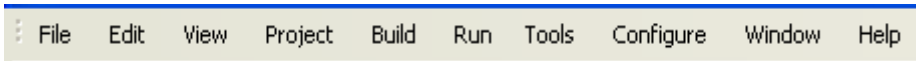
Guida all'uso dell'ambiente di sviluppo[1] integrato o IDE

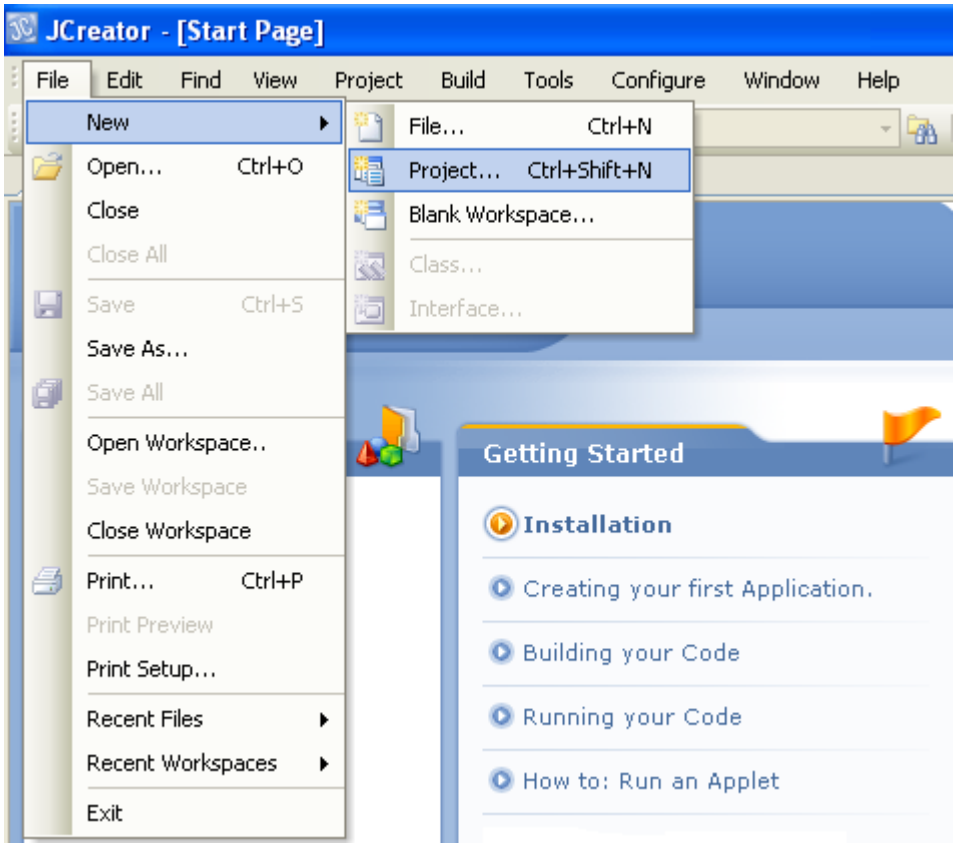
JCreator LE 4.50

Inizializzazione: creazione del nuovo *progetto* e del *file sorgente* in ambiente JCreator

Al lancio del programma si apre la finestra tipica delle applicazioni in ambiente Windows, che presenta la barra dei menù:

