

Capitolo 4

**Traduzione ed esecuzione di
programmi**

Ottobre 2006

Nota bene

- ❑ Alcune parti del presente materiale didattico sono derivate da:
 - trasparenze per il corso di Laboratorio di Informatica A.A. 2005-2006 prodotte da Alfonso Miola
- ❑ L'utilizzo di questo materiale è stato consentito dall'autore

Contenuti

- ❑ Traduzione ed esecuzione di programmi
- ❑ Compilatori ed interpreti
- ❑ L'approccio misto di Java
- ❑ L'uso di ambienti di sviluppo integrati (IDE)

Calcolatori e programmazione

- ❑ Un calcolatore è una **macchina per eseguire programmi**
 - qualcuno ha già fatto per noi alcuni programmi che possiamo eseguire così come facciamo con ogni applicazione
- ❑ Un calcolatore è anche una **macchina programmabile**, ossia una macchina le cui potenzialità possono essere aumentate da noi con la scrittura di nuovi programmi
 - per far questo dobbiamo far uso di opportuni linguaggi di programmazione che ci permettono di esprimere nel modo più naturale le nostre esigenze di calcolo

Linguaggi di programmazione

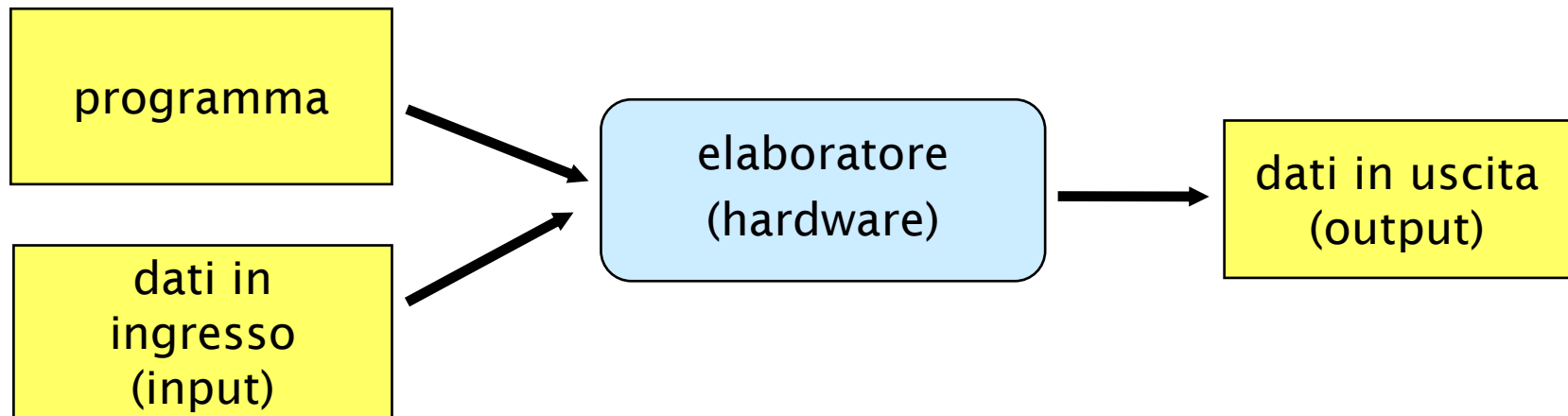
- ❑ Ogni calcolatore ha un proprio linguaggio macchina
 - ogni istruzione coincide con un'istruzione di macchina e pertanto è un **linguaggio binario** o anche di **linguaggi a basso livello**
- ❑ Esistono linguaggi di programmazione che consentono di scrivere i programmi in termini di istruzioni “vicine” alla logica delle tecniche di risoluzione di problemi
 - le istruzioni sono espresse mediante simboli e pertanto tali linguaggi sono anche detti **linguaggi simbolici** o **linguaggi ad alto livello**
 - sono linguaggi simbolici anche i **linguaggi assemblativi**, in cui ad ogni istruzione del linguaggio corrispondono una o più (poche) istruzioni di macchina

