Programmazione 2: Introduzione

Marco Patrignani[1]

mailto: marco.patrignani@unitn.it
website: https://squera.github.io/

1 UNIVERSITÀ DI TRENTO

Who am I?

- Laurea a Bologna, PhD a KU Leuven, PostDoc a MPI-SWS & CISPA, professore a Stanford
- ricerca: semantica dei linguaggi di programmazione, metodi formali

Burocrazia

- Orario
- Esame
- Tools
- Links utili

Obiettivi / Motivazioni del Corso

- Abstraction, encapsulation, inheritance, and polymorphism
- Java come linguaggio per capire OOP

Consigli

- non basta venire a lezione e fare laboratorio
- studiate / fate esercizi a casa!
- installate da subito tutto il necessario (JDK, IntelliJ, etc)
- fate domande!

Java Sneak Peek

Java e C/C++: cosa resta uguale?

Istruzioni e commenti

```
int a=5%3; // commento
```

Strutture di controllo

```
for {int a=2; a<10; a++ {…}
do {…} while (i<10);
while (i<10) {…}
switch (s) { case 1: …; defeult:… }
if (i<10){…} elseif (i>200) {…} else {…}
```

Definizione e chiamate di funzioni

```
int f(int k) { return k+2;}
f(3); // ma niente puntatori e indirizzi!
```

Se volete…

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html

Java e C/C++: cosa è diverso?

Varie cose. Ad esempio:

- arrays
- implementazione dei tipi (qual'è il massimo intero in C/C++ ?)
- il main
- assenza di variabili globali
- struct, union
- …

Java e C/C++: cosa è diverso?

- Java **toglie** al C alcune caratteristiche complesse e potenzialmente "pericolose" (es. puntatori)
- Java **aggiunge** al C le caratteristiche di un linguaggio object-oriented
  - es. classi, ereditarietà, polimorfismo
- Java **introduce** una gerarchia di classi predefinite che in pratica diventano parte del linguaggio:
  - gestione dell'I/O
  - gestione di stringhe
  - gestione delle eccezioni
  - networking
  - concorrenza (`Thread`)
  - strutture dati (es. `Vector`, `Dictionary`, `Hashtable`)

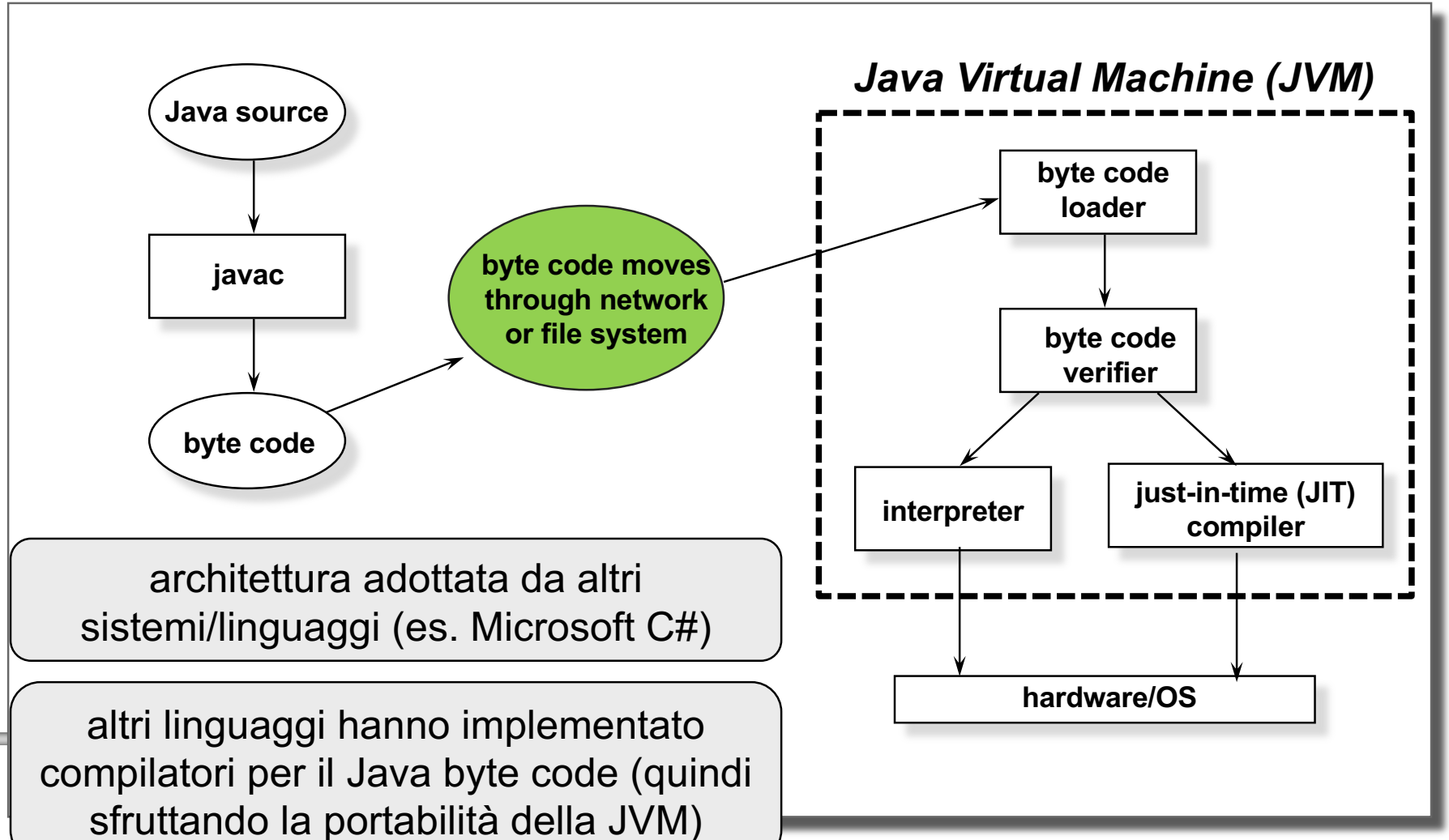
Perché è diverso?