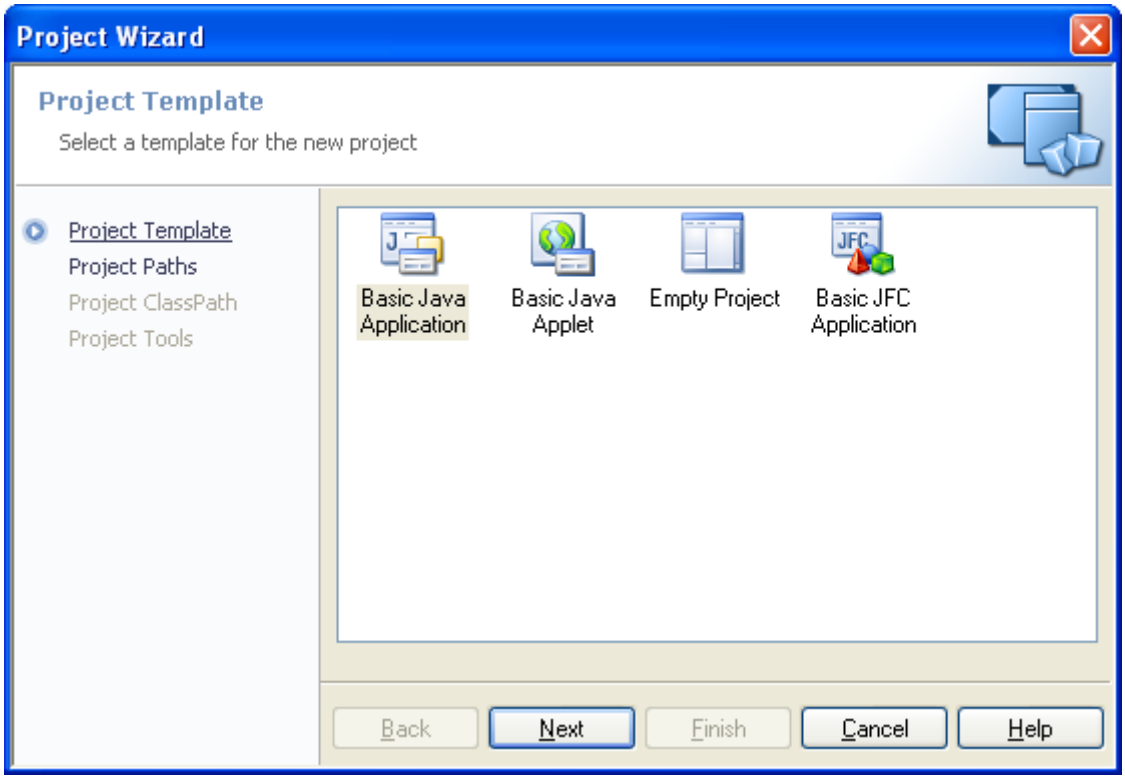


- Selezionando una prima volta i comandi **File → New** si scelga **Project**

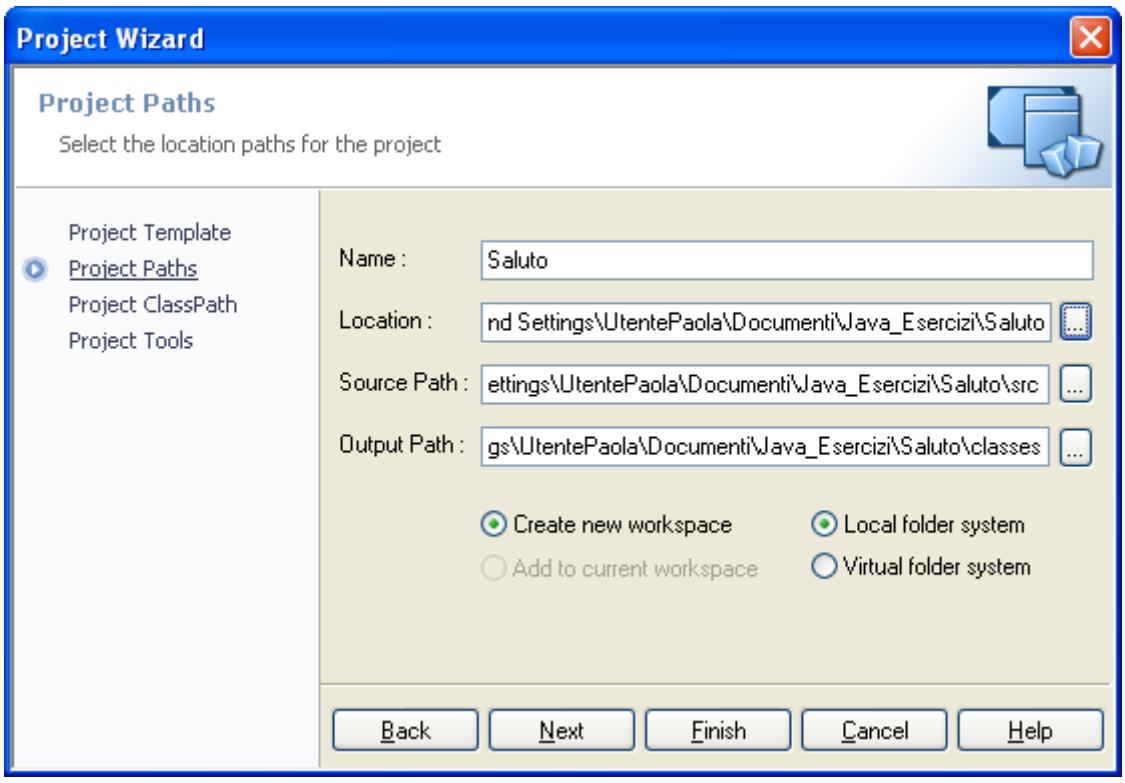


[1] Tra i livelli di funzionalità di un calcolatore si definisce **software di sviluppo** quell'insieme di programmi che consentono di sviluppare le applicazioni definite dall'utente: editor, compilatori, interpreti e linker-loader cioè programmi capaci