



Laboratorio di Programmazione

Laurea in Bioinformatica

Web: <http://www.scienze.univr.it/fol/main?ent=oi&id=34121>

Docente: *Carlo Drioli*
Email: *drioli@sci.univr.it*

2007/2008

Lucidi a cura di
Nicola Drago, Carlo Drioli

Il linguaggio JAVA: compilazione ed esecuzione

Sommario

- Compilare ed eseguire codice JAVA
- Scrittura di codice
- Primi esempi

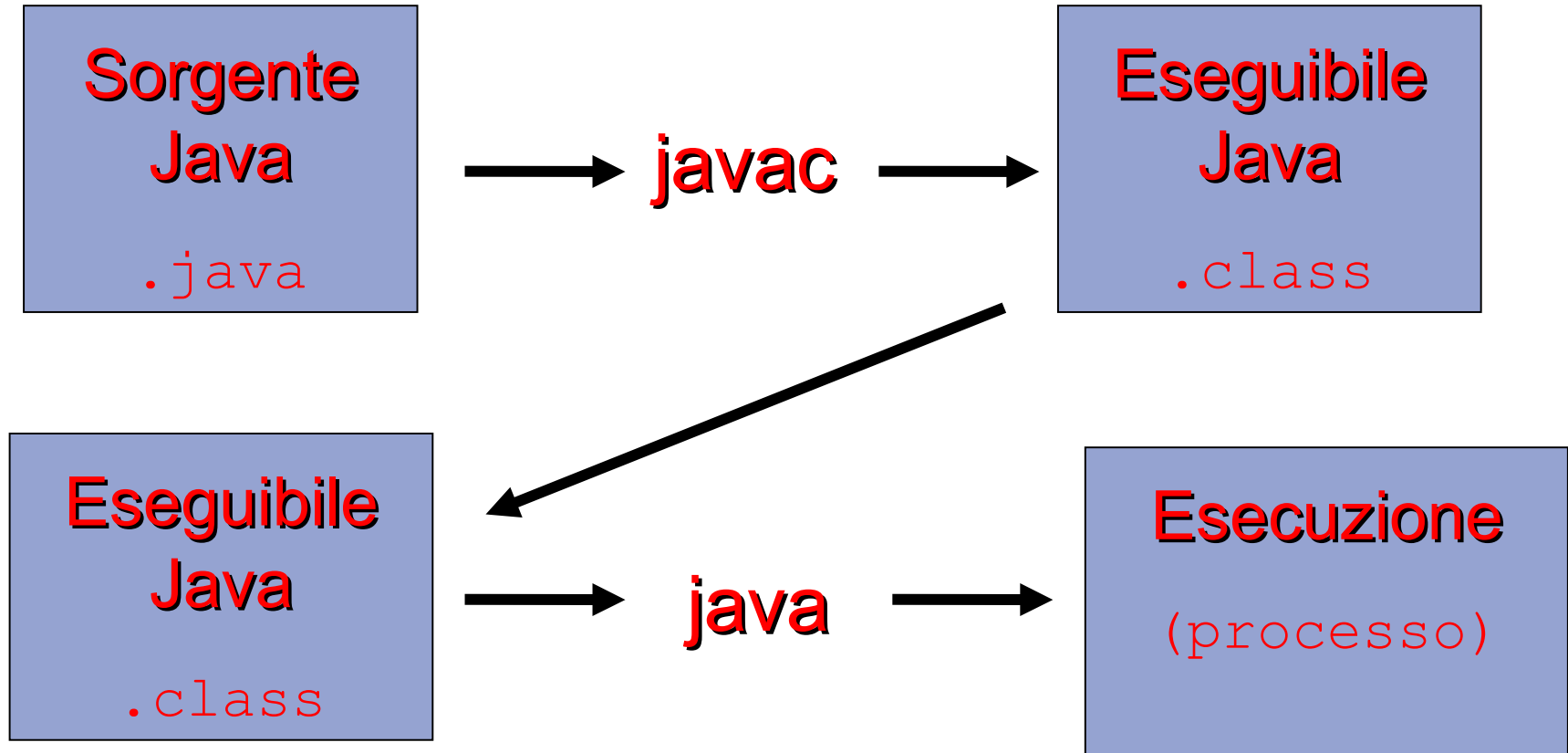
Il linguaggio JAVA

- Java è stato creato da *Sun Microsystems* ed è disponibile liberamente su <http://java.sun.com/>
- E' un linguaggio di programmazione orientato agli oggetti, derivato dal C++
- E' indipendente dalla piattaforma: un programma Java è compilato in un formato intermedio (bytecode) uguale per tutte le piattaforme.
- Il bytecode è interpretato da una Java Virtual Machine (JVM).
- Per lo sviluppo, Sun mette a disposizione un System Developer Kit (SDK)

