

JAVA

Introduzione al mondo Java



Principali caratteristiche di Java

- Object Oriented
- Semplice (evoluzione del C)
- Indipendente dalla piattaforma (portabile)
 - Indipendente dall'architettura
- Robusto
- Concorrente (multithreading)
- Sicuro

Object Oriented

- Java è un linguaggio di programmazione che sposa completamente il paradigma Object Oriented.
- Object Oriented è un modo di ragionare e non un linguaggio.

Object Oriented

- Esistono anche linguaggi funzionali con alcuni concetti di OO e sono detti *Object Based*, un esempio è VB 6.0.
- Ad oggi i principali linguaggi OO sono:
 - C++
 - Java
 - la piattaforma .net della Microsoft

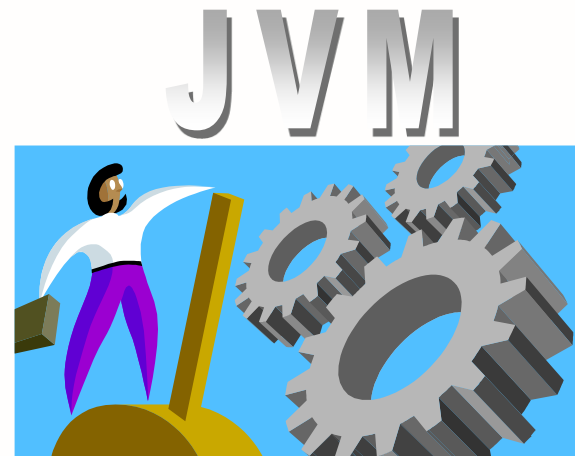
Semplice

- Sintassi simile a C e C++
- Elimina i costrutti più "pericolosi" di C e C++
 - aritmetica dei puntatori
 - (de)allocazione esplicita della memoria
- Funzionalità di garbage collection (GC) automatica