Esecuzione dei programmi

- ❑ Il Calcolatore, in particolare il suo hardware, esegue solo programmi in **linguaggio macchina**
- ❑ Se un programma è in linguaggio macchina può essere quindi direttamente eseguibile
 - tutte le applicazioni sono immediatamente eseguibili proprio per questo motivo
- ❑ Se un programma è in linguaggio simbolico non può essere eseguito immediatamente
 - c'è bisogno di un **processo di traduzione** che viene realizzato tramite opportune applicazioni già disponibili, e che sono quindi evidentemente in linguaggio macchina

Programmi in linguaggio macchina

- Un calcolatore è una macchina che ricevendo un programma in linguaggio macchina e i relativi dati in ingresso lo esegue producendo dati in uscita (i risultati dell'elaborazione)
 - durante questa esecuzione la macchina (hardware) diventa una **macchina virtuale**



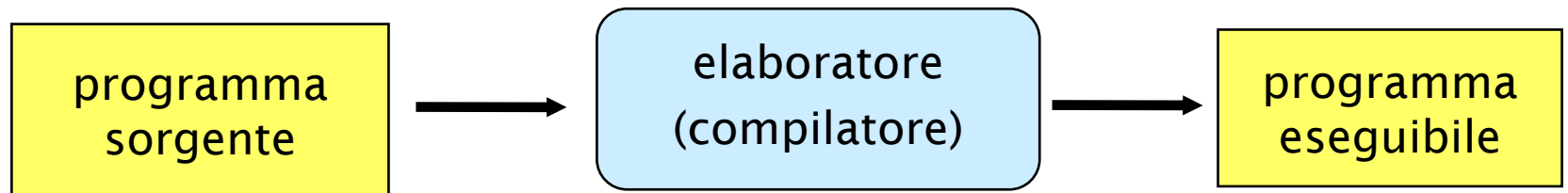
Compilazione e interpretazione

- Esistono due approcci alla traduzione ed esecuzione dei programmi
 - **compilazione**
 - un programma scritto in un linguaggio di programmazione di alto livello viene trasformato in un programma in linguaggio macchina e poi eseguito
 - **interpretazione**
 - ciascuna istruzione del programma scritto in un linguaggio di programmazione di alto livello viene trasformata in istruzioni del linguaggio macchina ed eseguita
 - i traduttori del primo tipo sono chiamati effettivamente **compilatori**, quelli del secondo tipo sono chiamati **interpreti**

- Una analogia con la traduzione tra linguaggi diversi
 - la compilazione è analoga alla traduzione di un libro
 - l'interpretazione è analoga alla traduzione simultanea