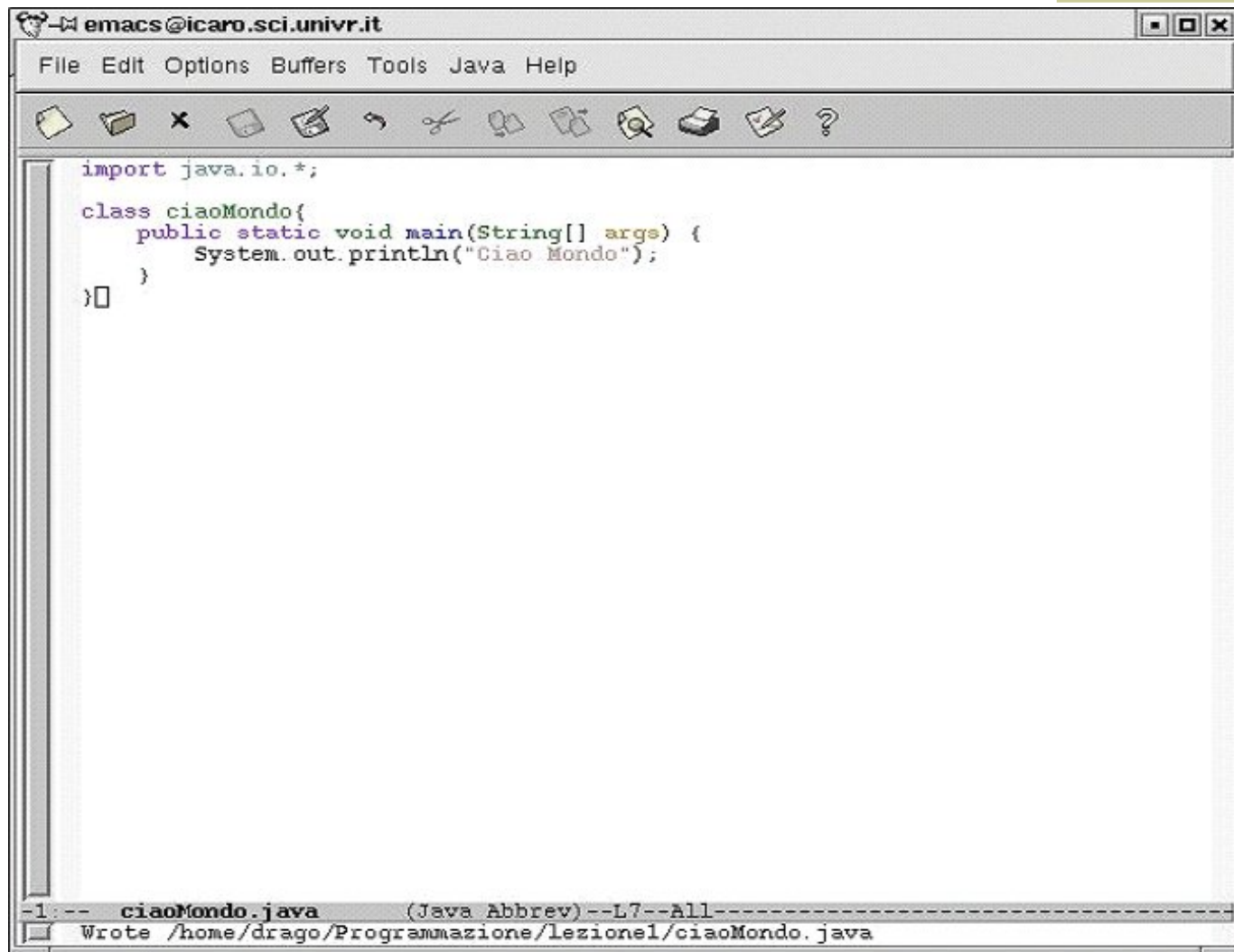
Compilare ed eseguire codice JAVA

- “javac”: compila il file sorgente producendo un file eseguibile in formato bytecode
- “java”: esegue il bytecode
- I sorgenti hanno estensione .java
- Gli eseguibili hanno estensione .class
- I file di libreria hanno estensione .jar

