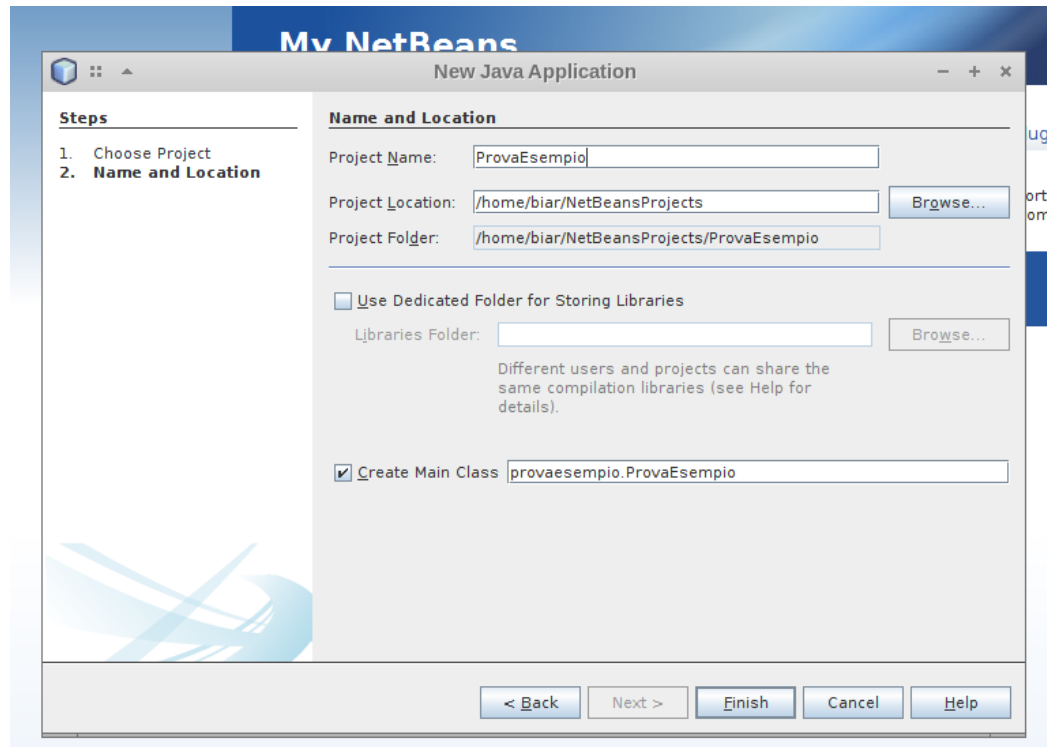
# IDE Esempio: NetBeans

Ora riproponiamo lo stesso esempio utilizzando NetBeans:



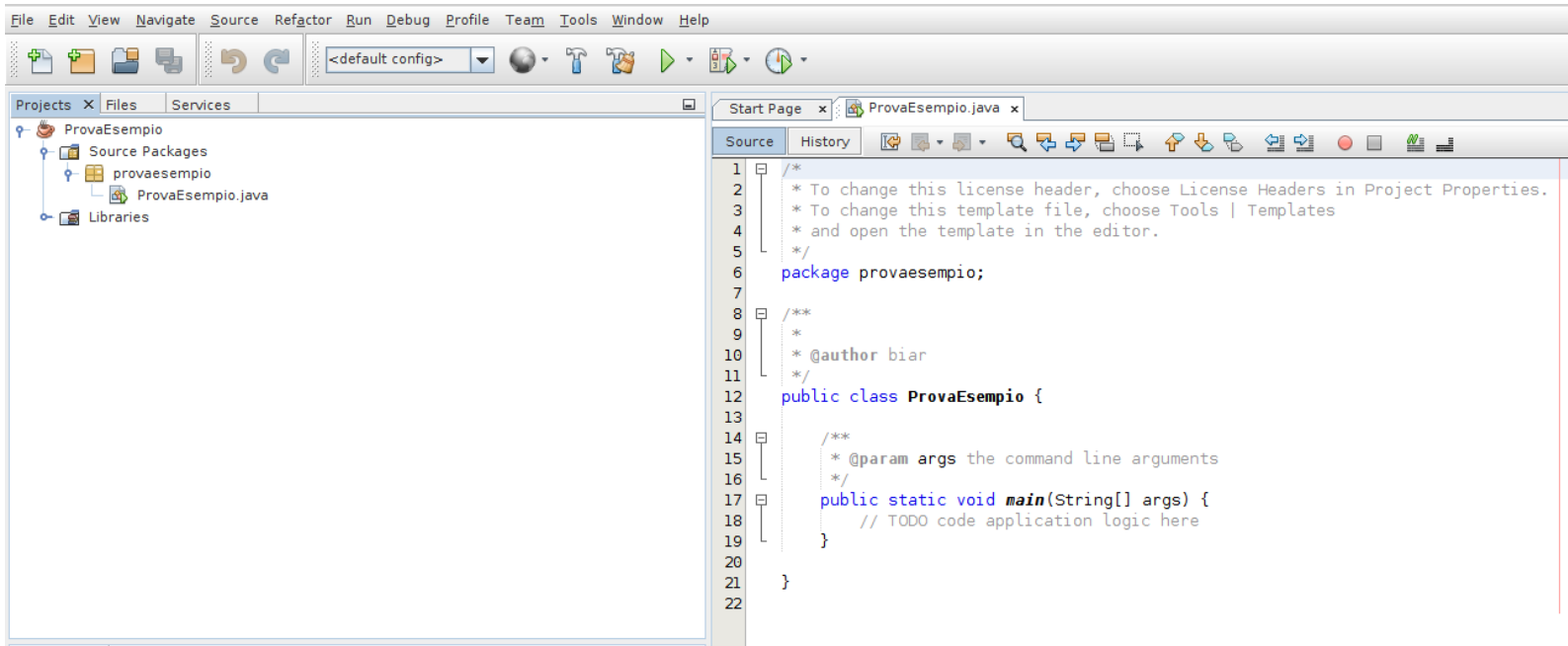


# IDE Esempio: NetBeans

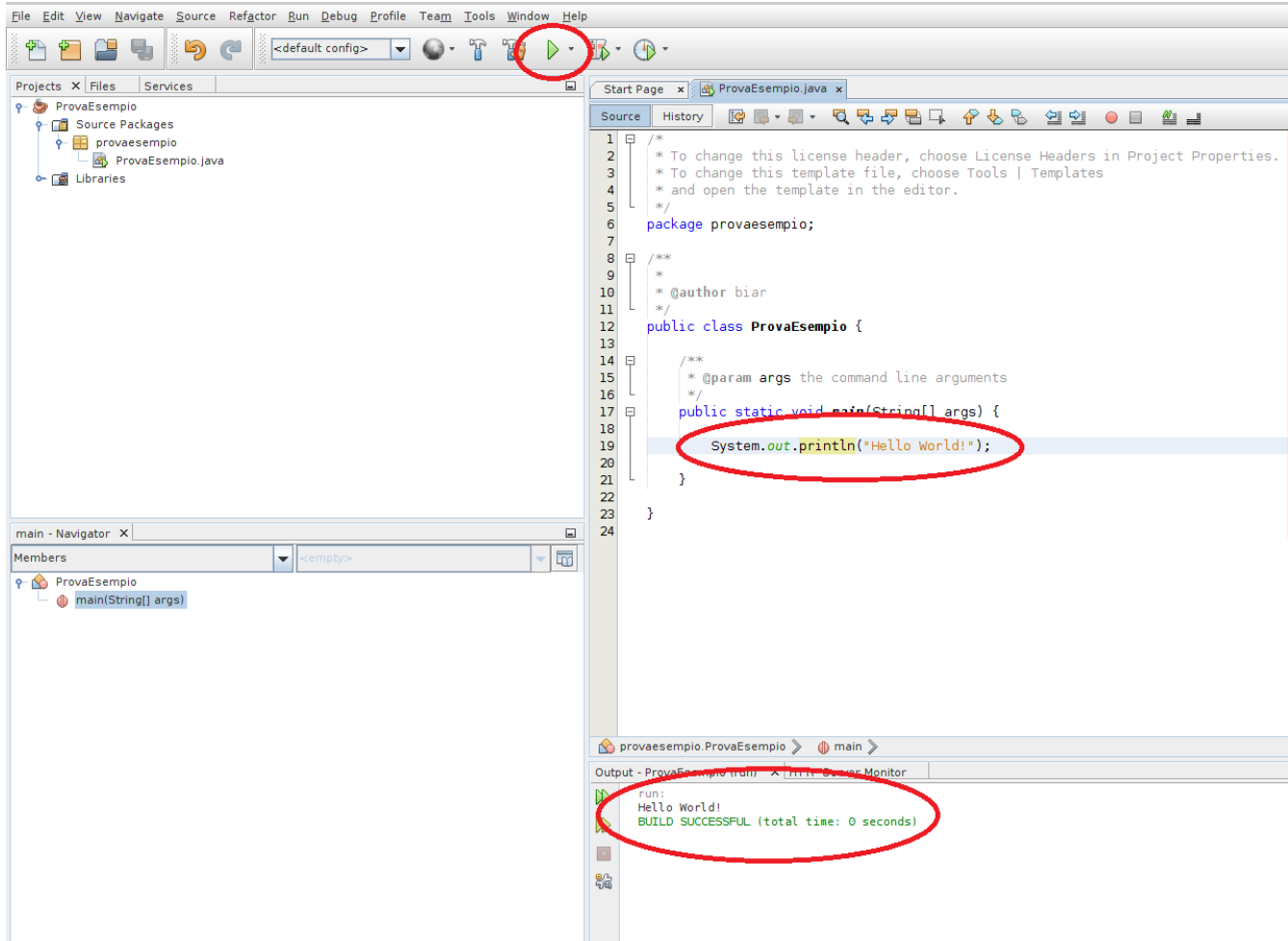


# IDE Esempio: NetBeans

NetBeans a differenza di Eclipse crea direttamente la classe main in un package diverso da quello di default



# IDE Esempio: NetBeans



# Dispiegare programmi in Java

- Un programma Java viene distribuito come un file Java Archive – `jar`
- Può essere creato da riga di comando con

```
jar cf <nomeArchivio> <file/package>
```
- Può essere creato dall'IDE
  - In Eclipse si fa l'export del progetto (funzionalità accessibile tramite tasto destro) avendo cura, nelle varie finestre del wizard, di indicare la classe con il main, in modo che venga correttamente creato il MANIFEST
  - In Netbeans si fa build&deploy (funzionalità accessibile tramite tasto destro)



# Classpath

- In Java il classpath è un parametro di configurazione della JVM che dice dove andare a cercare le classi
- Cf.  
[https://en.wikipedia.org/wiki/Classpath\\_\(Java\)](https://en.wikipedia.org/wiki/Classpath_(Java)) per tutta una serie di *technicalities* ed esempi