di tradurre sequenze di istruzioni formulate in linguaggio di programmazione in sequenze di numeri in un codice comprensibile dal calcolatore. Tale software è un esempio di software di base

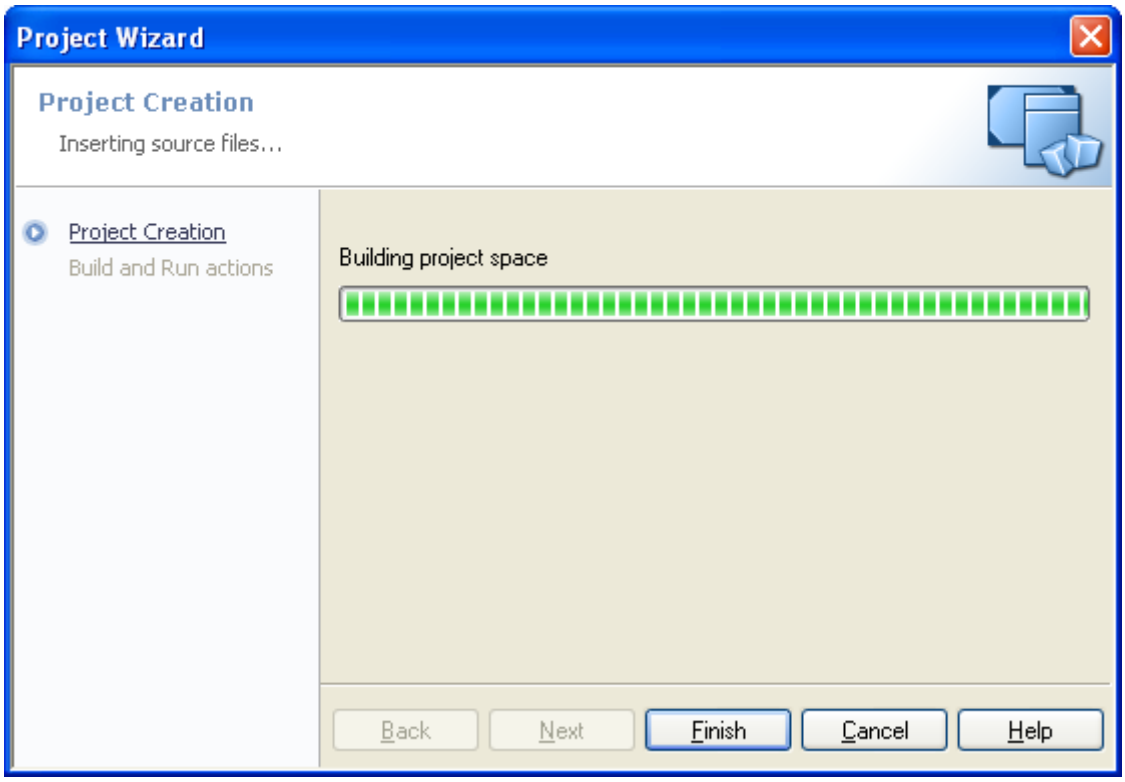
- si crei il progetto di tipo **Basic Java Application**



- si scelga un nome significativo impostando il corretto percorso (creando una cartella in documenti ad esempio Java_Esercizi)