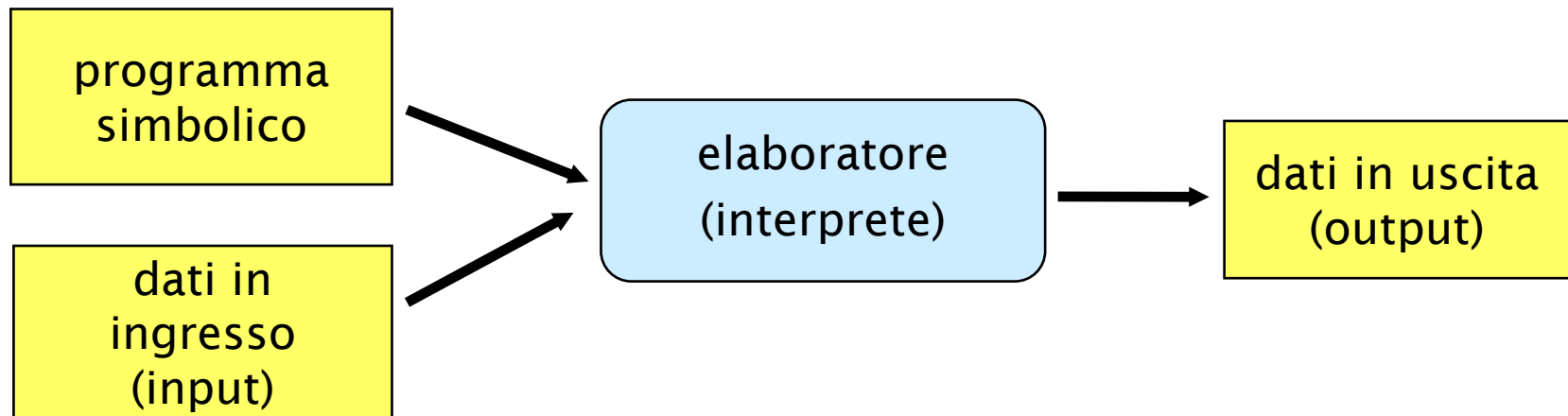
Compilazione

- ❑ Il calcolatore con un programma **compilatore**, ricevendo un **programma sorgente** in linguaggio simbolico esegue la traduzione producendo in uscita il corrispondente **programma eseguibile** in linguaggio macchina



Interpretazione

- ❑ Il calcolatore con un programma interprete, ricevendo un **programma sorgente** in linguaggio simbolico e i relativi dati in ingresso lo interpreta e lo esegue, istruzione dopo istruzione, producendo direttamente i dati in uscita



Confronto tra compilazione ed interpretazione

□ I linguaggi compilati

- prevedono la distribuzione di un file eseguibile (in linguaggio macchina) specifico per ogni piattaforma
- il file sorgente, che consentirebbe di modificare il programma, rimane di esclusiva pertinenza del programmatore

□ I linguaggi interpretati

- prevedono la distribuzione diretta del file sorgente (che è il medesimo per tutte le piattaforme)
- su ogni piattaforma deve essere disponibile il programma interprete

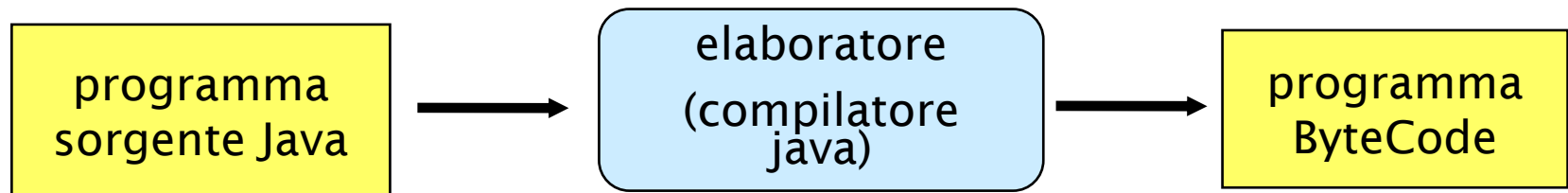
Il linguaggio Java

□ Java adotta una strategia mista

- il file sorgente contenente un programma Java viene prima compilato, generando un file in ByteCode
 - il file in ByteCode è un file in linguaggio macchina per una macchina non esistente, detta macchina virtuale o **Java Virtual Machine**
- per essere eseguito il ByteCode viene interpretato su ogni piattaforma
- esistono programmi interpreti, che realizzano una Java Virtual Machine, per ogni piattaforma
- in questo modo il file sorgente non viene distribuito, mentre il file ByteCode distribuito è il medesimo per tutte le piattaforme

