

# ***INTRODUZIONE A JAVA***

# Linguaggio Java

- **Robusto**
  - Non permette costrutti “pericolosi”
    - Eredità Multipla
    - Gestione della Memoria
- **Orientato agli oggetti**
  - Ogni cosa ha un tipo
  - Ogni tipo è un oggetto (quasi)
- **Protegge e gestisce dagli errori**
  - Verifica gli errori runtime
  - Gestisce esplicitamente gli errori

# Caratteristiche Java - cont.

- **Portabile**

- Astrae l'accesso alle risorse del SO
  - File, network, etc.
- Astrae i tipi di dato
  - int, float, double
- Ricco di librerie standard
  - collezioni, date, random number

- **Debug/Linking dinamico**

- Debug del bytecode
- Il bytecode contiene le informazioni per il linking a runtime

# Java Development Kit

- **bin**: contiene i file eseguibili
  - javac: compilatore
  - java: avvia la JVM ed esegue i programmi
  - jdb: esecuzione in modalità debug
  - javadoc: genera la documentazione
  - ...
- **demo**: esempi (anche complicati) per mostrare le potenzialità di Java
- **include** e **lib**: codice binario usato da JDK
- **jre**: la Java Virtual Machine
- **docs**: documentazione delle librerie java

# Java Virtual Machine

- È un programma installato nel vostro PC
- Crea l'ambiente esecutivo per i programmi
  - Li interfaccia con la macchina
- Avviata con “java”
- Non è legata al linguaggio Java
  - Esegue bytecode
- Il bytecode può essere generato in diversi modi
  - Il principale è javac

# JVM e bytecode

- Il programma compilato è “teoricamente” portabile
  - Non dipende dal SO
  - Mi serve solo la JVM
- JVM astrae i SO
- La JVM non è portabile
  - SO diversi hanno JVM diverse
- Tutte le JVM dovrebbero avere le stesse caratteristiche
  - Spesso non è vero
  - Ecco il perché del “teoricamente”

# Linguaggio ed esecuzione

- Java
  - Linguaggio (è uno solo)
- Diverse piattaforme
  - J2SE (Standard Edition)
    - Applicazioni desktop
  - J2EE (Enterprise Edition)
    - Applicazioni server oriented
  - J2ME (Mobile Edition)
    - Applicazioni Mobile
  - ...







# Per le esercitazioni





Installare nel proprio pc i seguenti software:

- Java Platform, Standard Edition
  - Java JDK
- Eclipse IDE for **Java EE Developers**
  - Ambiente integrato di sviluppo
  - "Eclipse IDE for Enterprise Java Developers"
- Editor txt
  - Sublime Text
  - notepad++
  - Fraise
  - ...

# Scaricare la JDK

**Menu**



 Fazer Login  País  Ligar 

Oracle Technology Network / Java / Java SE / Downloads

Java SE  
Soporte para Java SE  
Java Embedded  
Java EE  
Java ME  
Java FX  
Java DB  
Web Tier  
Comunidade

Resumo

Downloads

Documentação

Comunidade

Tecnologia

Formação

## 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 8u201 [checksum](#)  
JDK 8u202 [checksum](#)

Java SE Development Kit 8u201

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	72.98 MB	<a href="#">jdk-8u201-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.92 MB	<a href="#">jdk-8u201-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.98 MB	<a href="#">jdk-8u201-linux-i586.rpm</a>
Linux x86	185.77 MB	<a href="#">jdk-8u201-linux-i586.tar.gz</a>
Linux x64	168.05 MB	<a href="#">jdk-8u201-linux-x64.rpm</a>
Linux x64	182.93 MB	<a href="#">jdk-8u201-linux-x64.tar.gz</a>
Mac OS X x64	245.92 MB	<a href="#">jdk-8u201-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	125.33 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	88.31 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	133.99 MB	<a href="#">jdk-8u201-solaris-x64.tar.Z</a>
Solaris x64	92.16 MB	<a href="#">jdk-8u201-solaris-x64.tar.gz</a>
Windows x86	197.66 MB	<a href="#">jdk-8u201-windows-i586.exe</a>
Windows x64	227.46 MB	<a href="#">jdk-8u201-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](#)