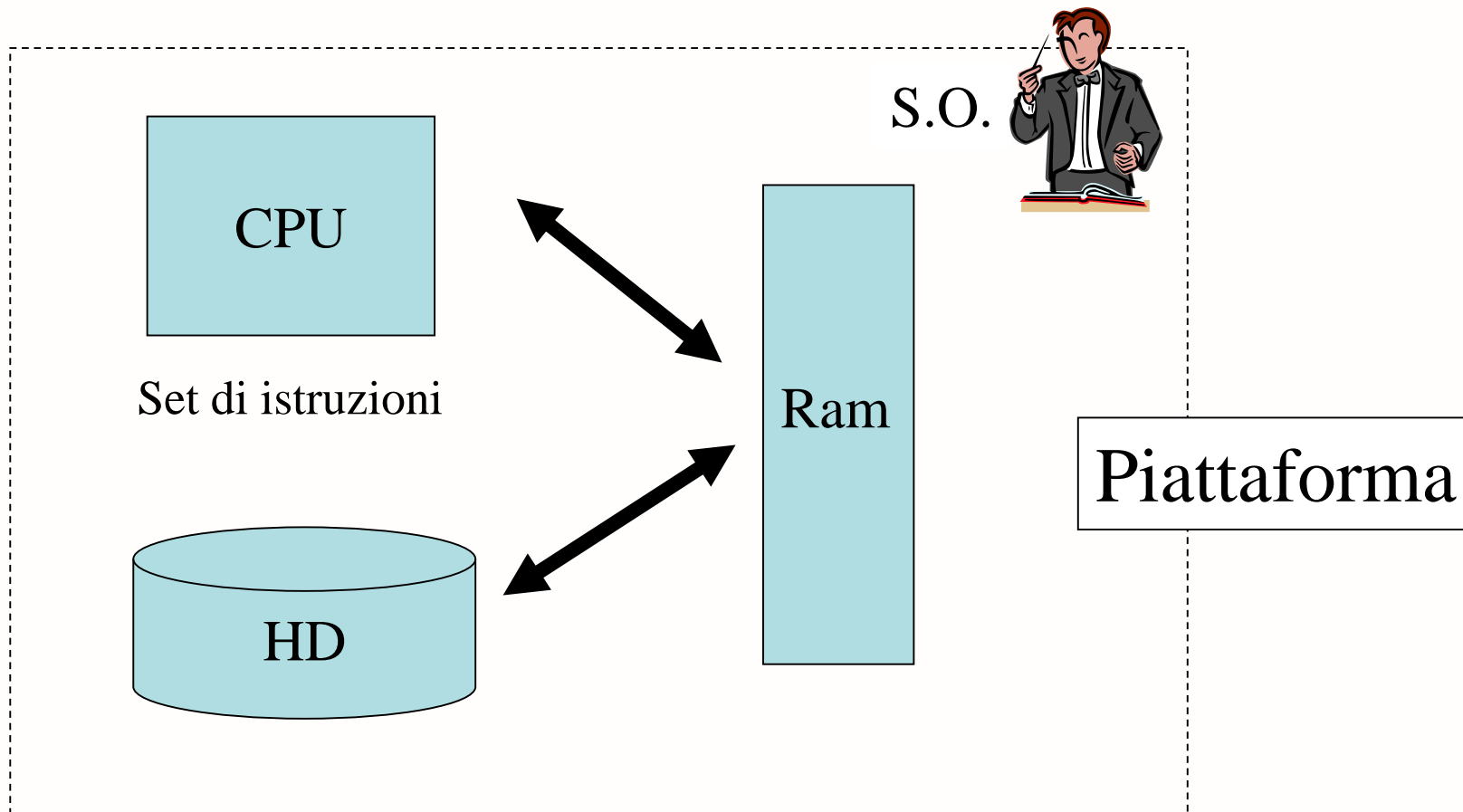
Indipendente dalla piattaforma

Un programma Java si **scrive una volta** sola
e deve **girare su tutte** le piattaforme che hanno
una macchina virtuale.

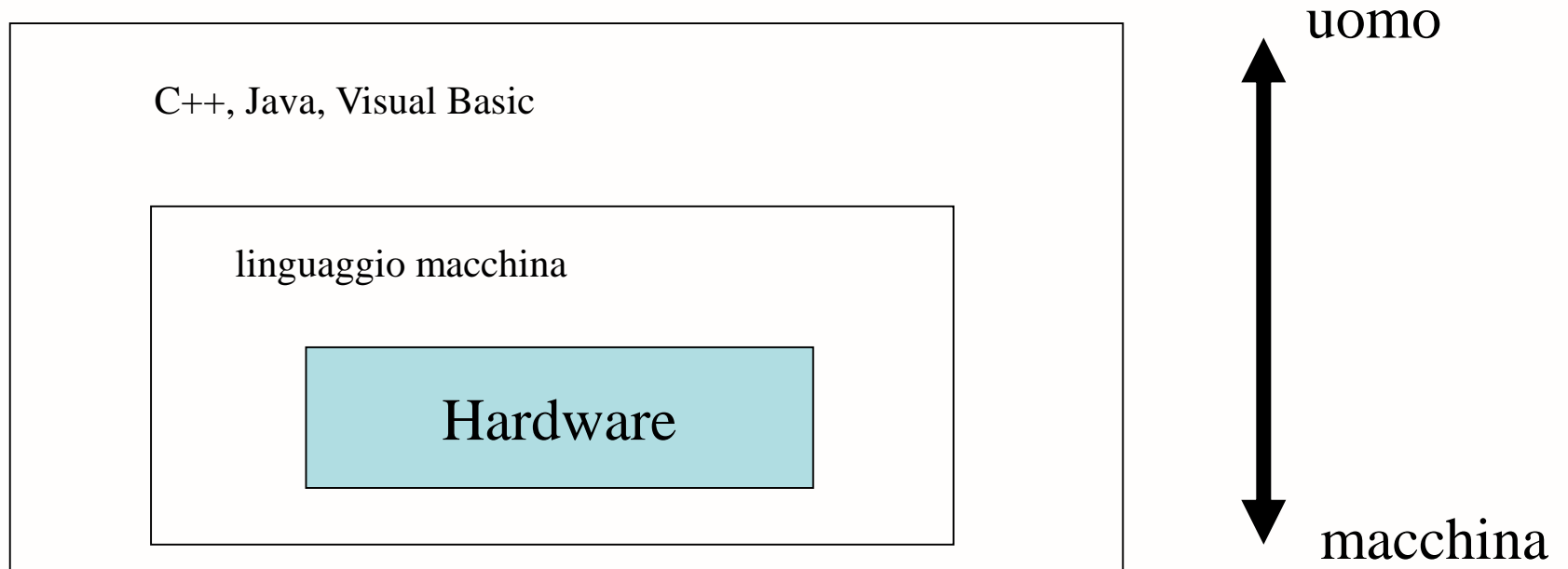
```
Public class Date  
{  
    int giorno;  
    int mese;  
    int anno  
    ...  
}
```