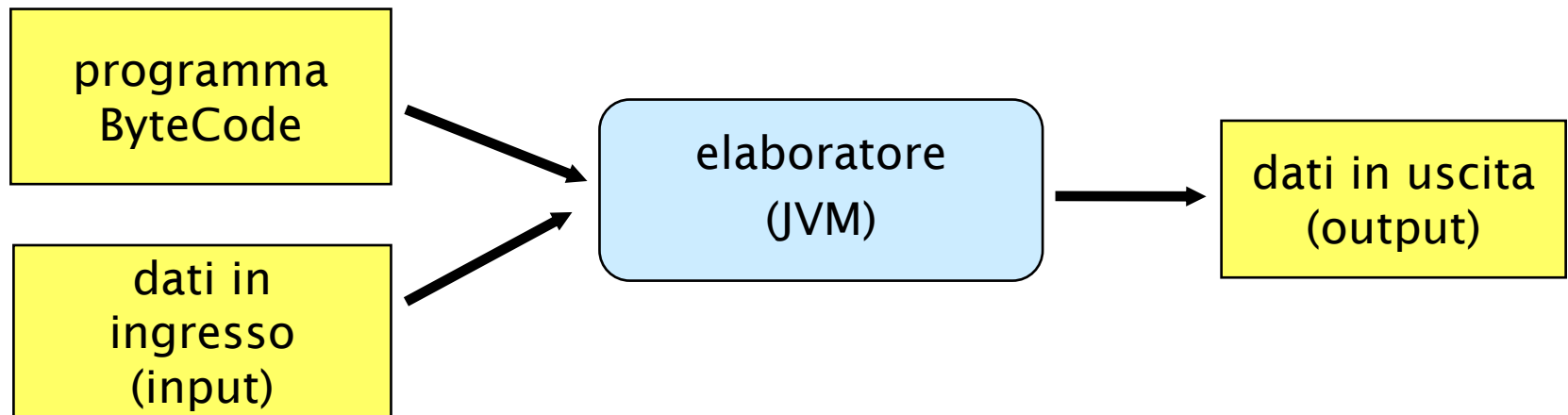
Compilazione di programmi Java

- ❑ Il compilatore con un programma **Compilatore Java**, ricevendo un programma sorgente in linguaggio Java esegue la traduzione producendo in uscita il corrispondente programma in un **linguaggio intermedio ByteCode**