V1

Programmazione Java per Dispositivi Mobili


10

# Scegliere la versione

<div> <div>Java SE Development Kit 8u201</div> <div>You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.</div> <div> <input type="radio"/> Accept License Agreement           <input checked="" type="radio"/> Decline License Agreement         </div> </div>		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.98 MB	<a href="#">jdk-8u201-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.92 MB	<a href="#">jdk-8u201-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.98 MB	<a href="#">jdk-8u201-linux-i586.rpm</a>
Linux x86	185.77 MB	<a href="#">jdk-8u201-linux-i586.tar.gz</a>
Linux x64	168.05 MB	<a href="#">jdk-8u201-linux-x64.rpm</a>
Linux x64	182.93 MB	<a href="#">jdk-8u201-linux-x64.tar.gz</a>
Mac OS X x64	245.92 MB	<a href="#">jdk-8u201-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	125.33 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	88.31 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	133.99 MB	<a href="#">jdk-8u201-solaris-x64.tar.Z</a>
Solaris x64	92.16 MB	<a href="#">jdk-8u201-solaris-x64.tar.gz</a>
Windows x86	197.66 MB	<a href="#">jdk-8u201-windows-i586.exe</a>
Windows x64	207.40 MB	<a href="#">jdk-8u201-windows-x64.exe</a>

[Link](#)

# Scaricare Eclipse


[Members](#)
[Working Groups](#)
[Projects](#)
[More ▾](#)
[Download](#)


[Home](#) / [Downloads](#) / [Packages](#) / [Release](#) / [Eclipse 2018-12](#) / [R](#)

[Eclipse Installer](#)
[Eclipse Packages](#)
[Eclipse Developer Builds ▾](#)

## Try the Eclipse **Installer** 2018-12 R


The easiest way to install and update your Eclipse Development Environment.

[Find out more](#)


 **2,388,519 Downloads**

### Download

Mac OS X **64 bit**  
Windows **64 bit**  
Linux **64 bit**



## Eclipse IDE 2018-12 R Packages



**Eclipse IDE for Enterprise Java Developers**


334 MB 379,211 DOWNLOADS

Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and more.


[Click here](#) to file a bug against Eclipse Web Tools Platform.

[Click here](#) to file a bug against Eclipse Platform.

[Click here](#) to file a bug against Maven integration for web projects.




Windows **64-bit**  
Mac Cocoa **64-bit**  
Linux **64-bit**




**Eclipse IDE for Java Developers**

184 MB 295,639 DOWNLOADS



Windows **64-bit**  
Mac Cocoa **64-bit**  
Linux **64-bit**



## Get **Eclipse IDE 2018-12**

Install your favorite desktop IDE packages.


[Download 64 bit](#)


[Download Packages](#) | [Need Help?](#)

### RELATED LINKS

- [Compare & Combine Packages](#)


# Scaricare Eclipse



MembersWorking GroupsProjectsMore-

Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.




 **Download**


**Download from:** United States - Princeton University Mathematics (<http>)


**File:** `eclipse-jee-2018-12-R-macosx-cocoa-x86_64.dmg` `SHA-512`


[>> Select Another Mirror](#)