Compilare ed eseguire codice JAVA



Emacs: editing di sorgenti java



The screenshot shows the Emacs editor window with the title bar "emacs@icaro.sci.univr.it". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Java", and "Help". The toolbar contains icons for file operations and editing. The main text area displays the following Java code:

```
import java.io.*;

class ciaoMondo{
    public static void main(String[] args) {
        System.out.println("Ciao Mondo");
    }
}
```

The status bar at the bottom shows the current file "ciaoMondo.java" and the line number "1". The buffer name is "(Java Abbrev)--L7--All". The file path is "/home/drago/Programmazione/lezionel/ciaoMondo.java".

Scrivere il codice

Il codice deve essere:

- leggibile (!)
 - Ordinamento verticale delle istruzioni
 - Indentazione delle righe
- sintatticamente corretto
 - Aiuto dall'editor (o ambiente di sviluppo)
 - Messaggi dal compilatore
- semanticamente corretto
 - Warning del compilatore
 - Esperienza del programmatore

Uso dei commenti

Per aumentare la leggibilità il codice va commentato:

- `//` Commento in riga
`/*` Commento di un'area `*/`
- `//` Dichiarazione di variabile
`Int I;`
- `/*`
Programma per calcolare
l'area del rettangolo
`*/`
`class rettangolo{`
`.....`
`}`

Il primo programma

Per creare un programma è necessario:

- Dare un nome al nostro programma
- Dare lo stesso nome al file .java che contiene la classe (vedremo che cos'è) che definisce il programma

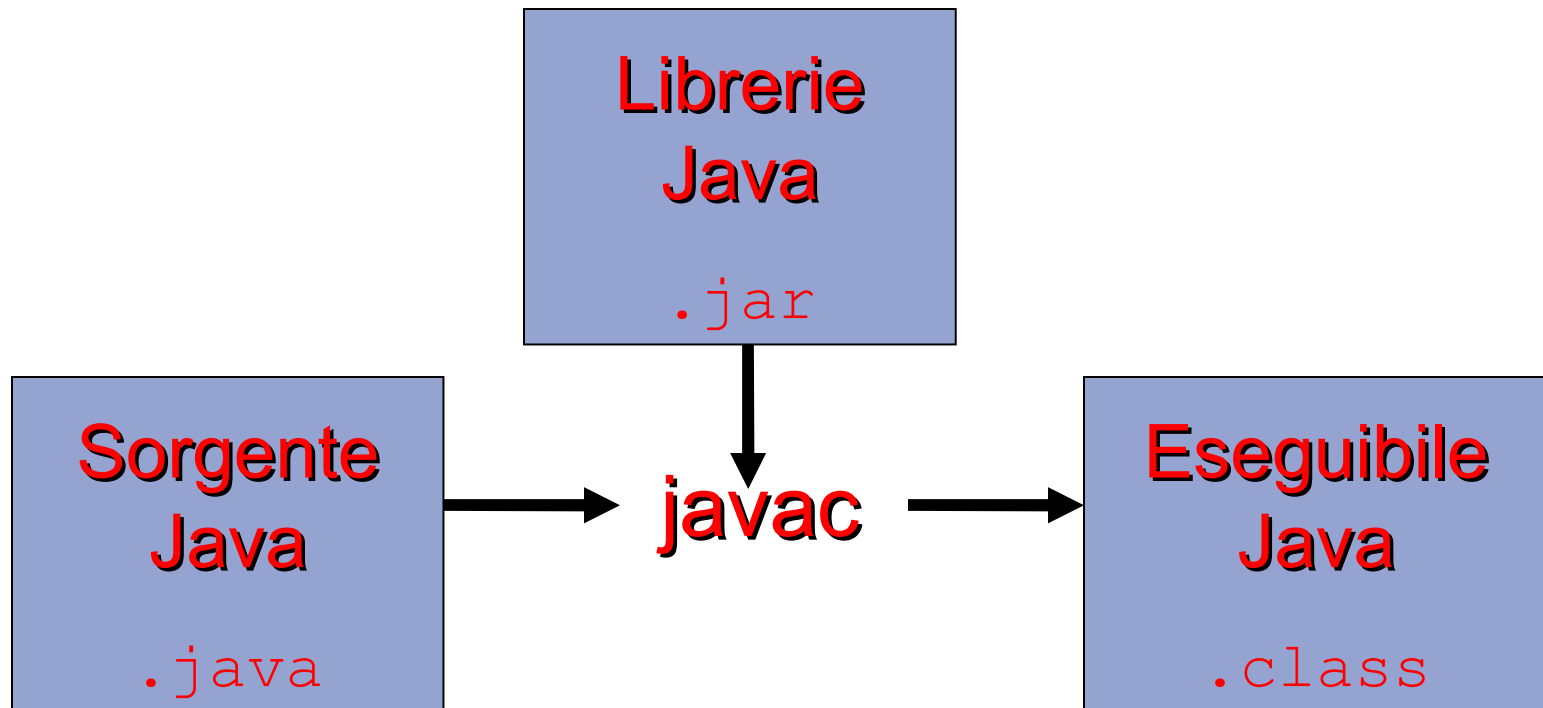
```
class ciaoMondo {  
...  
//corpo della classe  
...  
}
```



