



Java

il linguaggio

- 1991 - **James Gosling** lavora al “Green Project” (**SUN**) per studiare la convergenza tra dispositivi di elettronica di consumo e computer
- 1992 il “Green Project” realizza un palmare che controlla dispositivi di elettronica di consumo con un’interfaccia utente animata e uno schermo sensibile al tocco
- il dispositivo si chiamava StarSeven ed era programmato mediante un linguaggio totalmente nuovo **Oak** (quercia) indipendente dal processore



- sintassi di ***Oak*** simile al C
 - include il supporto alla programmazione a oggetti
- gestione degli oggetti più semplice del C++
- la parola chiave di Oak era ***semplicità di utilizzo e chiarezza della sintassi***
- il prototipo non ebbe molto successo così la tecnologia del Green Project iniziò a essere adattata a Internet e divenne il primo embrione di tecnologia Java

primary goals in the creation of the Java language:

1. it should use the ***object-oriented programming*** methodology
2. it should allow the same program to be executed on ***multiple operating systems***
3. it should contain built-in support for using ***computer networks***
4. it should be designed to execute code from remote sources ***securely***
5. it should be ***easy to use*** by selecting what was considered the good parts of other object-oriented languages

- **Java** è una varietà di caffè
 - *bevanda molto utilizzata dai programmatori*
- nel **1996** viene rilasciata la versione 1.0a2, la prima destinata al pubblico
 - il software viene rilasciato completo di sorgenti
- la versione 1.2 di Java viene chiamata Java2

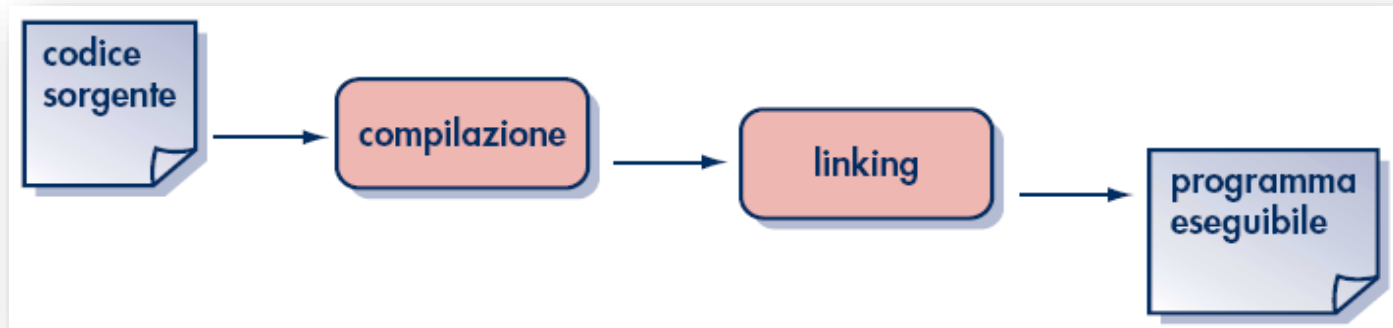
Version	Release date
JDK Beta	1995
JDK 1.0	January 1996
JDK 1.1	February 1997
J2SE 1.2	December 1998
J2SE 1.3	May 2000
J2SE 1.4	February 2002
J2SE 5.0	September 2004
Java SE 6	December 2006
Java SE 7	July 2011
Java SE 8 (LTS)	March 2014
Java SE 9	September 2017
Java SE 10 (18.3)	March 2018
Java SE 11 (18.9 LTS)	September 2018
Java SE 12 (19.3)	March 2019

linguaggi compilati e interpretati

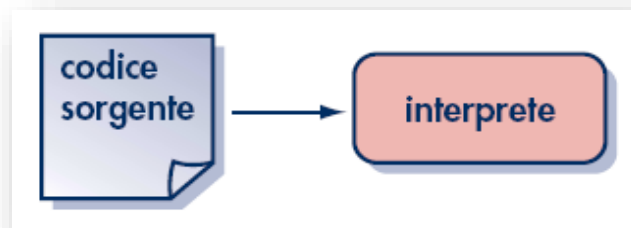
A. Ferrari

- nei programmi scritti con linguaggi di programmazione ad alto livello ogni istruzione si traduce in un insieme spesso corposo di istruzioni a livello macchina
- i linguaggi di programmazione si dividono in due tipologie:
 - linguaggi *compilati*
 - linguaggi *interpretati*
- la differenza è il modo in cui il linguaggio ad alto livello viene tradotto in istruzioni in codice macchina