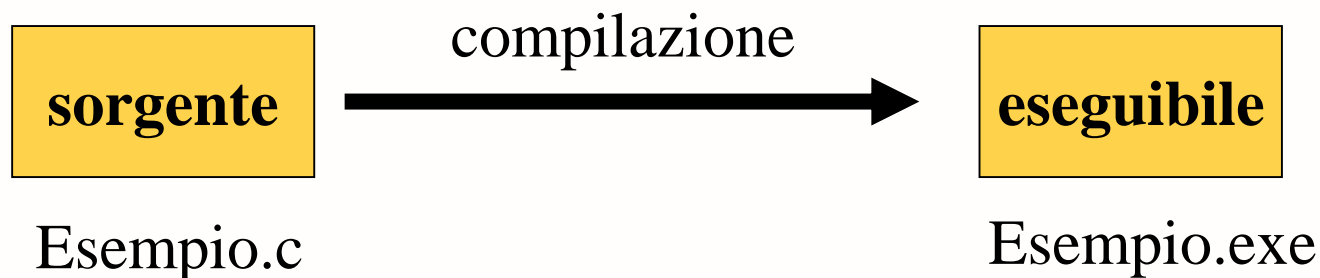
Piattaforma



Linguaggi di programmazione



Compilatore



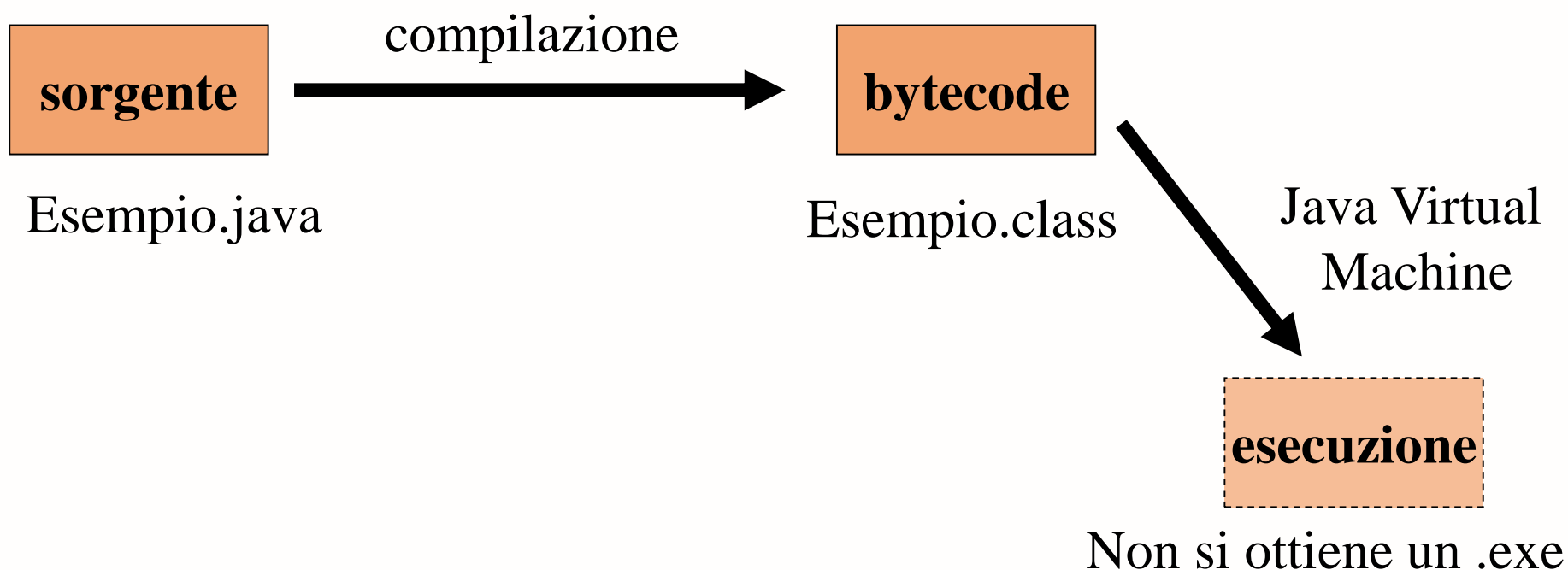
Il compilatore è un software in grado di:

