

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
 - ...

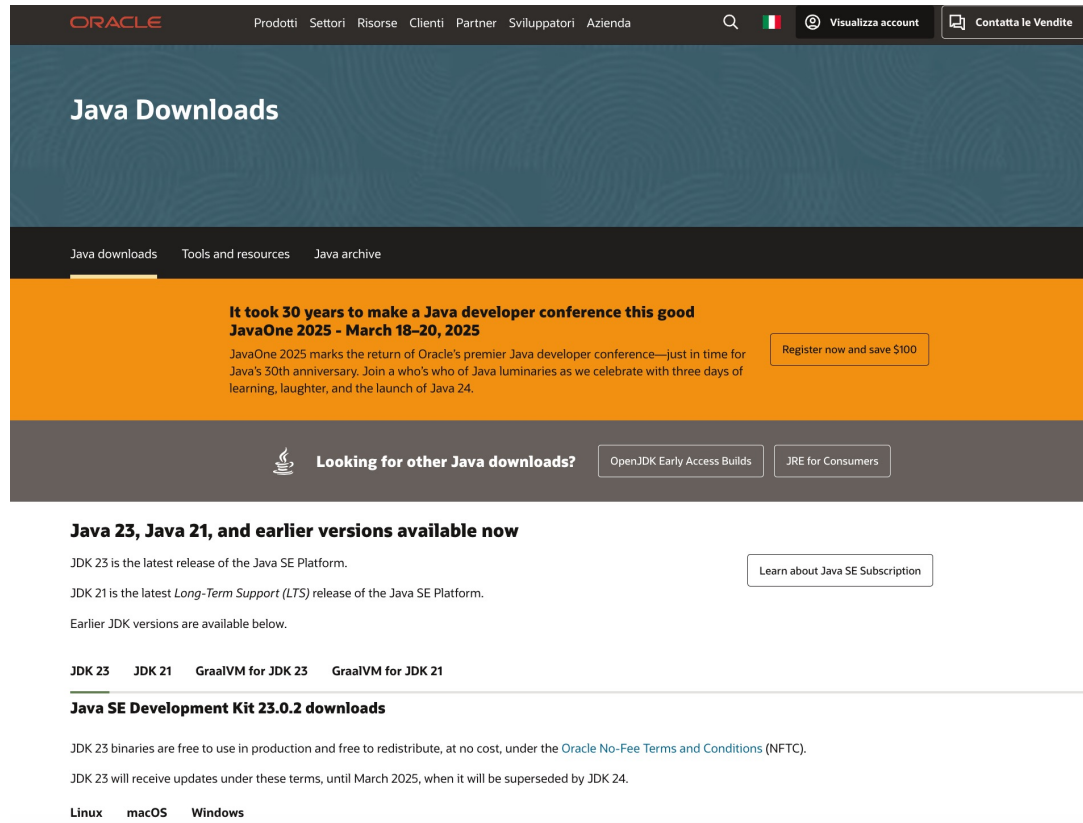
Per le esercitazioni

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

<https://www.oracle.com/it/java/technologies/downloads/>



The screenshot shows the Oracle Java Downloads page. The header includes the Oracle logo, navigation links (Prodotti, Settori, Risorse, Clienti, Partner, Sviluppatori, Azienda), a search icon, a language selector (Italy), and links for account management and sales. The main heading is "Java Downloads". Below this is a navigation bar with "Java downloads", "Tools and resources", and "Java archive". A prominent orange banner promotes "JavaOne 2025 - March 18-20, 2025", stating it took 30 years to make a Java developer conference this good. Below the banner, a section titled "Looking for other Java downloads?" includes links for "OpenJDK Early Access Builds" and "JRE for Consumers". The main content area is titled "Java 23, Java 21, and earlier versions available now". It states that JDK 23 is the latest release of the Java SE Platform, JDK 21 is the latest Long-Term Support (LTS) release, and earlier versions are available below. A link "Learn about Java SE Subscription" is present. Below this, there are links for "JDK 23", "JDK 21", "GraalVM for JDK 23", and "GraalVM for JDK 21". A section titled "Java SE Development Kit 23.0.2 downloads" follows, stating that JDK 23 binaries are free to use in production and free to redistribute under the Oracle No-Fee Terms and Conditions (NFTC). It also mentions that JDK 23 will receive updates under these terms until March 2025, when it will be superseded by JDK 24. At the bottom, there are links for "Linux", "macOS", and "Windows".

