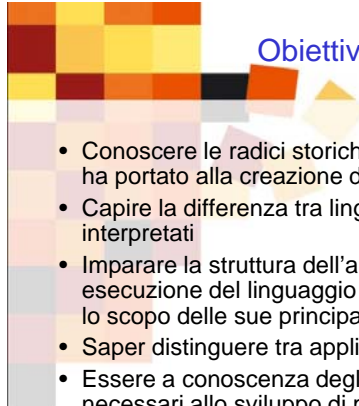




Unità A1

Introduzione a Java


© 2007 SEI-Società Editrice Internazionale, Apogeo



Obiettivi

- Conoscere le radici storiche e il contesto che ha portato alla creazione del linguaggio Java
- Capire la differenza tra linguaggi compilati e interpretati
- Imparare la struttura dell'ambiente di esecuzione del linguaggio Java e conoscere lo scopo delle sue principali componenti
- Saper distinguere tra applicazioni e applet
- Essere a conoscenza degli strumenti necessari allo sviluppo di programmi Java

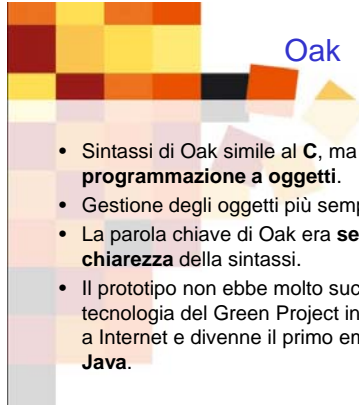
© 2007 SEI-Società Editrice Internazionale, Apogeo



La nascita di Java

- 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**.

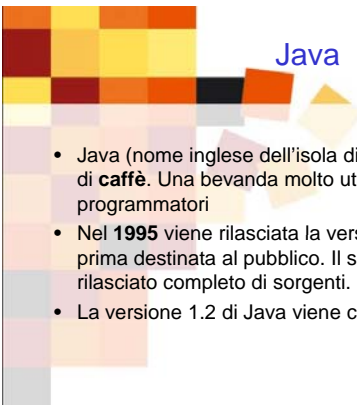
© 2007 SEI-Società Editrice Internazionale, Apogeo



Oak


- Sintassi di Oak simile al **C**, ma 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**.

© 2007 SEI-Società Editrice Internazionale, Apogeo

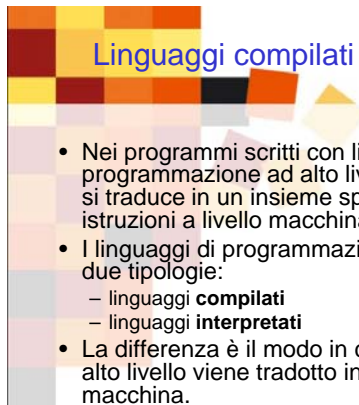


Java

- Java (nome inglese dell'isola di Giava), è una varietà di **caffè**. Una bevanda molto utilizzata dai programmatori
- Nel **1995** viene rilasciata la versione 1.0a2 di Java, la prima destinata al pubblico. Il software viene rilasciato completo di sorgenti.
- La versione 1.2 di Java viene chiamata Java2.



© 2007 SEI-Società Editrice Internazionale, Apogeo



Linguaggi compilati e interpretati

- 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.

© 2007 SEI-Società Editrice Internazionale, Apogeo

Linguaggi compilati

```

    graph LR
      A[codice sorgente] --> B[compilazione]
      B --> C[linking]
      C --> D[programma eseguibile]
  
```

- 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++.

© 2007 SEI-Società Editrice Internazionale, Apogeo

Linguaggi interpretati

```

    graph LR
      A[codice sorgente] --> B[interprete]
  
```

- 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, Perl

© 2007 SEI-Società Editrice Internazionale, Apogeo

Confronto

- I linguaggi **interpretati** sono generalmente **più lenti** dei linguaggi **compilati**
- Uno stesso programma realizzato mediante un linguaggio compilato produce in genere migliori performance rispetto a quando è eseguito con un interprete.
- I linguaggi **interpretati** offrono come vantaggio la **rapidità di sviluppo**.
- Un programma interpretato è eseguibile **immediatamente** per essere provato dal programmatore.

© 2007 SEI-Società Editrice Internazionale, Apogeo

Java: compilato e interpretato

- 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 poiché 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**)

© 2007 SEI-Società Editrice Internazionale, Apogeo

Porting dei linguaggi compilati

© 2007 SEI-Società Editrice Internazionale, Apogeo

Porting di Java

© 2007 SEI-Società Editrice Internazionale, Apogeo



Applicazioni e applet

- Con Java è possibile sviluppare programmi, orientati a diversi ambiti: ai dispositivi integrati (cellulari e PDA), ai personal computer, alle applicazioni client/server
- Le due principali tipologie di programmi che è possibile realizzare con Java sono le applicazioni e le applet.
- Le **applicazioni** si eseguono sul personal computer in modo indipendente
- Le **applet** vivono all'interno delle pagine web girano all'interno di un contesto di sicurezza limitato (es. non possono accedere a file locali e non possono stabilire connessioni di rete con computer diversi da quello da cui sono state scaricate)

© 2007 SEI-Società Editrice Internazionale, Apogeo



Sintesi

- La tecnologia Java è nata da un progetto di ricerca di SUN sulla convergenza tra dispositivi integrati e informatica
- Java è orientato agli oggetti, ha una sintassi molto simile a quella del C
- È indipendente dalla piattaforma, gira in modo identico su tutti i sistemi operativi/computer
- I linguaggi compilati prevedono una fase che trasforma il codice sorgente in codice oggetto e da questo in codice macchina
- I linguaggi interpretati sono eseguiti al volo da un programma interprete
- Java è un linguaggio compilato che produce bytecode che viene eseguito dall'ambiente di runtime
- Il compilatore Java (javac) prende in input un file .java e produce un file .class che viene eseguito dalla Virtual Machine
- Con Java è possibile realizzare due tipologie di programmi: applicazioni e applet.

© 2007 SEI-Società Editrice Internazionale, Apogeo