- Rilevare eventuali errori (sintassi ed altro...)
- Tradurre il codice sorgente nel codice macchina

Il file .exe è eseguibile solo sulla piattaforma per la quale è stato compilato.

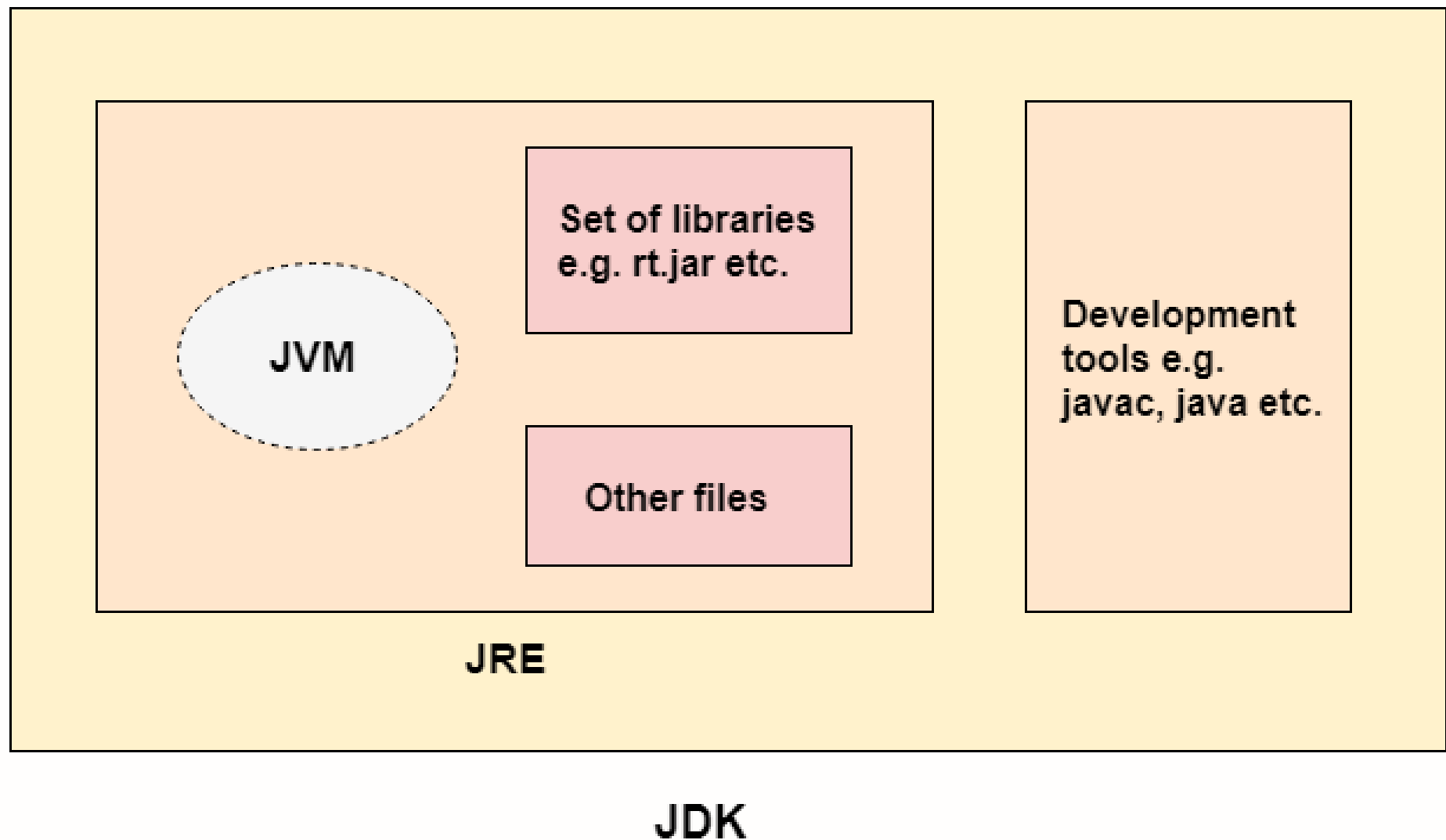
Esiste un compilatore per ogni piattaforma

