

# Programmazione ad oggetti

## Installazione JDK

---

A.A. 2022/2023

---

*Docente: Prof. Salvatore D'Angelo*  
*Email: [salvatore.dangelo@unicampania.it](mailto:salvatore.dangelo@unicampania.it)*



Università  
degli Studi  
della Campania  
*Luigi Vanvitelli*

*Dipartimento di Ingegneria*

# Argomenti affrontati

Nella lezione si descriveranno brevemente le procedure necessarie per l'installazione del kit di sviluppo per Java sotto i sistemi operativi Windows. Si vedrà inoltre come scrivere e compilare un semplice programma Java.

# Installazione del JDK sul disco fisso

Ricerca o download dell'eseguibile **J2SDK**

Doppio click sull'icona del file

A questo punto partirà la procedura automatica d'installazione. Seguire le istruzioni a video.

Java SE  
Java EE  
Java ME  
Java SE Advanced & Suite  
Java Embedded  
Java DB  
Web Tier  
Java Card  
Java TV  
New to Java  
Community  
Java Magazine

[Overview](#)[Downloads](#)[Documentation](#)[Community](#)[Technologies](#)[Training](#)

## Java SE Development Kit 8 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, applets, and components using the Java programming language.

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

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 8u144 [checksum](#)

## Java SE Development Kit 8u144

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	<a href="#">jdk-8u144-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u144-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	164.65 MB	<a href="#">jdk-8u144-linux-i586.rpm</a>
Linux x86	179.44 MB	<a href="#">jdk-8u144-linux-i586.tar.gz</a>
Linux x64	162.1 MB	<a href="#">jdk-8u144-linux-x64.rpm</a>
Linux x64	176.92 MB	<a href="#">jdk-8u144-linux-x64.tar.gz</a>
Mac OS X	226.6 MB	<a href="#">jdk-8u144-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.87 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.18 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.gz</a>
Solaris x64	140.51 MB	<a href="#">jdk-8u144-solaris-x64.tar.Z</a>
Solaris x64	96.99 MB	<a href="#">jdk-8u144-solaris-x64.tar.gz</a>
Windows x86	190.94 MB	<a href="#">jdk-8u144-windows-i586.exe</a>
Windows x64	197.78 MB	<a href="#">jdk-8u144-windows-x64.exe</a>

### Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

### Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

# Installazione del JDK sul disco fisso

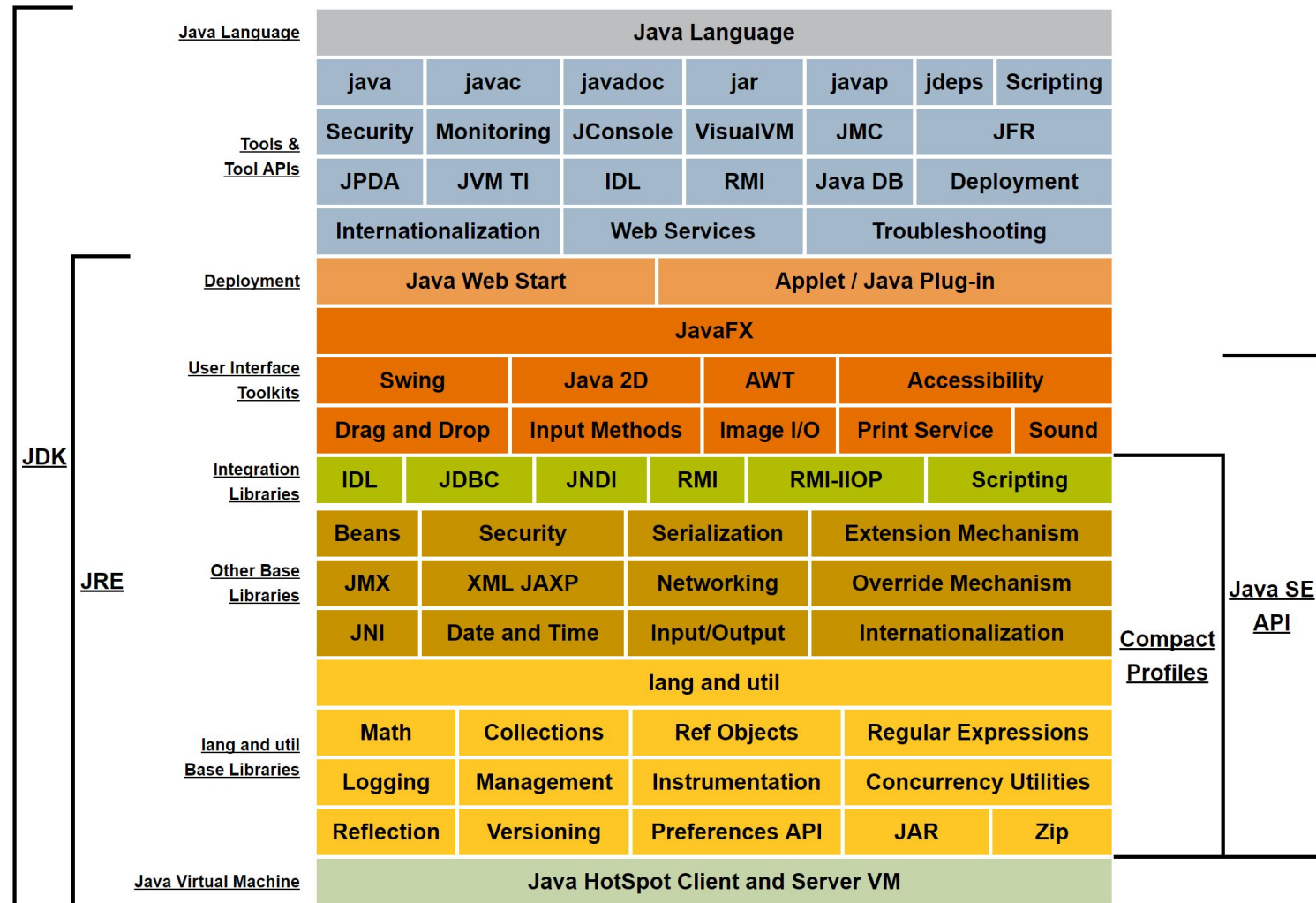
Se l'installazione va a termine correttamente sul disco fisso sarà presente la directory **jdkx.x.x** con le seguenti sotto directory:

- **bin**, contente il compilatore e gli altri eseguibili;
- **lib**, etc.

**JRE**: contiene l'interprete, estensioni, ed altre informazioni di configurazione

**docs**: contiene la documentazione (*N.B.: da scaricare separatamente*)

# Java Platform Standard Edition 8



# PATH e CLASSPATH

Cosa è una variabile di ambiente:

- Una variabile del sistema operativo e visibile da tutte le applicazioni
- Può essere settata temporaneamente in una finestra
- Può essere settata per l'intero sistema

**PATH:** contiene la lista delle directory dove il sistema operativo cerca i comandi da eseguire

**CLASSPATH:** contiene la lista delle directory dove il compilatore java e l'interprete cercano le classi

**JAVA\_HOME:** per far sapere ad applicazioni che utilizzano JAVA dove si trova la cartella di Java

# Aggiornamento delle variabili di ambiente

Prima di poter scrivere ed eseguire un programma Java è necessario aggiornare le variabili di ambiente per permettere al sistema operativo di trovare il compilatore Java e le classi di sistema.

Bisogna aggiornare quindi il PATH e aggiungere una variabile di ambiente detta CLASSPATH



## 2.1- Aggiunta nel Path di Jdk\bin

La procedura è leggermente diversa a seconda del sistema operativo adoperato:

Sotto Windows 98 si deve inserire nella variabile path dell'*AUTOEXEC.BAT* il percorso “*drive*”: *\jdk1.XX\bin*. Ad esempio la linea

*set path= C:\WINDOWS;C:\WINDOWS\COMMAND;*

diventa:

*Set  
path=C:\WINDOWS;C:\WINDOWS\COMMAND;c:\jdk1.X  
X\bin*

# Aggiunta nel Path di Jdk\bin

Sotto Windows XP/2000/Vista/Seven si deve aprire il *control panel*, selezionare *System* (o *Sistema* nella versione Italiana), quindi *Environment* (o *Ambiente*). Scorrere la lista fino a trovare la variabile chiamata PATH e aggiungere il testo “*drive*”\jdk1.XX\bin all’inizio o alla fine della linea.