- Molti errori comuni di programmazione sono legati alla gestione della memoria tramite ***puntatori***:
  - puntatori che puntano a locazioni illecite (non allocate)
  - puntatori che puntano a locazioni lecite ma sbagliate
  - memoria allocata e non più rilasciata (*memory leaks*)
- Soluzioni in Java:
  - ***Abolizione dei puntatori***: di conseguenza, **non** fornisce gli operatori di de-referenziazione e addressing **`*`**, **`->`**, **`&`**
  - ***Garbage collection***: gli oggetti vengono allocati e deallocati automaticamente in memoria: **`sizeof`**, **`malloc`**, **`free`** non sono più necessari

La struttura di un programma Java

- Un programma Java è organizzato come un insieme di **`classi`**
- Classi diverse possono essere raggruppate all'interno della stessa "*compilation unit*"
  - Es. un file **`.java`**
- Il programma principale è rappresentato da una funzione speciale (**`main`**) della classe il cui nome coincide con il nome del programma

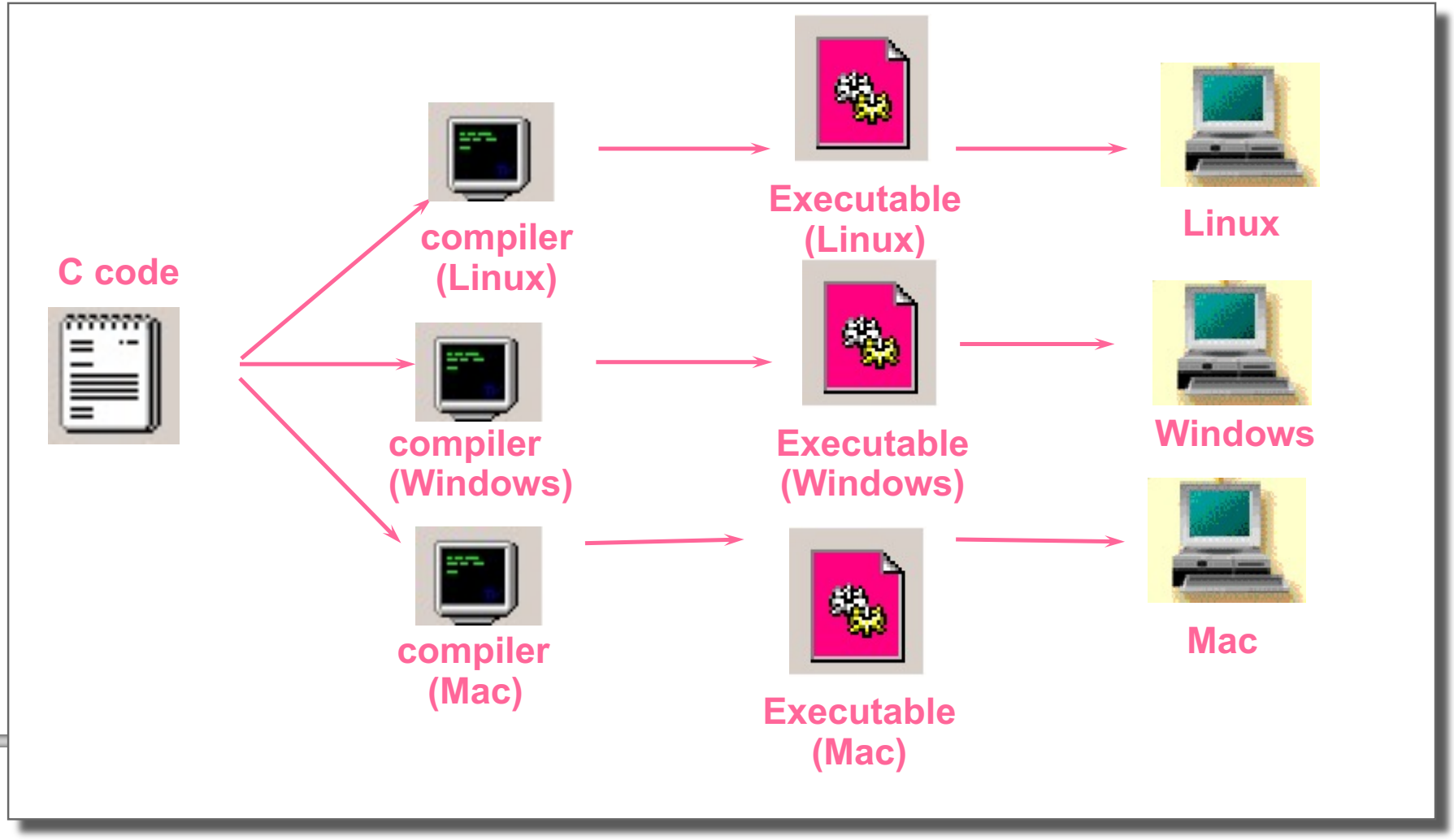
L'architettura di Java



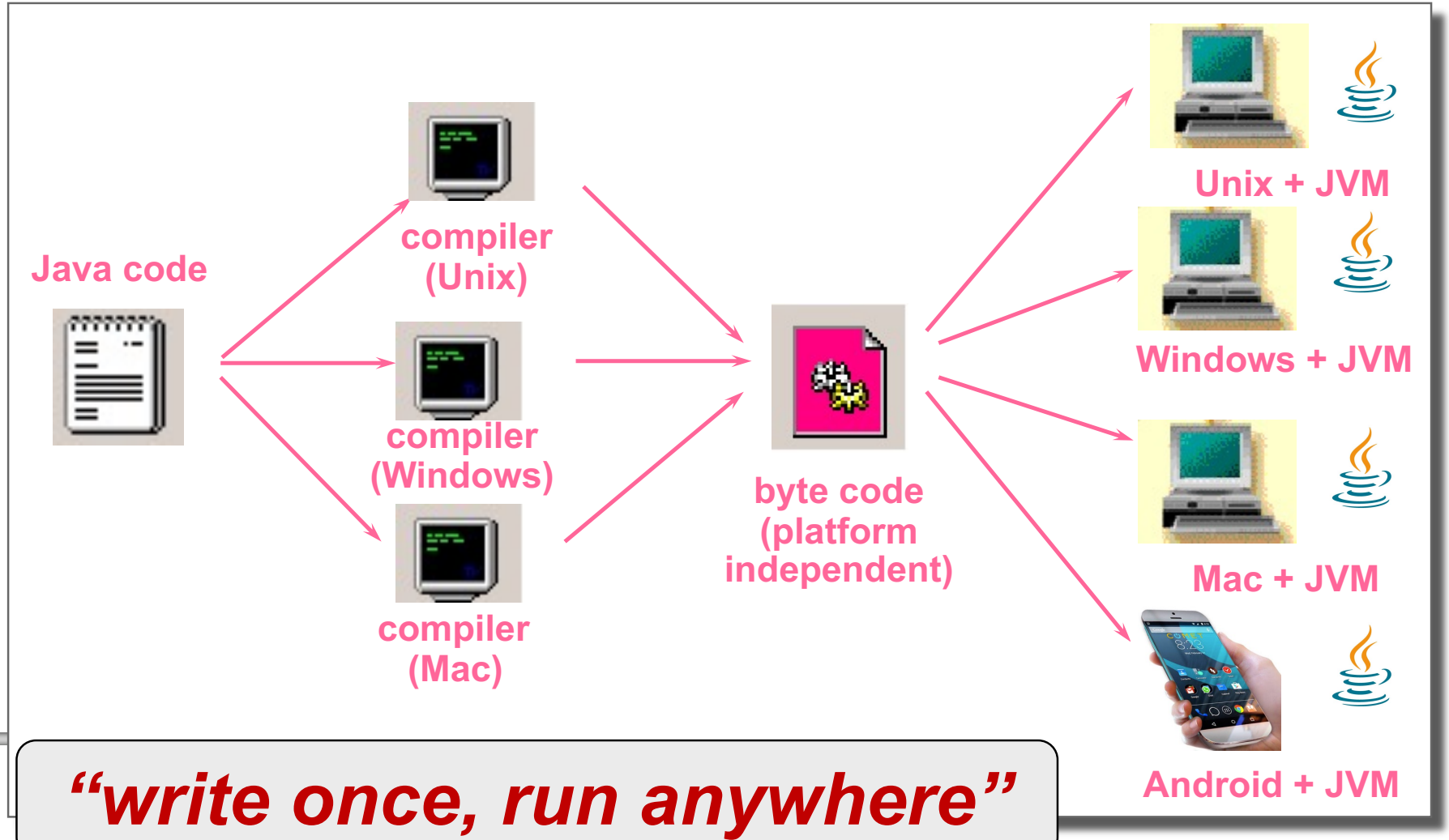
architettura adottata da altri sistemi/linguaggi (es. Microsoft C#)

altri linguaggi hanno implementato compilatori per il Java byte code (quindi sfruttando la portabilità della JVM)

Piattaforme e "portabilità"



Piattaforme e "portabilità"