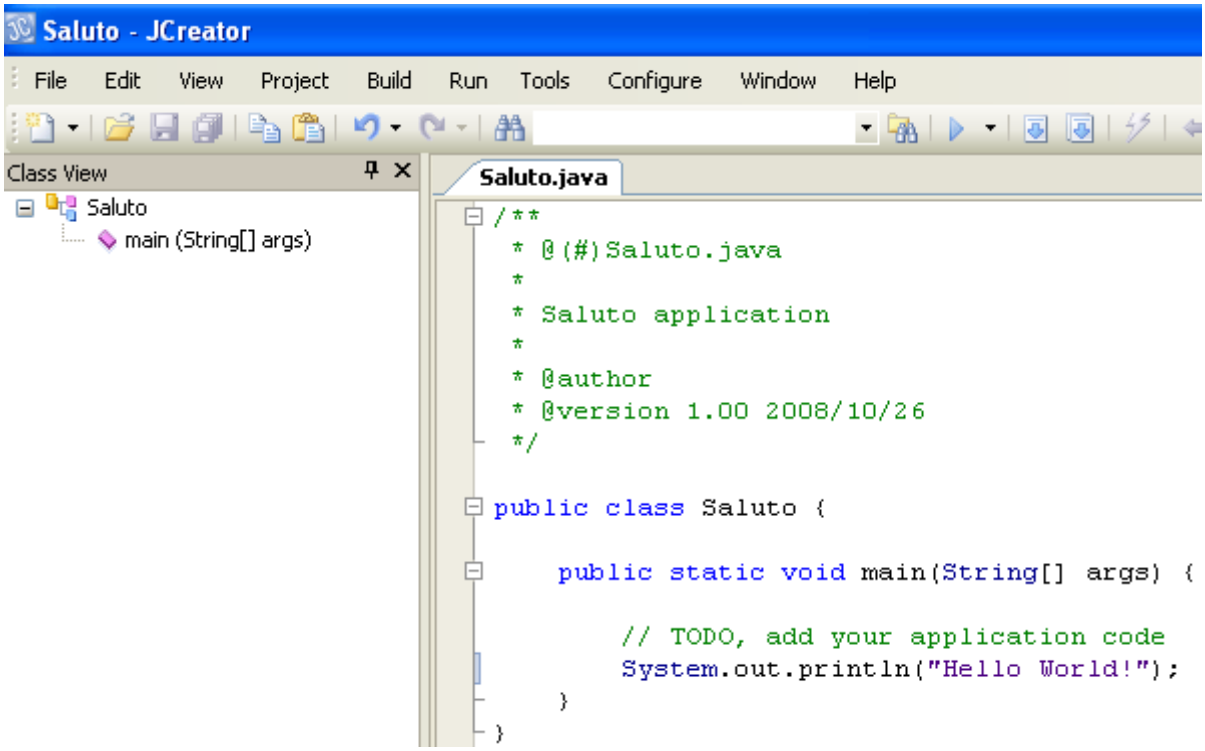
- si prema Finish



- si prema Finish nuovamente

Prima FASE: Editing

uso dell'**editor integrato** per modificare il ***file sorgente*** che ha lo stesso nome della classe (ogni applicazione contiene la definizione di almeno una ***classe***: la descrizione generale delle caratteristiche di un *esecutore* ad esempio per visualizzare un Saluto) ed estensione ***.java***

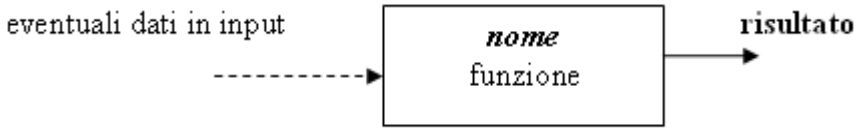


sono automaticamente inseriti:

- il **metodo principale**

  main()

  è una **funzione**[2] che individua la prima istruzione da eseguire ed il flusso principale delle successive operazioni.



  Tale metodo è visibile a tutte le altre classi (*public*), è associato alla classe (*static*) e sarà "lanciato in esecuzione" dal SO (senza dover definire oggetti a cui richiederlo come servizio), non restituisce valori (è di tipo *void*) ed è previsto di potergli "passare" al momento dell'esecuzione (***run time***) come argomenti delle *stringhe* organizzate in modo efficiente:

  public static void main (String**[]** args)

- uso dell'oggetto di sistema, di tipo canale (***stream*** o flusso) di nome **System.out** che permette di prelevare dalla RAM e **scrivere su monitor** una *stringa* (sequenza di caratteri) ed andare a capo, grazie al metodo:

  println("*stringa*")

  la stringa inserita per prova è *Hello World!*

- una **intestazione** a commento compresa tra i simboli /** e */
  in cui aggiungere il nome dell'autore del programma
  che sarà utilmente usata per *autodocumentare* con il programma *javadoc*

  nb: un altro modo per inserire commenti su più linee sarà iniziare con /* e terminare con */

- un **commento su unica linea** preceduto dal simbolo //

  // TODO, add your application code

  che indica dove inserire il codice della propria applicazione

Per "default" sono visualizzati sia l'ambiente di progetto (*File View* e *Class View*), sia l'Output per informazioni sull'esito delle varie fasi nella costruzione del programma, sia il foglio in cui scrivere.

