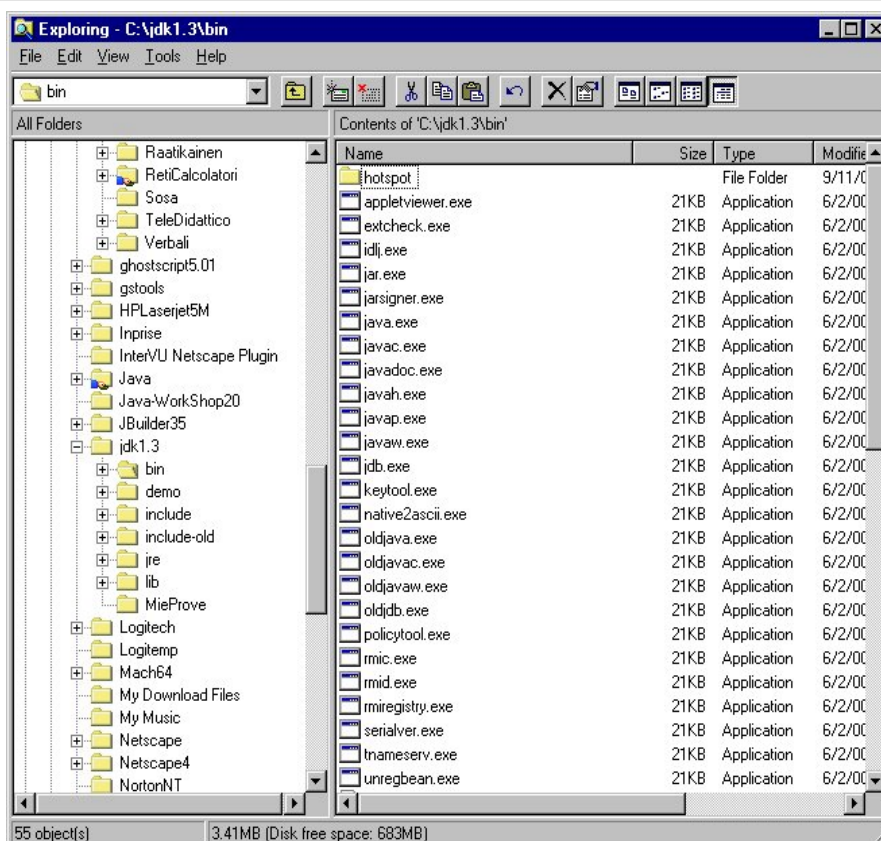


# Esercitazione n° 1

- Introduzione all'utilizzo di *Java Development Kit (JDK)* versione 1.3
- Sviluppare programmi Java tramite linea di comando
  - javac, java, jdb, javadoc
- Sviluppare programmi Java in ambienti integrati (*Integrated Development Environment - IDE*)
- Primo esempio di programma Java con due classi:
  - variante in un solo file
  - variante in due file distinti

## JDK1.3: il direttorio /bin



## Strumenti a linea di comando

---

- **Compilatore** - javac
- **Macchina virtuale** - java
- **Debugger** - jdb
- Generatore automatico di **documentazione ipertestuale** - javadoc
  
- Altri **comandi vari** - appletviewer, javap, rmic, ...
- Documentazione completa:  
[www.java.sun.com/j2se/1.3/docs/tooldocs/tools.html](http://www.java.sun.com/j2se/1.3/docs/tooldocs/tools.html)

## javac

---

- Legge *sorgenti Java* di classi e interfacce e li **compila** in *classfile* in formato *bytecode*
- Permette di compilare **classi singole** e **gruppi di classi**, anche mantenendo direttori separati per i file sorgenti e compilati
  
- **javac** [opzioni][sorgenti][@ElencoSorg]
- possibili opzioni:
  - **-classpath**: classi bootstrap, extension, poi *classpath* di utente (*variabile di ambiente e opzione*)
  - **-d** (direttorio per classi), **-g** (debugging abilitato), **-verbose**, ...

# java

---

- Mette in *esecuzione* una applicazione Java
- Avvia una macchina virtuale Java, *carica* una classe specificata e invoca il suo metodo *main* (*pubblico e statico*)
- **java** [opzioni] File.class [parametri]
- **java** [opzioni] -jar File.jar [parametri]
- varianti: javaw, oldjava, oldjavaw
- opzioni: -classpath/-cp, -jar, -verbose, -version, -?/  
-help, ...

