
Ambiente di sviluppo Java

Prendiamo confidenza

1. Login

2. Compilatore Java

- linea di comando (javac, java)
- ambiente integrato

Eclipse

(consigliato + utilizzato in laboratorio nelle esercitazioni)

Compilazione (da linea di comando)

- Scrivere il programma con un editor di testo
 - Salvarlo con un nome, es.: *Hello.java*
- Compila digitando **javac Hello.java**
 - La compilazione crea il *bytecode* nel file *Hello.class* che contiene le istruzioni per la macchina virtuale oltre ad altre informazioni
- Esegui digitando **java Hello**
 - Un interprete Java carica il bytecode del programma (e delle eventuali librerie necessarie) ed avvia il programma
 - Per localizzare le classi (standard, fornite con ambiente di sviluppo Java), settare la variabile CLASSPATH o usare opzione *-classpath*

java nomeFile -classpath pathClassi

Eclipse

- <http://www.eclipse.org/>
 - Editor fornito dal tool
 - Codice raggruppato per progetti
 - Ogni progetto può contenere diversi pacchetti
 - Esegui premendo il tasto run
 - Compila ed esegue
 - Salvare il file prima di eseguire il programma
-

Il primo programma

File HelloTester.java

```
public class HelloTester
{   public static void main(String[] args)
    {
//visualizza sulla finestra di console: Hello, World
System.out.println("Hello, World!");
    }
}
```

- NOTA:**
- JAVA distingue tra maiuscolo e minuscolo
 - nome file = nome classe
-

Prime nozioni di Java

- Ogni programma Java è costituito da una o più classi
 - `public class HelloTest`
 - definisce una classe (blocco costruttivo fondamentale dei programmi Java)
 - Ogni classe contiene la definizione dei metodi
 - Ogni metodo contiene istruzioni e decisioni
 - come le funzioni nella programmazione strutturata
 - `public static void main(String[] args)`
 - definisce un metodo speciale chiamato `main`
 - Ogni programma Java contiene un metodo **main**.
 - Quando viene eseguito un programma, l'esecuzione parte dal metodo **main**
-

Ancora

- `//` visualizza sulla finestra di console: `Hello, World`
 - Commenti di riga cominciano per `//`
 - Un metodo si invoca specificando un oggetto, il nome del metodo e i suoi parametri
 - `System.out.println("Hello, World!")`
 - Invoca il metodo `println` sull'oggetto `out`:
 - definito nella classe `System` (oggetti e metodi per accedere alle risorse di sistema – già implementato)
 - istanza della classe `PrintStream` (metodi di input/output – già implementato)
 - Il parametro passato è la stringa "Hello, World!" (oggetto della classe `String`)
-

PrintStream e System

Classe System

....

....

....

PrintStream out = new PrintStream();

...

....

...



Classe PrintStream

....

....

....

..... println(...)

..... print(.....)

...

...

Istanziare

Classe String

....

....

....

..... length(...)

..... toUpperCase(.....)

...

...

(forma, stampo,
template,etc)

Oggetto "Hello, world"

....

....

....

..... length(...)

..... toUpperCase(.....)

...

...

Oggetto "Ling. di Prog. 2"

....

....

....

..... length(...)

..... toUpperCase(.....)

...

...

Altro programma

File SplitTest.java

```
public class SplitTest
{
    public static void main(String[] args)
    {
        String sentence = "Mary had a little lamb";
        String[] result = sentence.split("\\s");

        for (int x=0; x<result.length; x++)
            System.out.println(result[x]);
    }
}
```

Analizziamo il programma

- `String sentence = "Mary had a little lamb";`

definisce una variabile di tipo `String` contenente l'oggetto stringa

```
Mary had a little lamb
```

- `String[] result`

definisce una variabile di tipo "array di oggetti `String`" di nome `result`

- `split` è un metodo della classe `String` che spezza una frase a seconda del parametro passato

- `"\\s"` indica di spezzare la stringa in corrispondenza degli spazi

- ```
for (int x=0; x<result.length; x++)
 System.out.println(result[x]);
```

ciclo che stampa il contenuto di `result` un elemento per ogni riga

- Dove prendiamo le informazioni sulle classi delle librerie Java?

---

# Documentazione classi Java

- Specifica API (***A**pplication **P**rogramming **I**nterface*)
  - Tutte le informazioni sulle classi e i pacchetti
  - Possiamo installarla:
    - si accede da pagina index.html directory docs dell'installazione di J2SE
  - oppure consultarla on-line
    - <http://docs.oracle.com/javase/7/docs/api/>
-

---

# Esercizi

- Usare metodi `length()`, `concat(String)` e `toUpperCase()` su stringhe e verificare il risultato in output
  - Scrivere un programma che scriva “Hello, world!” sulla prima riga e il vostro nome e cognome sulla seconda.
  - Scrivere un programma che data la stringa  
`Mary had, a little lamb.`  
stampa le parole `Mary`, `had`, `a`, `little` e `lamb` una per riga (utilizzare metodo `split`)
  - Scrivere un programma C per simulare il funzionamento di un semplice conto corrente bancario
-