Dove “drive” è la cartella del disco su cui si è installato il Jdk

Attenzione il Sistema Operativo cerca in Ordine



# Aggiunta della variabile ClassPath

La procedura è identica al caso precedente, con la differenza che la variabile d'ambiente deve essere aggiunta in quanto non presente in genere fra quelle standard.

Sotto Win98 quindi bisogna aprire l'AUTOEXEC.BAT e aggiungere la linea:  
*SET CLASSPATH=.;c:\Nome\_dir\_lavoro\*

# Aggiunta della variabile ClassPath

Sotto WinXP/2000/Vista/Seven  
bisogna aprire il pannello di controllo,  
quindi aprire l'icona *System* e  
selezionare la scheda *Environment*. Nel  
campo Variabile scrivere:

***CLASSPATH***

Nel campo valore:

***.;c:\Nome\_dir\_lavoro\***

# Aggiunta della variabile ClassPath

Con il punto s'indica al compilatore Java di cercare le eventuali altre classi nella stessa directory dove si trova il file sorgente.

La stringa “Directory Lavoro” si dovrà sostituire con il nome dato alla directory che conterrà i file .class (mentre inserire il punto nel ClassPath è necessario, indicare una directory di lavoro è opzionale)

# Scrittura del programma di esempio

Il modo più semplice per editare un file sorgente Java è quello di adoperare un programma di scrittura come Notepad o WordPad di Windows.

Supponiamo di voler scrivere il seguente programma Java, che stampa a video un messaggio di saluto:

# Testo del primo programma

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

# Generazione del bytecode

Una volta scritto con un editor il testo si salvi il file con il nome HelloWorld.java (in Java è obbligatorio chiamare il file sorgente con lo stesso nome della classe che contiene il main).

N.B.: *Visualizza estensioni*

Aprire una finestra DOS, posizionarsi nella directory dove si è salvato il file sorgente e lanciare il “*Compilatore*” (in realtà non è un vero compilatore in quanto non genera codice oggetto ma bytecode).



# Generazione del bytecode

Il comando da eseguire è:

```
javac HelloWorld.java
```

Si ricorda che Java distingue tra maiuscole e minuscole, quindi i nomi *helloWorld*, *helloworld* e *Helloworld* sono tutti e tre errati.

Perche trova javac ?

Perche trova HelloWorld.java?

# 3.3-Esecuzione del bytecode

Per eseguire il file HelloWorld.class generato si deve lanciare la Java Virtual Machine.

Il comando da adoperare sarà:

```
java HelloWorld
```

Si noti che non deve essere presente l'estensione .class

Perche trova java ? Quale Java usa ?

Perche trova HelloWorld ?

Se mi trovassi altrove ?

# Un altro esempio

In questo secondo esempio si adopererà la libreria grafica di Java, creando una dialog box per l'immissione di alcuni dati. Lo scopo del programma è solo quello di mostrare un primo esempio di output grafico, non è necessario comprendere tutte le istruzioni.

Il testo sorgente è:

# Testo secondo esempio

```
import javax.swing.*;

public class InputTest {

    public static void main(String[] args) {
        String nome;
        nome=JOptionPane.showInputDialog ("Come ti chiami?");
        System.out.println("Salve "+nome);
    }
}
```

# Compilazione ed esecuzione del secondo esempio

Il file va salvato con lo stesso nome della classe con in più l'estensione .java (InputTest.java). Per la generazione del bytecode il comando sarà:

```
javac InputTest.java
```

Per l'esecuzione:

```
java InputTest
```

# Esecuzione del programma

Eseguendo il programma compare la seguente finestra di input:



# Esecuzione del programma

Immettendo dei caratteri e premendo OK si otterrà un output su console.