# jdb

---

- Permette di eseguire il *debugging* (esecuzione *passo-passo*, *breakpoint*, osservazione di *variabili*, ...) di un programma Java
  - invoca una *nuova* macchina virtuale Java che esegue il programma in modalità debugging - **jdb** File.class
  - “aggancia” una macchina virtuale già in esecuzione (opzione Xdebug di **java**)
- **Comandi:** *help/?*, *run*, *cont*, *print* (mostra oggetti o tipi primitivi), *stop at/in* (posiziona breakpoint), *clear* (rimuove breakpoint), *step* (esecuzione passo-passo)

## javadoc

---

- Analizza le *dichiarazioni* e i *commenti* di un insieme di file sorgenti Java e produce la corrispondente **documentazione ipertestuale** (formato html), descrivendo *classi, interfacce, costruttori, metodi e campi*
  - Può essere invocato su *singole classi* o *interi package*

```
javadoc [opzioni][package][sorg][@ElencoS/P]
```

- Vari tag standardizzati per i commenti (@author, @param, @return, @throws, @version)

## Sviluppo di Programmi in Java IDE

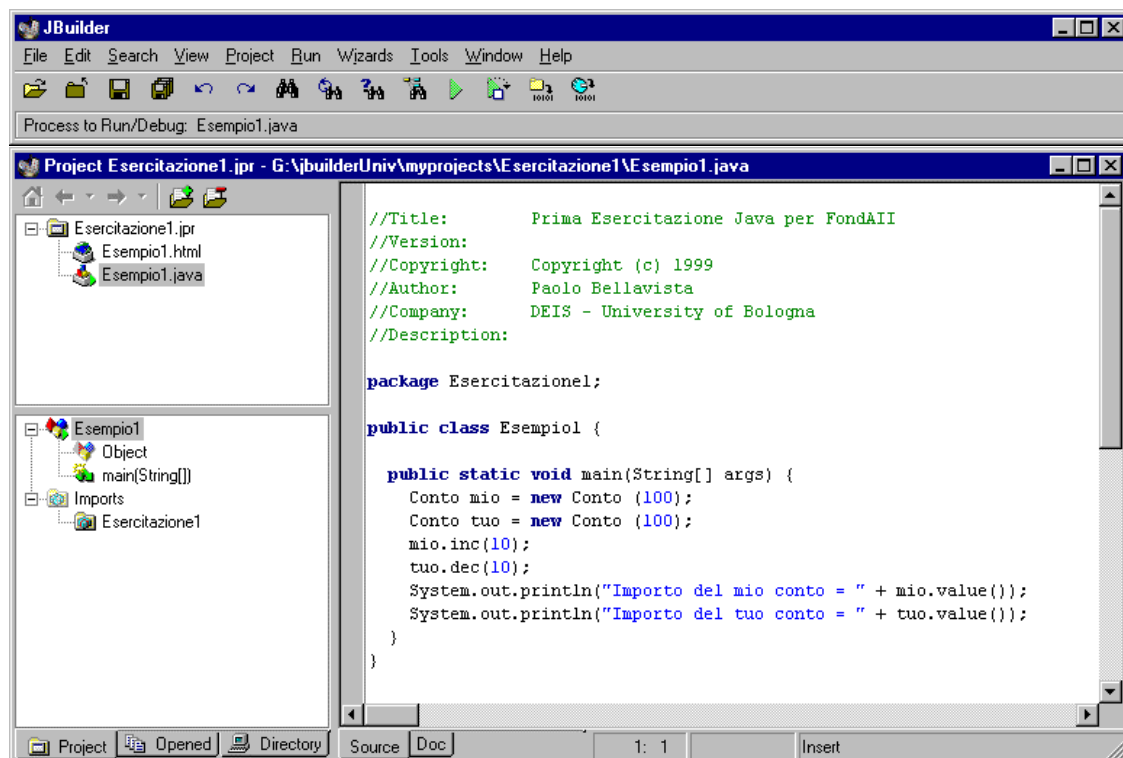
---

- Vari **editor** con *highlighting* della sintassi Java:
  - *TextTool, WinEdt, Jpad, ...*
- Vari ambienti di sviluppo integrati e visuali (**IDE**):
  - **freeware**      *Forte* - [www.sun.com/forte/](http://www.sun.com/forte/)  
                    *FreeBuilder* - [www.freebuilder.org](http://www.freebuilder.org)
  - **commerciali**    *JBuilder* - [www.borland.com/jbuilder/](http://www.borland.com/jbuilder/)  
                    *Kawa* - [www.allaire.com/products/kawa/](http://www.allaire.com/products/kawa/)  
                    *CodeWarrior* - [www.metrowerks.com](http://www.metrowerks.com)  
                    *VisualAge* - [www-4.ibm.com/software/ad/](http://www-4.ibm.com/software/ad/)