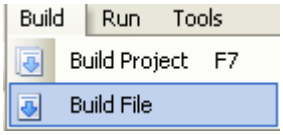
---

[2] Una **funzione** è un *modulo di programma* che può essere inteso come una "scatola nera" capace di realizzare una specifica elaborazione fornendo sempre un risultato (se la funzione non restituisce un valore è detta *void*) potendo utilizzare dati che gli vengono passati, ad esempio come valori, al momento della *chiamata* da programma.

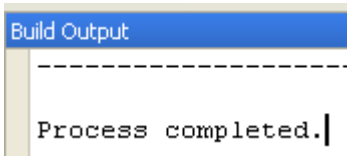
Seconda FASE: compilazione

traduzione in **bytecode** (meta-codice binario universale) con **controllo della sintassi** del linguaggio Java, eventuale importazione di package (librerie con definizione di classi riutilizzabili) con salvataggio su file che viene memorizzato con estensione *.class* con percorso settato

- Selezionando il comando **Build File**



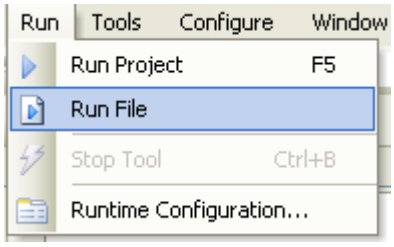
in assenza di errori, compare avviso di elaborazione completata



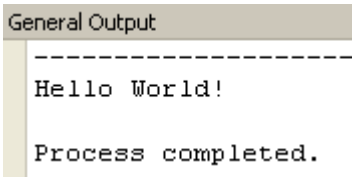
Terza FASE: interpretazione

traduzione simultanea del **bytecode** (meta-codice universale) da parte della **J**ava **V**irtual **M**achine cioè generazione del codice specifico per la piattaforma su cui viene eseguito il programma caricato in RAM

- Selezionando il comando **Run File**

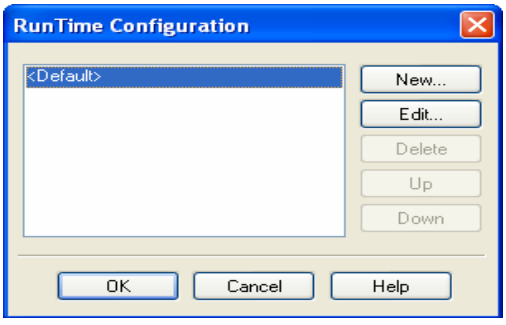


in assenza di errori, compare il risultato:
la visualizzazione della stringa *Hello World!*
su video nella sezione General Output



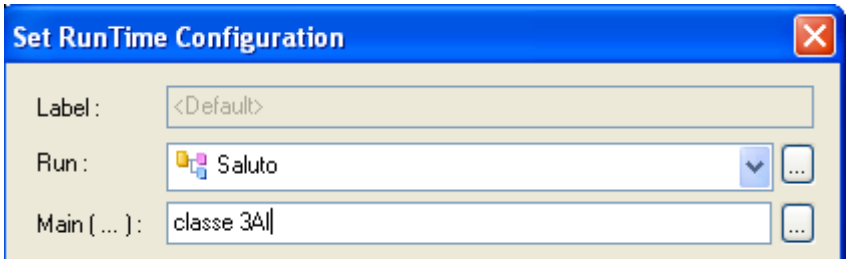
Per "passare" al momento dell'esecuzione (***run time***) delle stringhe da elaborare, prima di eseguire, si configura l'ambiente:

- Selezionando il comando **Runtime Configuration...**
  cliccare <Default>



ed **Edit**are le *stringhe*

che si desiderano come argomenti del main(...)