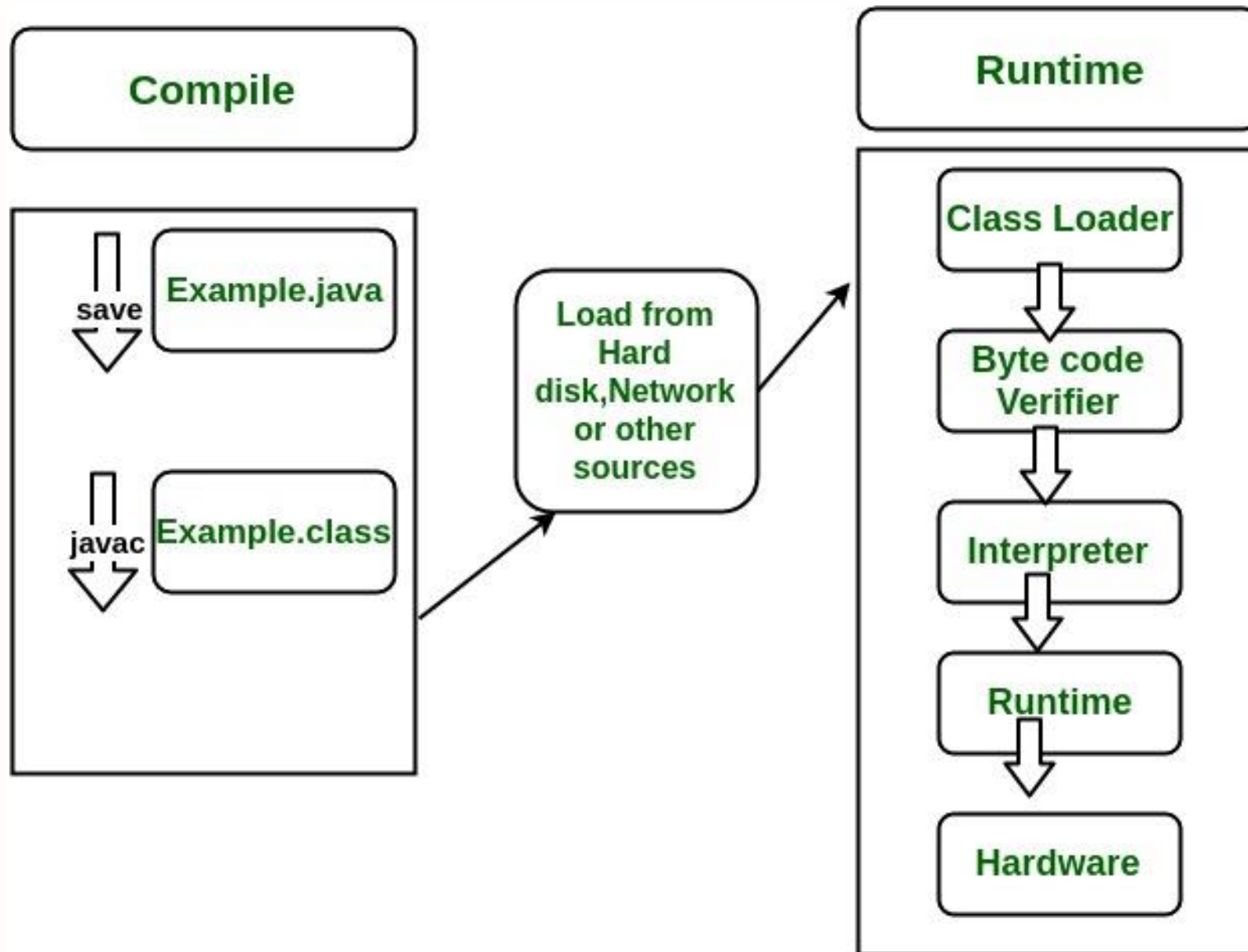
Linguaggio interpretato