**OR** Get It Faster from our Members





**INSTALL, LAUNCH AND SHARE ECLIPSE** 


 TEAM SETTINGS

 PLUGINS

 PROJECTS

 PREFERENCES

 SERVER

 REPOSITORIES

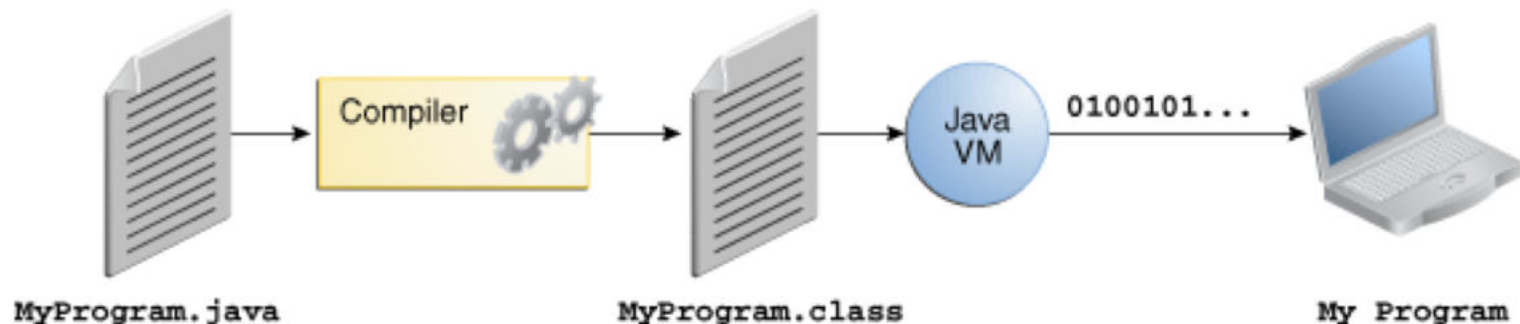
**DOWNLOAD LAUNCHER**

**Other options for this file**

- [All mirrors \(xml\)](#)
- [Direct link to file](#) (download starts immediately from best mirror)

# Il processo di sviluppo

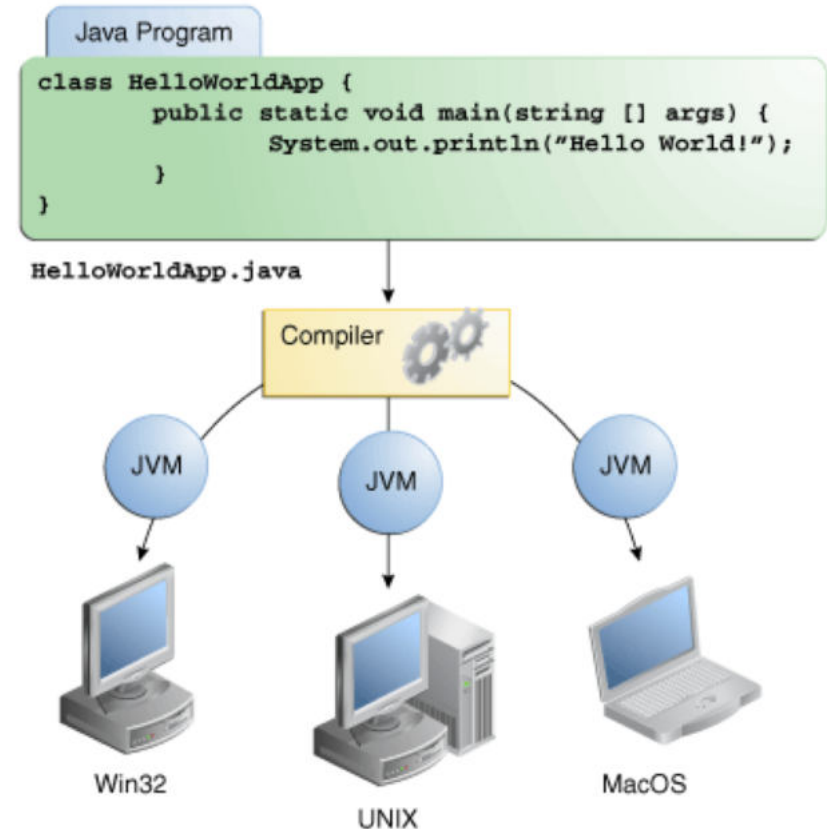
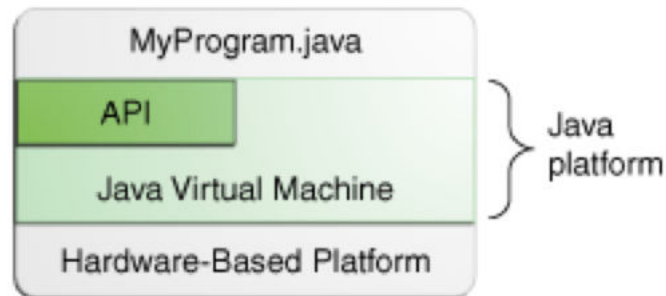
1. Scrittura di un file txt salvato con estensione .java
2. javac esegue la compilazione in bytecode
  - non è codice nativo
  - è il linguaggio macchina della JVM
3. java esegue la JVM che esegue il bytecode





# Portabilità

- Lo stesso bytecode può essere eseguito su più piattaforme
- La JVM isola il programma dall'hardware specifico







# Hello world

```

1 public class HelloWorld
2 {
3     public static void main(String args[])
4     {
5         System.out.println("Hello World!");
6     }
7 }