E' preferibile scaricare la versione JAVA 21
Trattasi di una realease con LONG TERM SUPPORT
<https://endoflife.date/java>

(oppure) Scaricare la Open-JDK

- E' possibile utilizzare la versione OpenSource di JAVA
 - <https://openjdk.org/>
 - <https://openjdk.org/projects/jdk/21>
- In generale può essere dipendente dal sistema operativo
 - Ubuntu 22.04:
<https://linuxhint.com/install-java-ubuntu/>
 - Apple:
<https://devqa.io/brew-install-java/>

Scaricare Eclipse

<https://www.eclipse.org/downloads/packages/>

The Eclipse Installer 2024-12 R now includes a JRE for macOS, Windows and Linux.

Try the Eclipse **Installer** 2024-12 R

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

[Find out more](#)

📄 **961,533 Installer Downloads**

📄 **732,723 Package Downloads and Updates**

Download

macOS [x86_64](#) | [AArch64](#)

Windows [x86_64](#) | [AArch64](#)

Linux [x86_64](#) | [AArch64](#) | [riscv64](#)

Eclipse IDE 2024-12 R Packages

Eclipse IDE for Enterprise Java and Web Developers

544 MB

325,221 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.



Click [here](#) to raise an issue with the Eclipse Web Tools Platform. Maintainers will move opened issues to the right place.



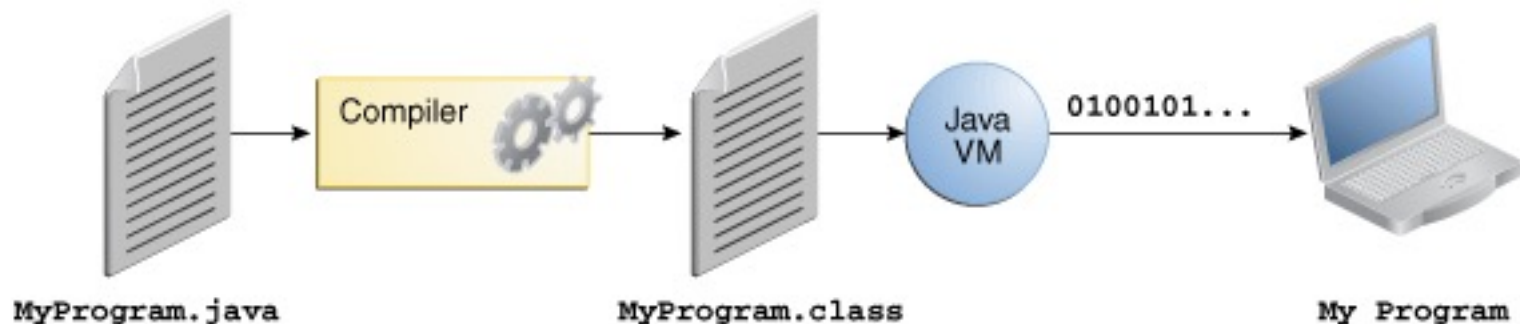
Windows | [x86_64](#) | [AArch64](#)

macOS [x86_64](#) | [AArch64](#)

Linux [x86_64](#) | [AArch64](#) | [riscv64](#)