file ciaoMondo.java

Il primo programma

Le librerie contengono programmi utili, che per semplicità il sistema ci mette a disposizione.



Il primo programma

- Per compilare anche un programma elementare occorrono funzionalità di libreria:

```
import java.io.*
```

- E' necessario dichiarare un metodo di nome "main"

```
public static void main(String args[]){
```

```
...
```

```
//corpo del programma
```

```
...
```

```
}
```

- Scrivere le istruzioni necessarie per far svolgere al programma il compito desiderato

```
System.out.println("CiaoMondo");
```

Il primo programma

I comandi per la compilazione e l'esecuzione di un programma sono:

- **Compilazione del programma**

```
javac CiaoMondo.java
```

- **Esecuzione del bytecode**

```
java CiaoMondo
```

Il primo programma

```
import java.io.*;

class CiaoMondo{
    public static void main(String[] args){
        System.out.println("Ciao Mondo\n");
    }
}
```

Esercizi

- Scrivere ed eseguire un programma che quando viene eseguito visualizzi:

Ciao sono il programma di <Nome Cognome>

- Modificare il programma precedente ed eseguirlo in modo che quando viene eseguito visualizzi:

Ciao sono il programma di <Nome Cognome>
Questo è il mio secondo programma

Input da tastiera

- Si vuole assegnare un valore a una variabile, inserendolo da tastiera in fase di esecuzione del programma.
- L'input da tastiera richiede operazioni complicate o l'utilizzo di una libreria con funzioni estese. Noi useremo la classe semplificata “**ConsoleInputManager**” contenuta nella libreria “**prog**” fornita a corredo del teso di riferimento.

Input da tastiera

Istallazione della libreria “prog” e altre modifiche delle variabili d’ambiente

- Copiare il file “prog.tgz” in una directory della propria home (ad es. `local_java_dir`)
[http://mordente.sci.univr.it/samba/Public/LabJava_temp/prog.tar]
- Decomprimere il file mediante il comando `tar -xzf prog.tgz`
- Modificare il file `.bashrc` nella propria home aggiungendo le seguenti righe:

```
CLASSPATH=$JAVAPATH/lib:../home/accounts/.../<ACCOUNT>/<LOCAL_JAVA_DIR>  
export CLASSPATH
```


Input da tastiera

Uso della classe di input da tastiera della libreria “prog”

- Direttiva di inclusione della libreria

```
import prog.io.*;
```

- Costruzione di un oggetto `ConsoleInputManager` di nome *tastiera*

```
ConsoleInputManager tastiera = new ConsoleInputManager(),
```

- Invocazione del metodo *readInt* dell'oggetto *tastiera* per la lettura di un intero

```
int numero = tastiera.readInt();
```

Input da tastiera

Un semplice programma si test della libreria:

```
import prog.io.*;

class Input{
    public static void main(String[] args) {
        ConsoleInputManager tastiera = new ConsoleInputManager();
        int numero;

        System.out.print("Digita un numero: ");
        numero = tastiera.readInt();
        System.out.print("Hai inserito il numero: ");
        System.out.println(numero);
    }
}
```