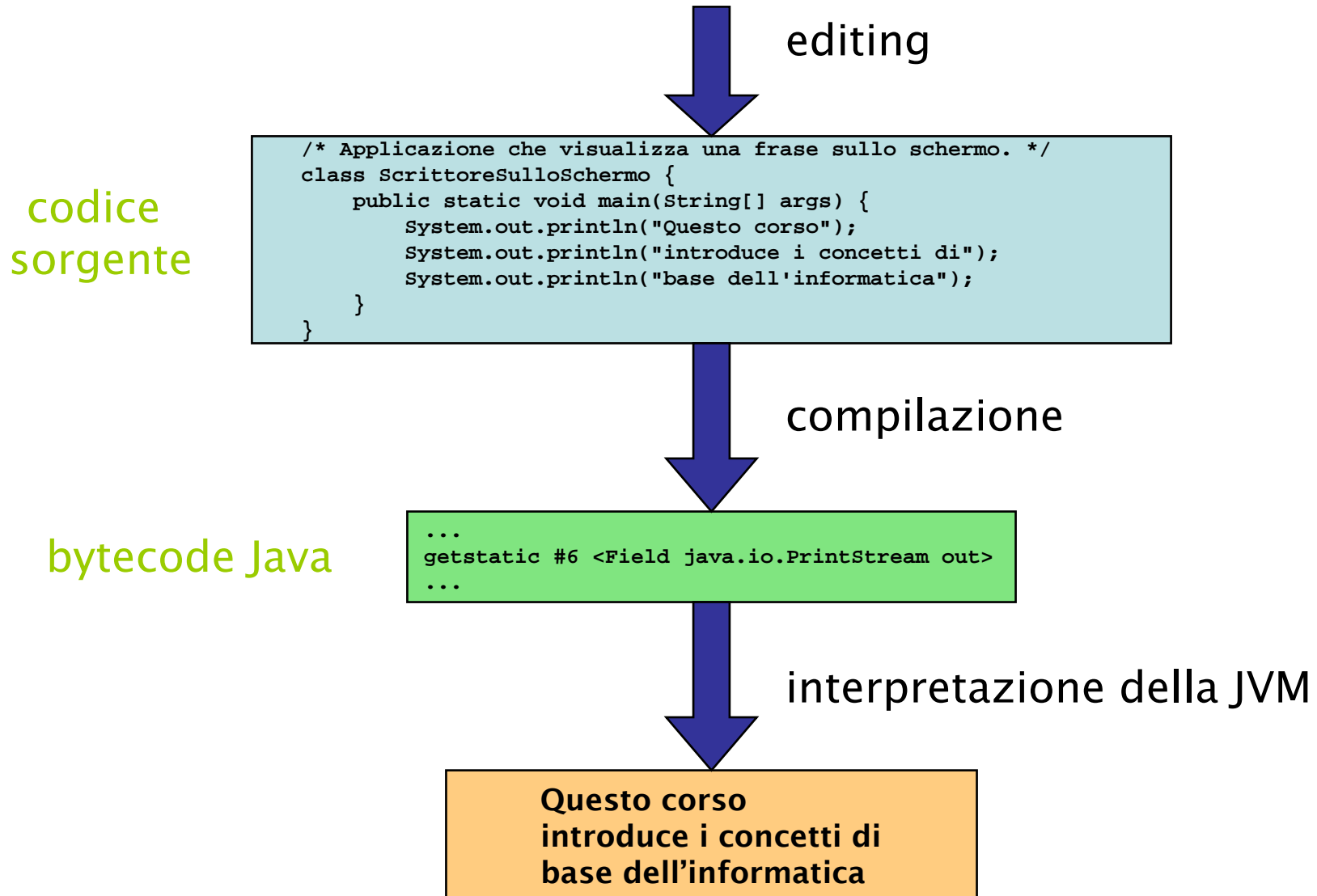


Interpretazione di programmi in ByteCode

- Il calcolatore con un programma interprete **Java Virtual Machine (JVM)**, ricevendo un **programma ByteCode** e i relativi dati in ingresso lo interpreta e lo esegue, istruzione dopo istruzione, producendo direttamente i dati in uscita