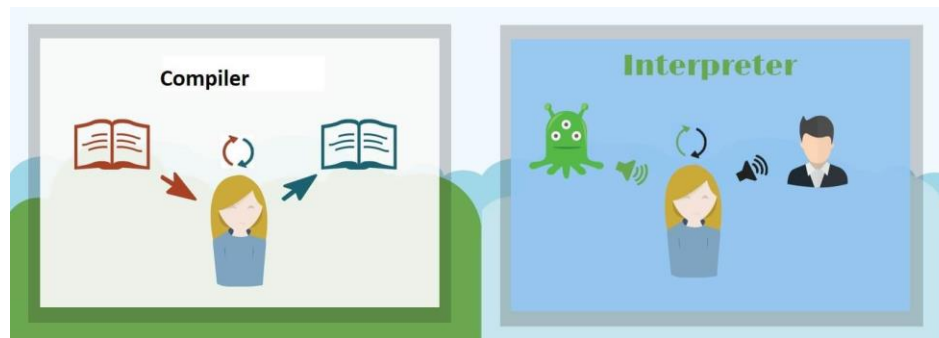
- il **compilatore** è un software che traduce il codice sorgente in codice macchina e lo memorizza in un file (**codice oggetto**)
- un programma eseguibile contiene istruzioni in codice macchina **specifiche** di un processore
- esempi di linguaggi **compilati** sono **C** e **C++**



- il programma viene ***eseguito*** direttamente da un software (***interprete***) che esegue le istruzioni in codice macchina necessarie per le funzionalità richieste
- l'interprete simula il funzionamento di un processore
- esempi di linguaggi interpretati: ***Visual Basic, JavaScript***



- i linguaggi interpretati sono generalmente ***più lenti*** dei linguaggi compilati
- i linguaggi interpretati offrono come vantaggio la ***rapidità di sviluppo***
- un programma interpretato è eseguibile immediatamente per essere provato dal programmatore