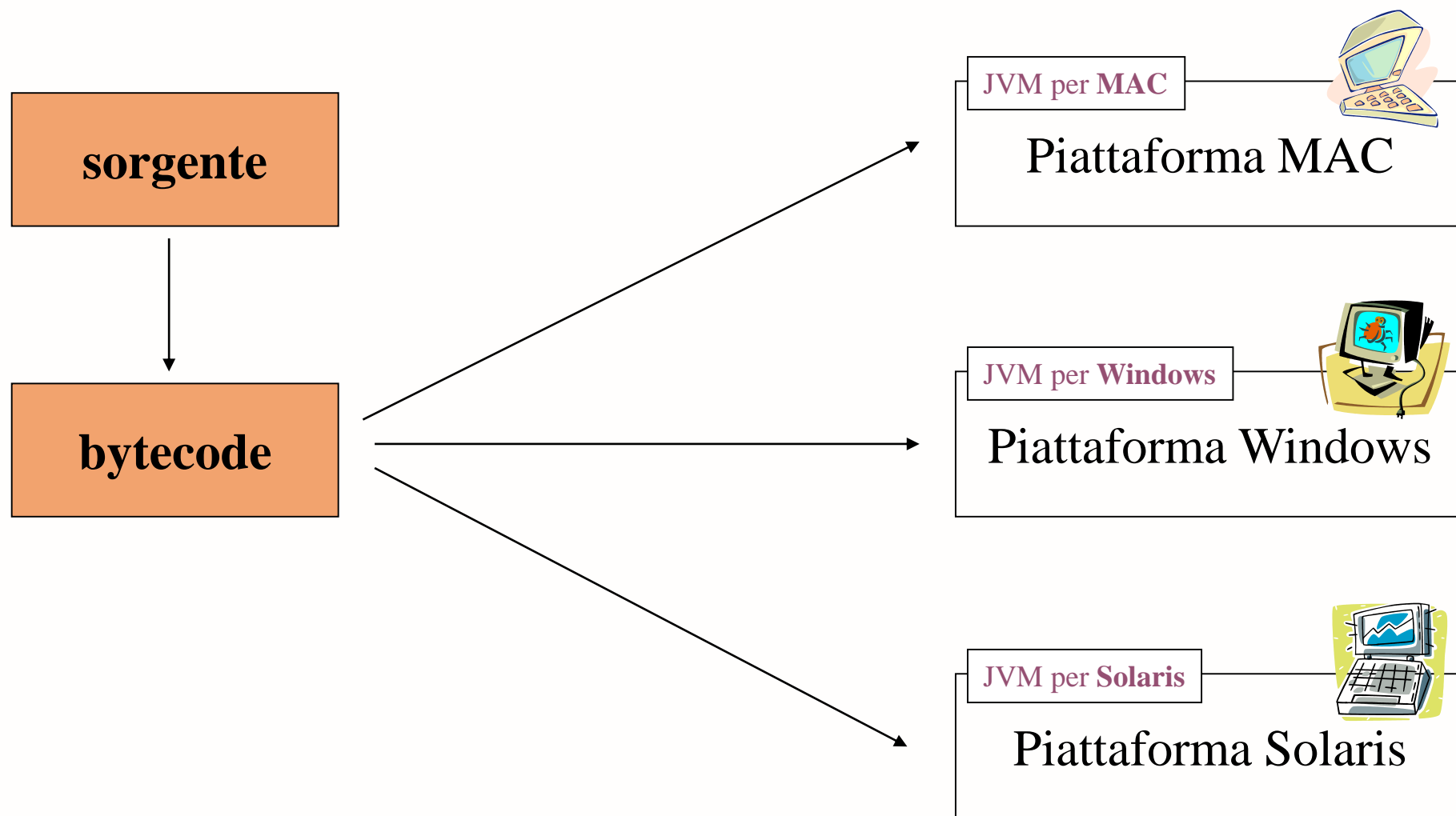
Il **bytecode** è un linguaggio macchina virtuale.