```

- Riga1: dichiarazione della classe
- Righe 2-7: parentesi graffe che contengono la classe
- Riga 3: dichiarazione del metodo main
- Riga 5: stampa la stringa “Hello World!” usando una libreria Java

## Esercizio 1

digitarlo in notepad e salvarlo come HelloWorld.java  
 compilarlo con javac e eseguirlo con java

## Esercizio 2

Eseguirlo con Eclipse (vedi Tutorial)

HelloWorld
<u>+main(\$: String[]): void</u>



# ***ALGORITMI***

# Algoritmi

- **Definizione:** Una sequenza di passi per risolvere un problema
- **Esempio: "Preparare biscotti"**
  - Mischiare gli ingredienti a secco
  - Amalgamare il burro e lo zucchero
  - Sbattere le uova
  - Mescolare gli ingredienti
  - Impostare la temperatura del forno
  - Impostare il timer
  - Mettere i biscotti nel forno
  - Aspettare che siano cotti
  - Preparare la glassa
  - Ricoprire di glassa i biscotti
  - ...



# Problemi degli algoritmi

- *Manca di una struttura:*
  - *tanti piccoli passi difficili da ricordare*
- *Ridondanza: es. fare due infornate*
  - Mischiare gli ingredienti a secco
  - Amalgamare il burro e lo zucchero
  - Sbattere le uova
  - Mescolare gli ingredienti
  - Impostare la temperatura del forno
  - Impostare il timer
  - Mettere i biscotti nel forno
  - Aspettare che siano cotti
  - Impostare il timer
  - Mettere i biscotti nel forno
  - Aspettare che siano cotti
  - Preparare la glassa
  - Ricoprire di glassa i biscotti

# Algoritmi strutturati

- Divisione in task coerenti

- 1 Fare l'impasto

- Mischiare gli ingredienti a secco
    - Amalgamare il burro e lo zucchero
    - Sbattere le uova
    - Mescolare gli ingredienti

- 2 Cuocere I biscotti

- Impostare la temperatura del forno
    - Impostare il timer
    - Mettere i biscotti nel forno
    - Aspettare che siano cotti

- 3 Decorare I biscotti

- Preparare la glassa
    - Spargere la glassa sui biscotti

- ...

# Rimuovere la ridondanza

- Un algoritmo strutturato descrive task ripetitivi con meno ridondanza

## 1 Fare l'impasto

- Mischiare gli ingredienti a secco

...

## 2 Cuocere I biscotti

- Impostare la temperatura del forno

...

## 3 Cuocere I biscotti

- Impostare la temperatura del forno

...

## 4 Decorare I biscotti

- Preparare la glassa

...

# Algoritmo con ridondanza

```
public class PreparaBiscotti {
    public static void main(String[] args) {
        System.out.println("Mischiare gli ingredienti a secco");
        System.out.println("Amalgamare il burro e lo zucchero");
        System.out.println("Sbattere le uova");
        System.out.println("Mescolare gli ingredienti");

        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");

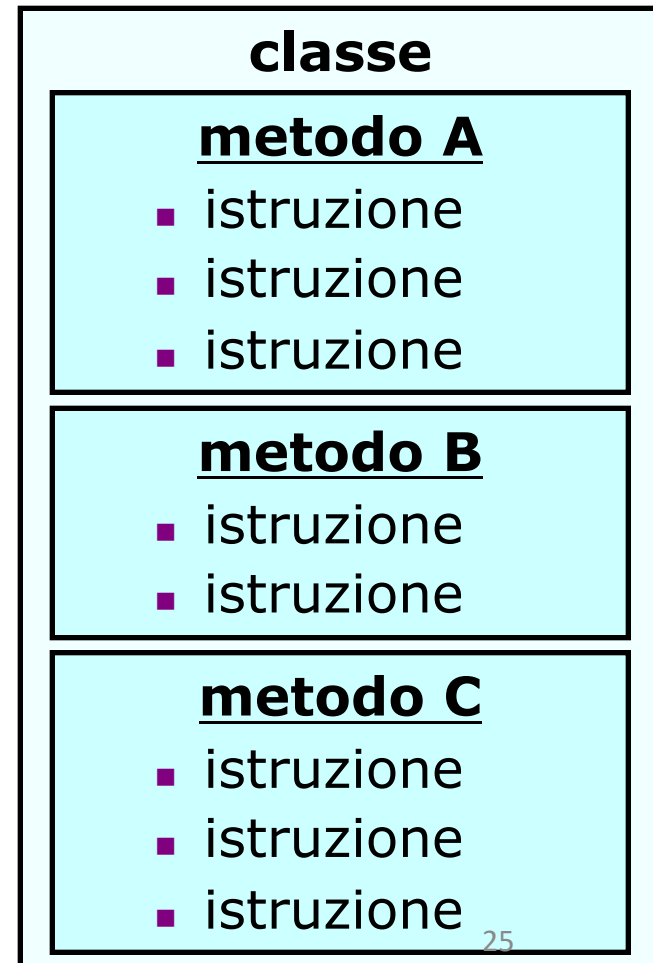
        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");

        System.out.println("Preparare la glassa");
        System.out.println("Ricoprire di glassa i biscotti");
    }
}
```