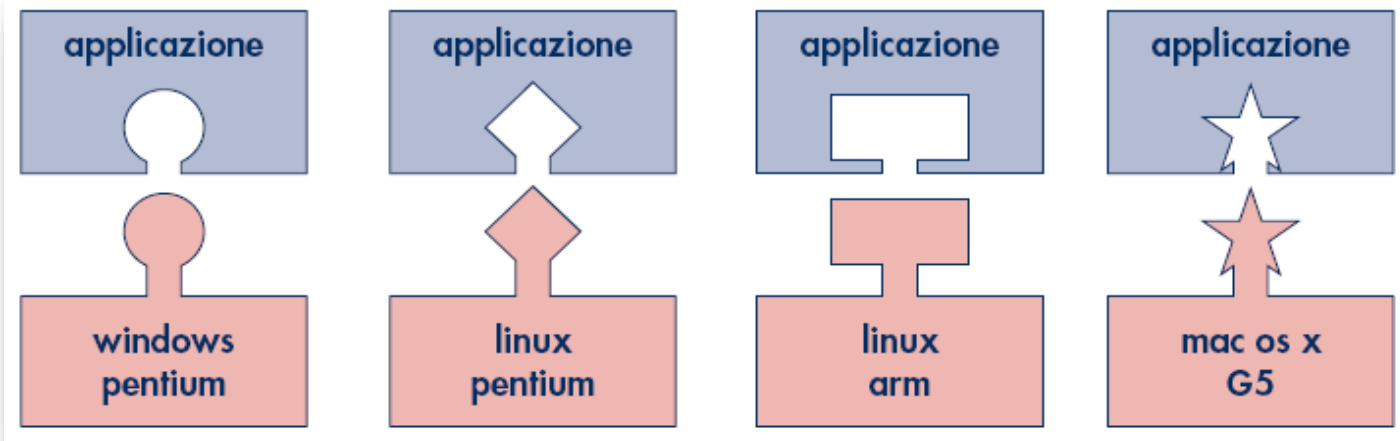
Java: compilato e interpretato

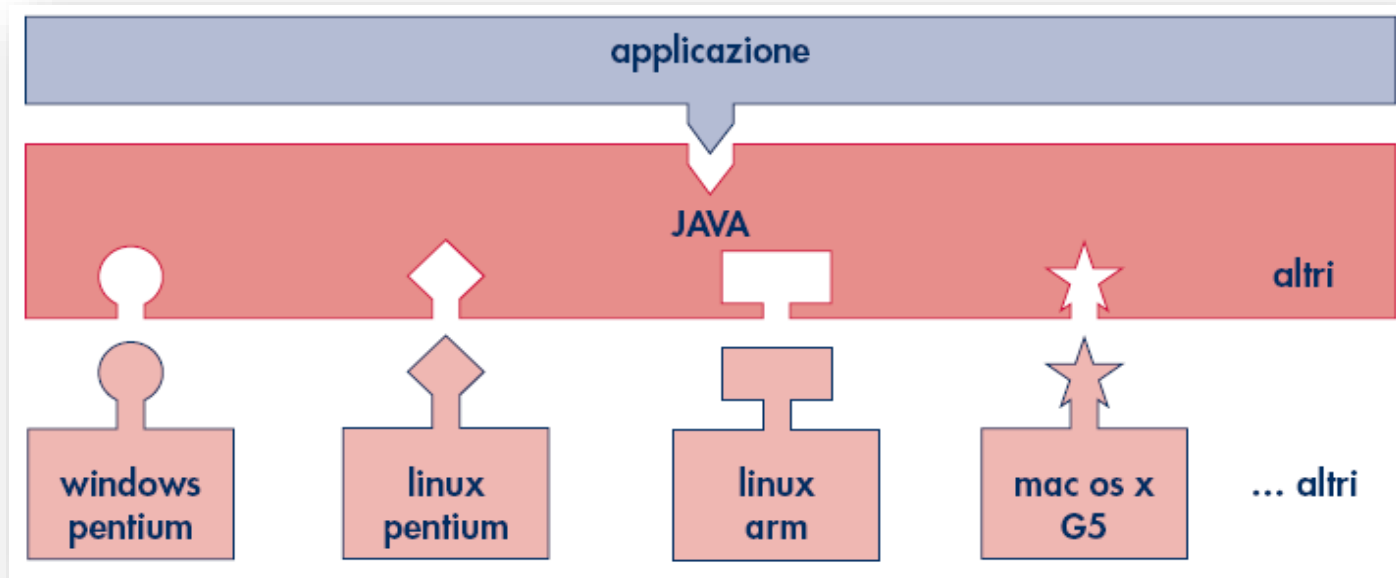
A. Ferrari

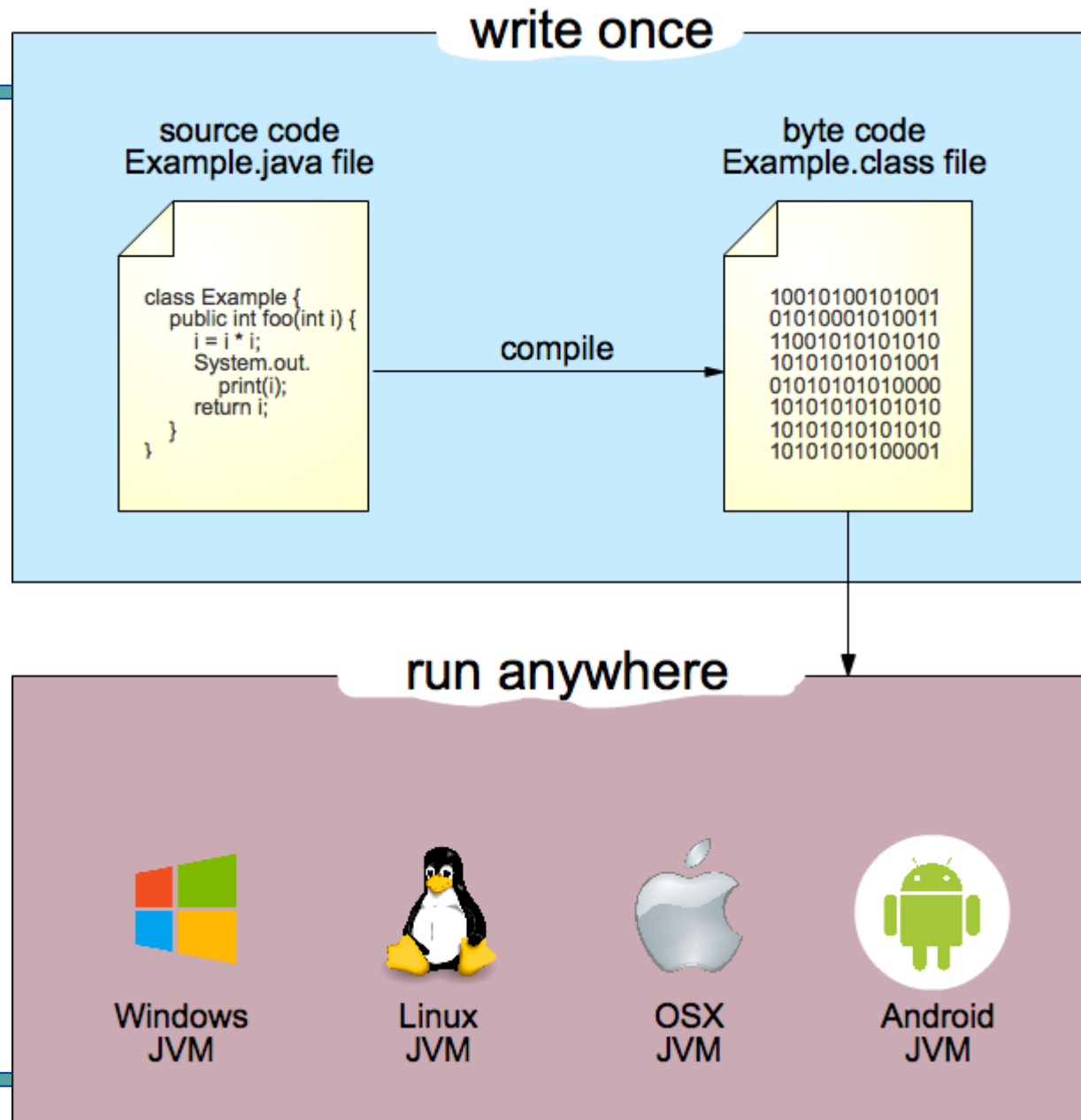
- Java è sia compilato che interpretato
- i file che contengono il codice *sorgente* di un programma Java (*.java*) sono compilati (*javac*)
- il file generato ha lo stesso nome del sorgente ed estensione *.class*
- il file compilato non contiene codice oggetto specifico di un determinato processore ma un *bytecode*
- il bytecode non è specifico per alcun processore
 - Java è un linguaggio multiplatforma (può essere eseguito su diverse combinazioni di processori e sistemi operativi)
- per eseguire il bytecode è necessario un *interprete*, che traduca il codice oggetto (bytecode) in istruzioni del processore in uso (*Java Virtual Machine*)

porting dei linguaggi compilati

A. Ferrari







- con Java è possibile sviluppare programmi, orientati a diversi ambiti:
 - dispositivi integrati (smartphone)
 - personal computer
 - applicazioni client/server