Virtuale: significa che nessun processore reale lo supporta.

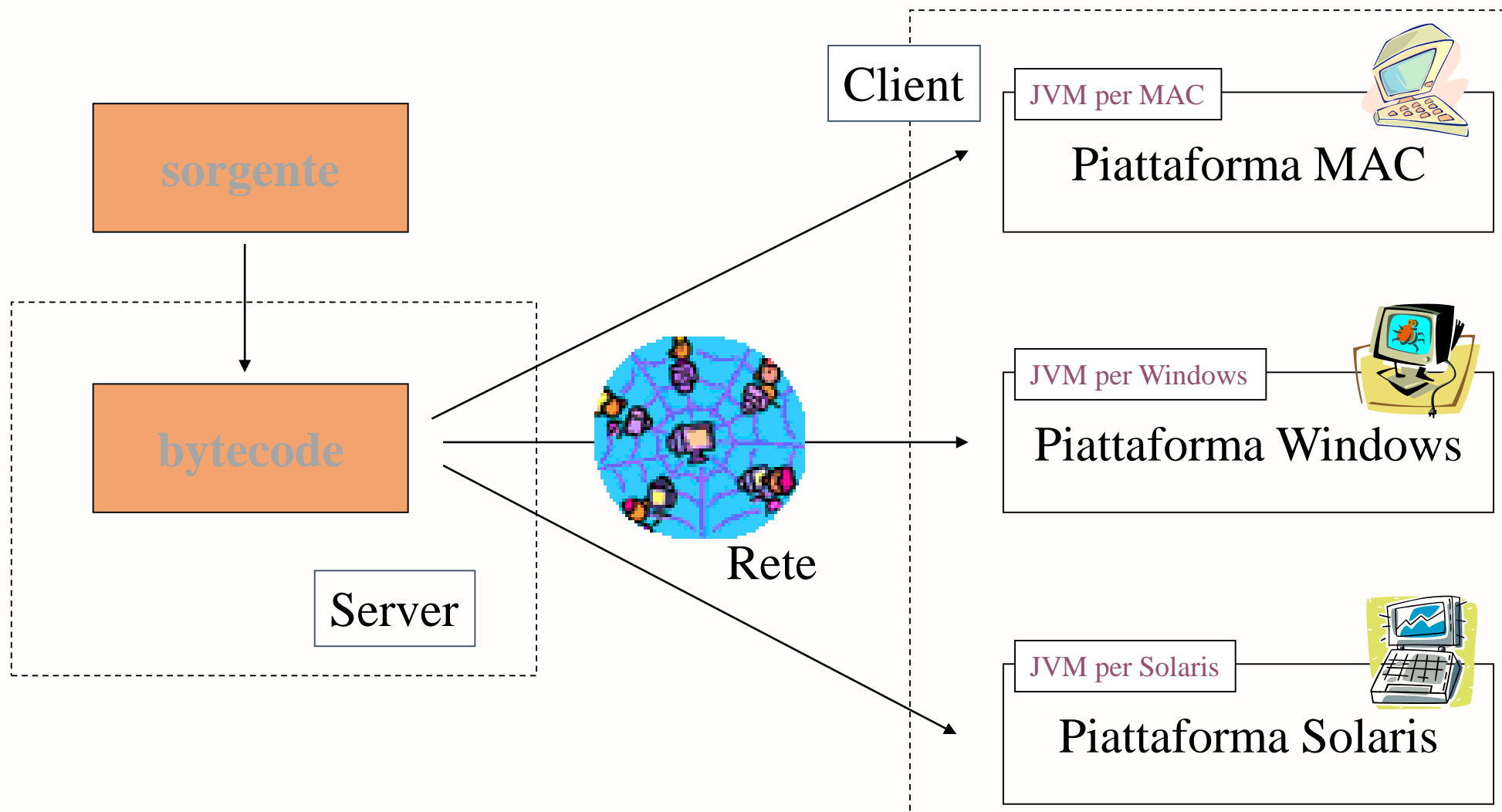




Indipendenza dalla piattaforma



Java e la rete



Indipendenza dall'architettura

- ♦ I programmi che si eseguono su tutte le piattaforme, devono girare **ALLO STESSO MODO.**



Robusto (affidabile)

- Rilevamento errori in fase di compilazione e di esecuzione
- Mascheramento dei puntatori all'utente
- Gestione delle eccezioni da parte dell'utente

Concorrente

- Come il sistema operativo è in grado di gestire contemporaneamente più programmi (***Multitasking***), in Java è possibile mandare in esecuzione parti diverse di uno stesso programma contemporaneamente (***Multithreading***)
- Nella libreria della Sun/Oracle sono disponibili strumenti (API) che consentono di sviluppare facilmente programmi multithreading

Sicuro

Scelte architettureali:

- ✓ Assenza dei puntatori (non permette di manipolare locazioni di memoria in modo arbitrario)
- ✓ No overrun dello stack a runtime (Java Exception)

API disponibili:

- ✓ JCE (Java Cryptography Extension)
- ✓ JSSE (Java Secure Socket Extension)
- ✓ JAAS (Java Authentication and Authorization Service)

Utilizzare Java

Java 2 Standard Edition development kit



Che cosa serve?

Dal sito della Oracle/Sun è possibile scaricare il Kit minimo per lavorare in Java:

- ✓ Il compilatore
- ✓ L'interprete java (JVM per la piattaforma prescelta)