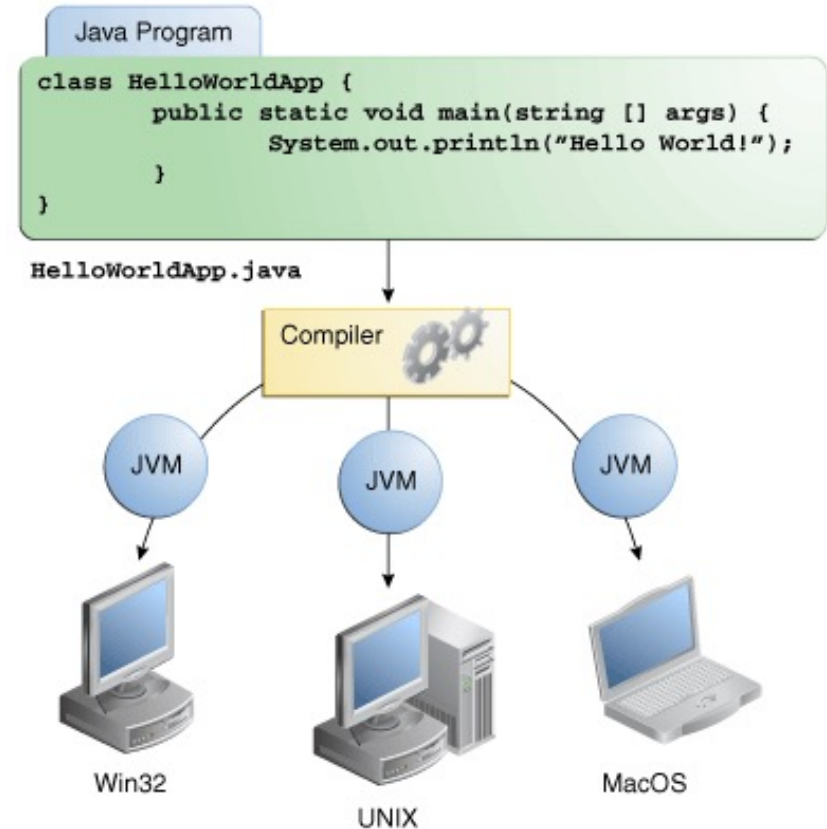
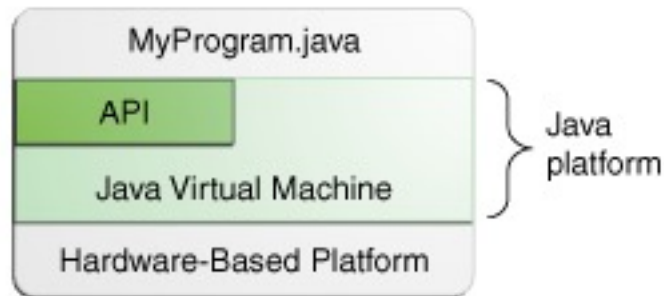
Il processo di sviluppo

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");

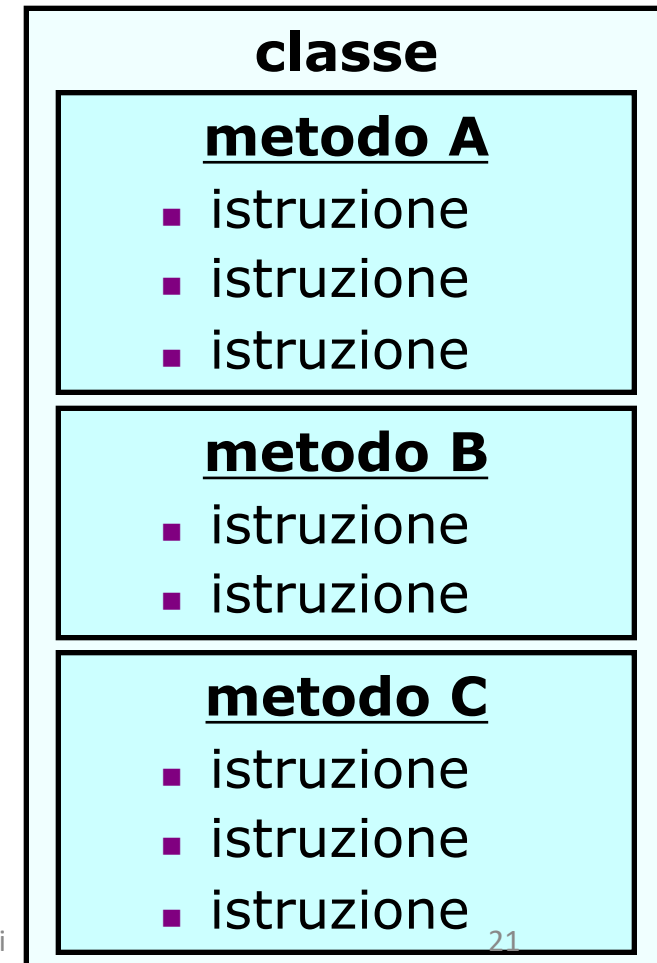
        System.out.println("Impostare la temperatura del forno");
        System.out.println("Impostare il timer");
        System.out.println("Mettere i biscotti nel forno");
        System.out.println("Aspettare che siano cotti");

        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:**

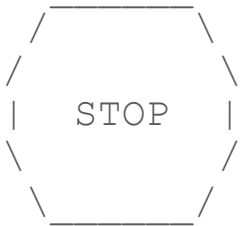
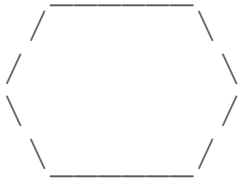
```
This is message1.
This is message2.
This is message1.
Done with message2.
Done with main.
```

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>