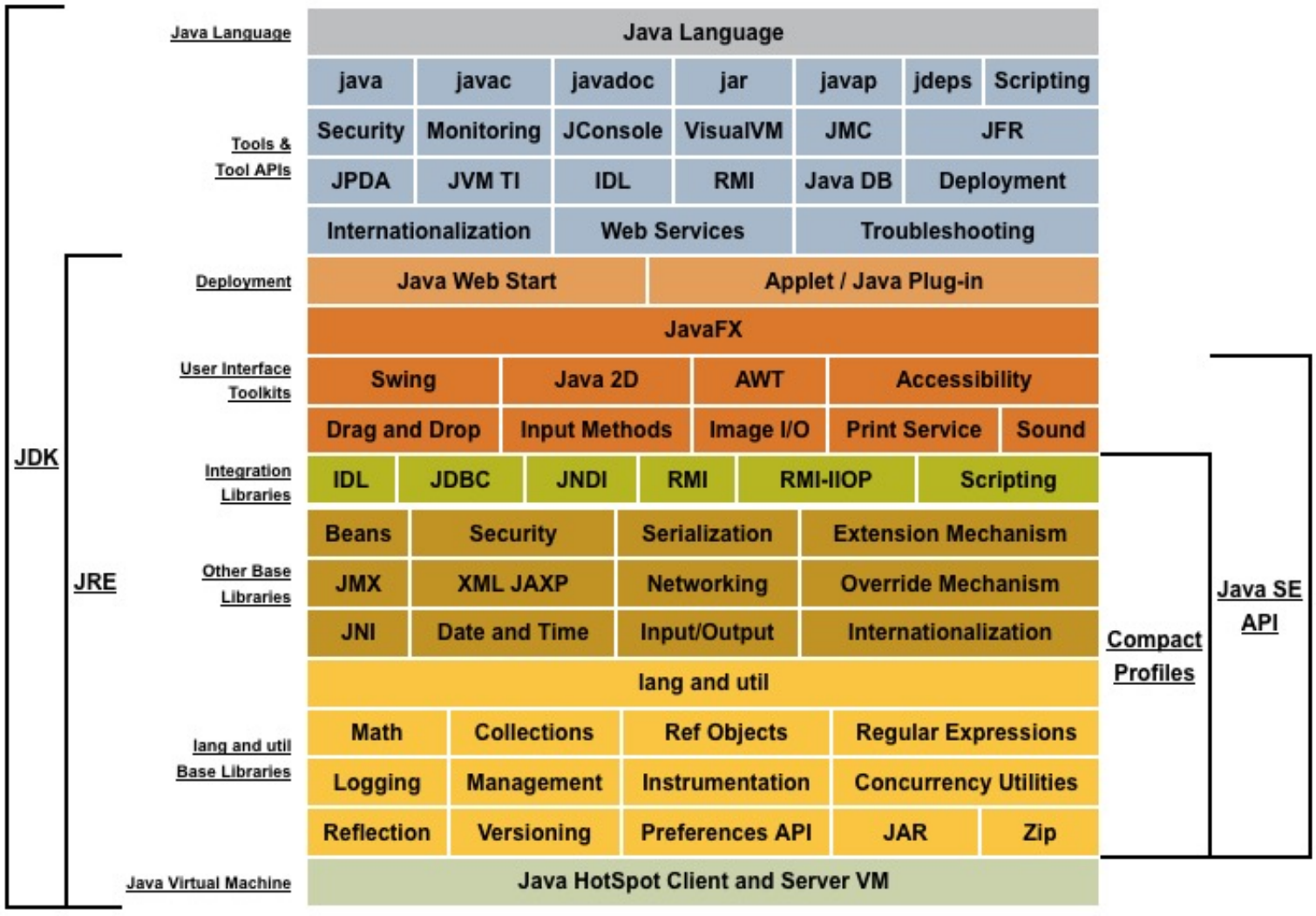


"write once, run anywhere"

Java: oggi

- Si stima che Java sia installato su oltre 15 miliardi di dispositivi…
- Android ha avuto un ruolo chiave nella diffusione recente di questo linguaggio
  - Anche se sfrutta una sua JVM incompatibile con quella standard, e si basa su librerie specifiche…
- Java è usato ampiamente in ambito Web e cloud
  - Web services, servlets, JSP, Struts, EJB, XML, …
- Numerosi IDE (*Integrated Development Environments*) sono scritti in Java
  - Eclipse, IntelliJ, NetBeans, …

The Java Platform



Tutorial ed esempi

https://docs.oracle.com/javase/tutorial

ORACLE Java Documentation

The Java™ Tutorials

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases.

The Java Tutorials are practical guides for programmers who want to use the Java programming language to create applications. They include hundreds of complete, working examples, and dozens of lessons. Groups of related lessons are organized into "trails".

Trails Covering the Basics

These trails are available in book form as *The Java Tutorial, Sixth Edition*. To buy this book, refer to the box to the right.

- Getting Started — An introduction to Java technology and lessons on installing Java development software and using it to create a simple program.
- Learning the Java Language — Lessons describing the essential concepts and features of the Java Programming Language.
- Essential Java Classes — Lessons on exceptions, basic input/output, concurrency, regular expressions, and the platform environment.
- Collections — Lessons on using and extending the Java Collections Framework.
- Date-Time APIs — How to use the `java.time` pages to write date and time code.
- Deployment — How to package applications and applets using JAR files, and deploy them using Java Web Start and Java Plug-in.
- Preparation for Java Programming Language Certification — List of available training and tutorial resources.

Not sure where to start?
See Learning Paths

Tutorial Contents

Really Big Index

Tutorial Resources

- Last Updated 7/19/2016
- The Java Tutorials' Blog has news and updates about the Java SE tutorials.
- Download the latest Java Tutorials bundle.

Un buon libro…

Thinking in Java, Bruce Eckel

- È disponibile online:
  `https://people.inf.elte.hu/delsaai/java/6Eckel%20-%20Thinking%20in%20Java%20(4th%202006)%20p1079.pdf`
- … non è l'edizione più recente, ma è gratis
- Se invece desiderate un'edizione cartacea e/o in italiano, la trovate (con lo stesso titolo/autore) edita da Apogeo – ma è sconsigliata perché la traduzione ha introdotto errori.