- ✓ La libreria SE
- ✓ altri tools utili



www.oracle.com

Kit disponibili in Java

J2SE

J2EE

J2ME

Java 2 Standard Edition

Java 2 Enterprise Edition

Java 2 Micro Edition

Variabili d'ambiente

- Sotto Pannello di controllo → sistema → avanzate → variabili d'ambiente

- **PATH** = `jdk_path\bin`

I programmi java si possono compilare e lanciare da tutte le cartelle.

- **CLASSPATH**=`. ;jdk_path\lib`

Si specifica dove cercare le classi coinvolte nei programmi che si vanno a scrivere (comprese quelle della libreria)

Dove `jdk_path` è il path della JDK

Esempio:

PATH = `C:\Programmi\Java\jdk1.5.0_01\bin`

CLASSPATH = `. ;C:\Programmi\Java\jdk1.5.0_01\lib`

Variabili d'ambiente

- Sotto Pannello di controllo → sistema → avanzate → variabili d'ambiente

- `PATH = jdk_path\bin`

I programmi java si possono compilare e lanciare da tutte le cartelle.

- `CLASSPATH= . ;jdk_path\lib`

Si specifica dove cercare le classi coinvolte nei programmi che si vanno a scrivere (comprese quelle della libreria)

Dove `jdk_path` è il path della JDK

Esempio:

```
PATH = C:\Programmi\Java\jdk1.5.0_01\bin
```

```
CLASSPATH = .;C:\Programmi\Java\jdk1.5.0_01\lib
```

Primo programma



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

Salvare il file come **HelloWorld.java**

Lanciare

javac HelloWorld.java	(compilazione)
java HelloWorld	(esecuzione)