# Metodi Statici

- Un gruppo di istruzioni con un nome.
  - indica la struttura di un programma
  - elimina la ridondanza riusando il codice
  - **decomposizione procedurale:**
    - divide il problema in metodi
- Creiamo dei nuovi comandi



# Usare metodi statici

## 1. Progettare l'algoritmo.

- Guardare la struttura e vedere i comandi ripetuti
- Decidere quali sono i task importanti

## 2. Dichiarare i metodi

- Organizzare le istruzioni in gruppi e dargli un nome

## 3. Chiamare i metodi

- il main chiama gli altri metodi ...

# Biscotti versione 2

```
public class PreparaBiscotti {
    public static void main(String[] args) {
        // Step 1 - Fare l'impasto
        System.out.println("Mischiare gli ingredienti a secco");
        System.out.println("Amalgamare il burro e lo zucchero");
        System.out.println("Sbattere le uova");
        System.out.println("Mescolare gli ingredienti");

        //Step 2 - Cuocere i biscotti 1
        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");

        //Step 3 - Cuocere i biscotti 2
        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");

        // Step 4 - Decora i biscotti
        System.out.println("Preparare la glassa");
        System.out.println("Ricoprire di glassa i biscotti");
    }
}
```

# Dichiarare dei metodi

- Sintassi:

```
public static void nome() {  
    istruzione;  
    istruzione;  
    ...  
    istruzione;  
}
```

- Sintassi per chiamare il metodo:  
**nome** ( ) ;

# Programma finale

```
public class PreparaBiscotti3 {
    public static void main(String[] args) {
        prepara();
        cuoci();
        decora();
    }

    public static void prepara(){
        System.out.println("Mischiare gli ingredienti a secco");
        System.out.println("Amalgamare il burro e lo zucchero");
        System.out.println("Sbattere le uova");
        System.out.println("Mescolare gli ingredienti");
    }

    public static void cuoci(){
        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");
    }

    public static void decora(){
        System.out.println("Preparare la glassa");
        System.out.println("Ricoprire di glassa i biscotti");
    }
}
```

# Metodi che chiamano metodi

```
public class MethodsExample {
    public static void main(String[] args) {
        message1();
        message2();
        System.out.println("Done with main.");
    }

    public static void message1() {
        System.out.println("This is message1.");
    }

    public static void message2() {
        System.out.println("This is message2.");
        message1();
        System.out.println("Done with message2.");
    }
}
```

- **Output:**

```
This is message1.
This is message2.
This is message1.
Done with message2.
Done with main.
```

# Quando usare i metodi

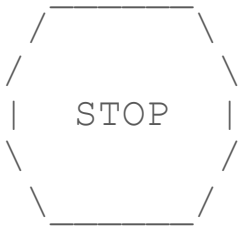
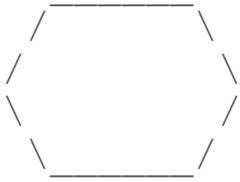
- Inserire delle istruzioni in un metodo statico se:
  - Sono strutturalmente correlati
  - Sono ripetuti
- Non creare dei metodi per
  - Istruzioni singole
  - Istruzioni non correlate o debolmente correlate
    - provare a dividere in metodi più piccoli





# Esercizio 3

- Scrivere un programma che stampi:



Versione 1:

usare metodi statici per strutturare la stampa

Versione 2:

eliminare la ridondanza con i metodi statici

<http://ppl.eln.uniroma2.it/pjdm/es3-output.txt>