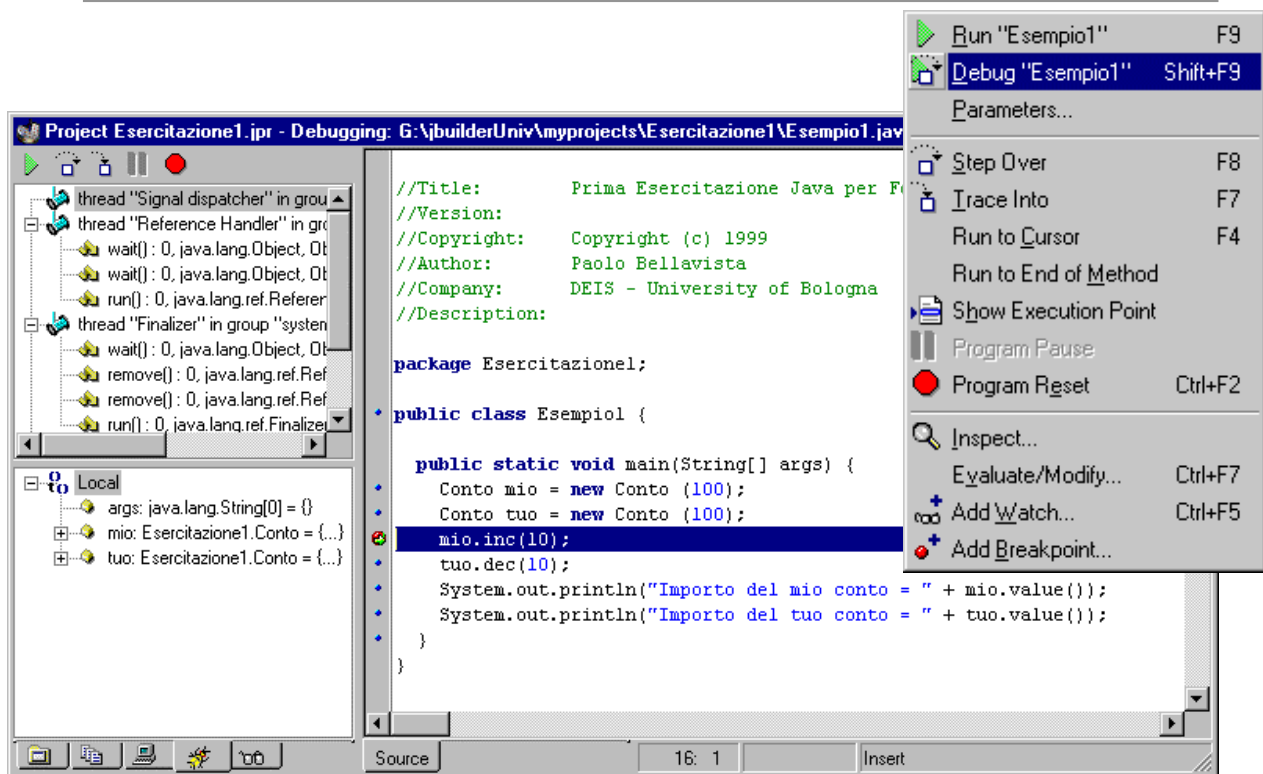
# Ambienti di Sviluppo Integrati (IDE)

- Tutti supportano il concetto di *progetto*
- *Editor + Compilatore + JVM + Debugger + Navigazione* file e documentazione
- Disponibili in Laboratorio:
  - strumenti a linea di comando *JDK 1.3*
  - *Borland JBuilder 3.0 University Edition*
- Primo esempio di programmazione: la classe `Esempio1` e la classe `Conto`