Riassumendo

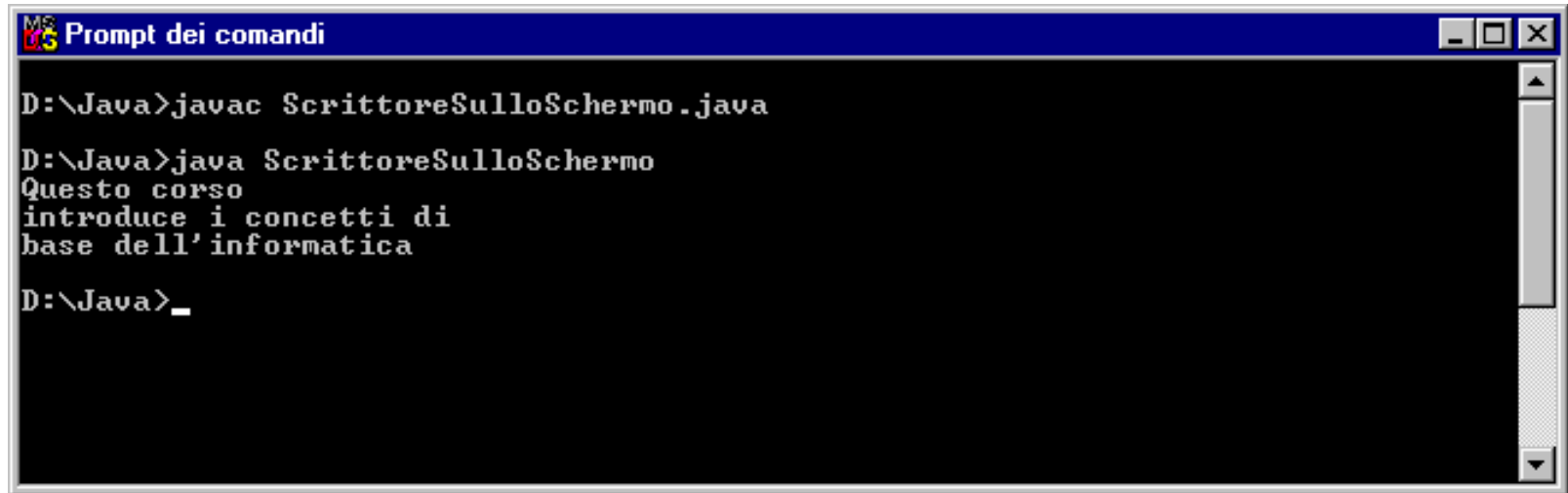


Che cosa fare in pratica — esecuzione

- ❑ Per eseguire una applicazione si deve usare il comando **java** di Java 2 SDK
 - per eseguire il comando **java** si deve specificare il nome della classe applicazione che si vuole eseguire
 - una applicazione Java viene eseguita come segue
 - viene inizializzata una macchina virtuale Java (JVM)
 - la JVM costruisce l'oggetto classe corrispondente alla classe applicazione
 - la JVM invia il messaggio **main(...)** a questo oggetto classe
- ❑ Per eseguire l'applicazione **ScrittoreSulloSchermo** è necessario eseguire il comando
 - `java ScrittoreSulloSchermo`
 - si osservi come il nome del metodo **main** non compare esplicitamente nel comando

Uso di Java 2 SDK

- La seguente figura mostra la compilazione ed esecuzione dell'applicazione **ScrittoreSulloSchermo** nella finestra del prompt dei comandi di Windows



```
MS-DOS Prompt dei comandi
D:\Java>javac ScrittoreSulloSchermo.java
D:\Java>java ScrittoreSulloSchermo
Questo corso
introduce i concetti di
base dell'informatica
D:\Java>_
```

- si osservi che **System.out** corrisponde alla finestra in cui viene eseguita l'applicazione

Ambienti integrati di sviluppo

- Un **ambiente integrato di sviluppo** (o **IDE**, che è l'acronimo di **Integrated Development Environment**) è una applicazione che permette di editare, compilare ed eseguire programmi nell'ambito di un unico ambiente

- Alcune funzionalità tipiche di un IDE Java sono le seguenti
 - editor (integrato) guidato dalla sintassi
 - sintassi evidenziata mediante colori
 - aiuto in linea e sensibile al contesto
 - editor visuale di interfacce grafiche (GUI)
 - accesso al compilatore e alla macchina virtuale Java mediante menu e/o bottoni
 - altri strumenti
 - browser della documentazione
 - debugger integrato
 - gestore di progetti e classi

Ambienti integrati di sviluppo per Java

□ Ci sono due tipologie di IDE per Java

- IDE dotati di una propria versione di Java SDK
 - Borland JBuilder Foundation,
<http://www.borland.com/jbuilder/foundation/>
 - Metrowerks CodeWarrior
 - Microsoft Visual J++
 - Symantec Visual Café
- IDE che non sono dotati di una propria versione di Java SDK — ma che ne richiedono una installazione separata
 - Kawa (Allaire, Tek-Tools) —
<http://www.allaire.com/products/kawa/>
 - Forte for Java, Community Edition (Sun) —
<http://www.sun.com/forte/ffj/ce/>
 - TextPad (Helios Software Solutions) — <http://www.textpad.com> — diversamente dai precedenti, TextPad è un editor di testi arricchito con alcune funzionalità degli IDE

Riferimenti al libro di testo

- ❑ Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al capitolo 4 (Strumenti per la programmazione)
 - 4.1 Strumenti per la programmazione
 - 4.1.1 Editing ed editor
 - 4.1.2 Compilazione e compilatori
 - 4.1.3 Esecuzione
 - 4.1.4 Compilatori e interpreti
 - 4.1.5 Compilazione ed esecuzione di programmi Java
 - 4.1.6 Java 2 Software Development Kit
 - 4.2 Che cosa fare in pratica
 - 4.4 Ambienti integrati di sviluppo