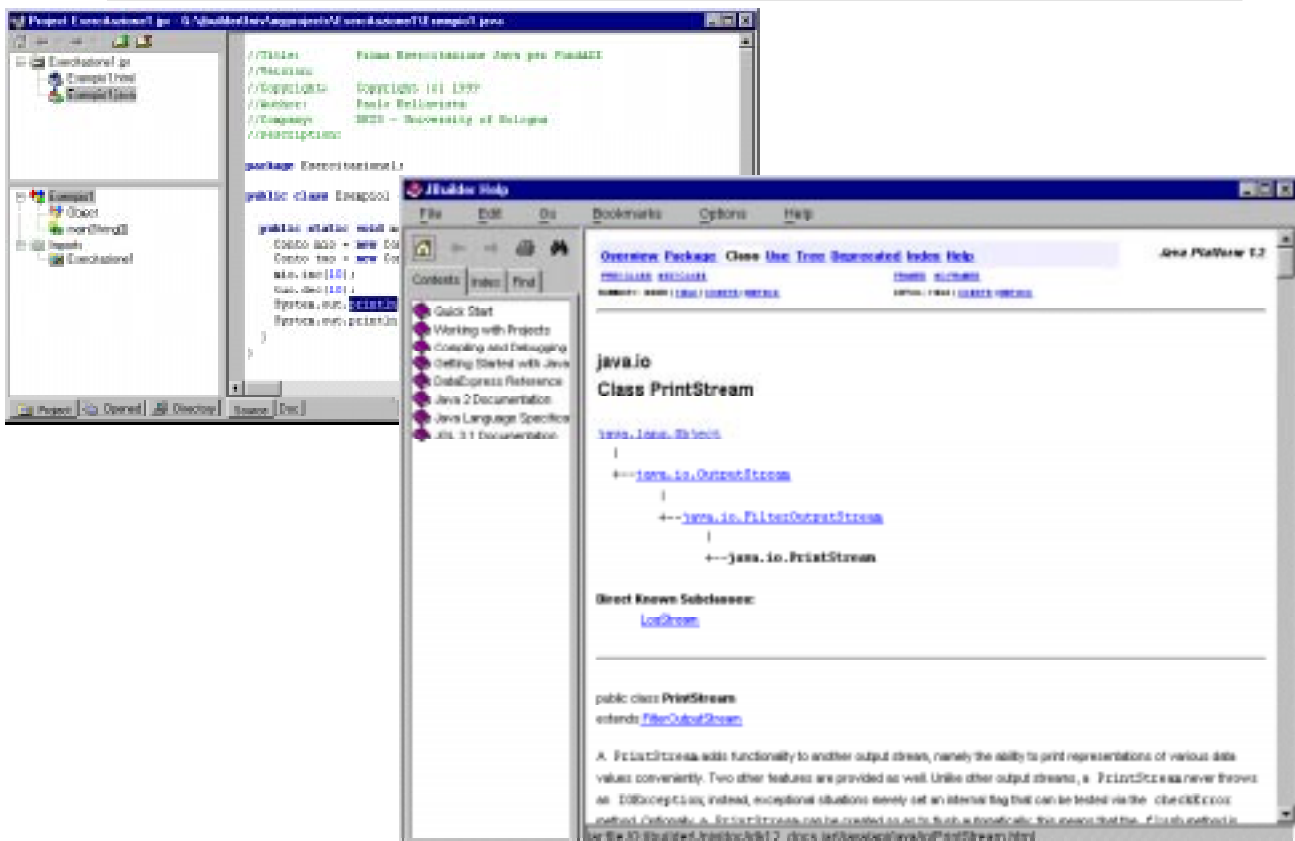
## JBuilder: creazione di un progetto



# JBuilder: strumenti di debugging



# JBuilder: context-sensitive help



# Esercitazione n°1: la classe Conto

---

```
//Title:      Prima Esercitazione Java per FondAII
//Version:    1.0
//Copyright:  Copyright (c) 2000
//Author:     Paolo Bellavista
//Company:    DEIS - University of Bologna
//Description: Primo esempio di applicazione Java - Classe Conto
```

```
package Esercitazione1;

public class Conto {

    private int i=0;
    public Conto(int j) { i=j;
    }
    public void inc(int j) { i=i+j;
    }
    public void dec(int j){ i=i-j;
    }
    public int value() { return i;
    }
}
```

# Esercitazione n°1: la classe Esempio1

---

```
//Title:      Prima Esercitazione Java per FondAII
//Version:    1.0
//Copyright:  Copyright (c) 2000
//Author:     Paolo Bellavista
//Company:    DEIS - University of Bologna
//Description: Primo esempio di applicazione Java - Classe principale
```

```
package Esercitazione1;

public class Esempio1 {

    public static void main(String[] args) {
        Conto mio = new Conto (100);
        Conto tuo = new Conto (100);
        mio.inc(10);
        tuo.dec(10);
        System.out.println("Importo del mio conto = " + mio.value());
        System.out.println("Importo del tuo conto = " + tuo.value());
    }
}
```