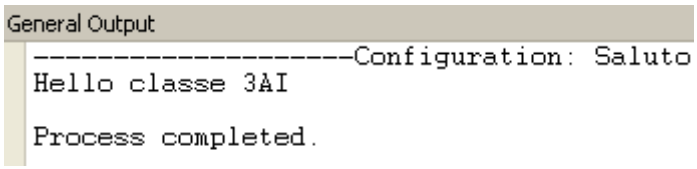


Infine confermare (premendo **OK** in entrambe le finestre)

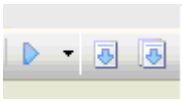
Ad **esempio** il codice seguente:

```
System.out.println("Hello " + args[0] + " " + args[1]);
```

produrrà il seguente effetto:



Si possono, in alternativa, usare le opportune[3] **icone**:

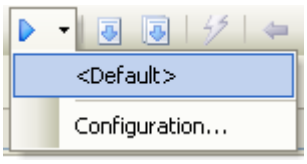


- per realizzare la fase di COMPILAZIONE del File

- per realizzare la fase di COMPILAZIONE del Progetto

- per realizzare la fase di **interpretazione** al ***run time*** :

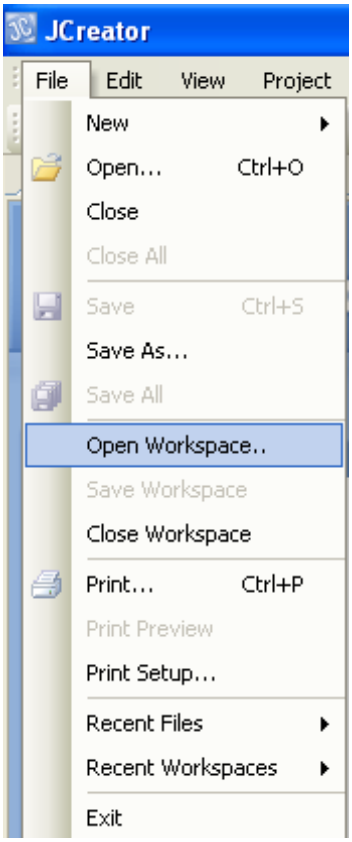
  - in configurazione di *default*
  - impostando una diversa *configurazione* ad esempio presettando gli argomenti da passare al ***run time***



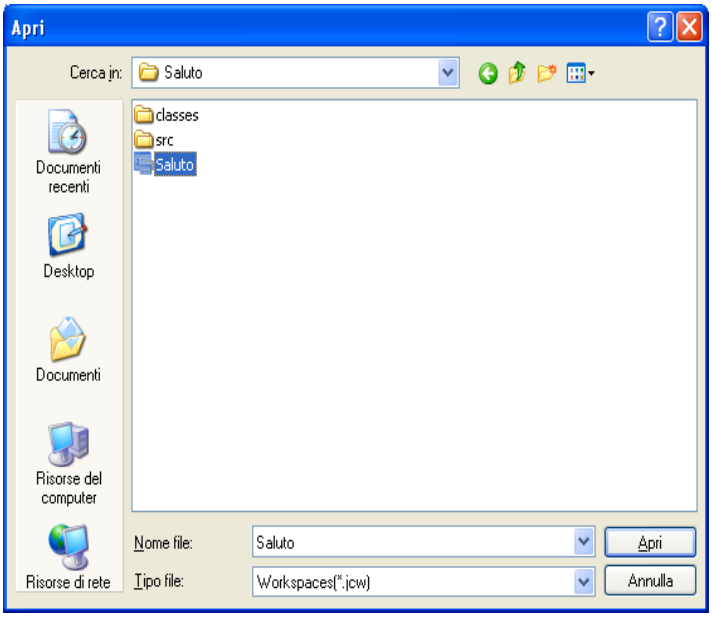
[3] Nella versione Professional è prevista opzione di debugging **Build → Start Debug**

Sviluppare programmi nell'ambiente di lavoro

- **Aprire** non il singolo file ma il **Workspace**



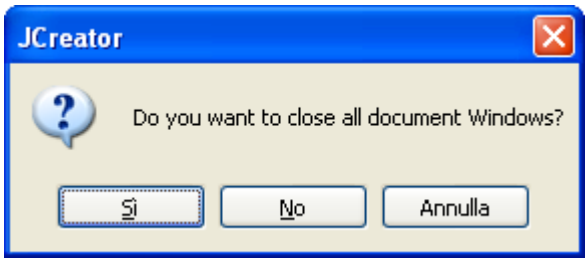
selezinando l'opportuno **Workspace** (file.jcw)



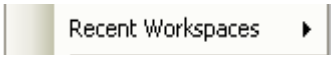
- **Salvare** il **Workspace**

- **Chiudere** il **Workspace**

accettando di chiudere tutti i documenti Windows

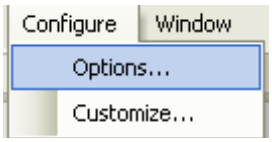


- potendo visualizzare i più **recenti Workspaces**



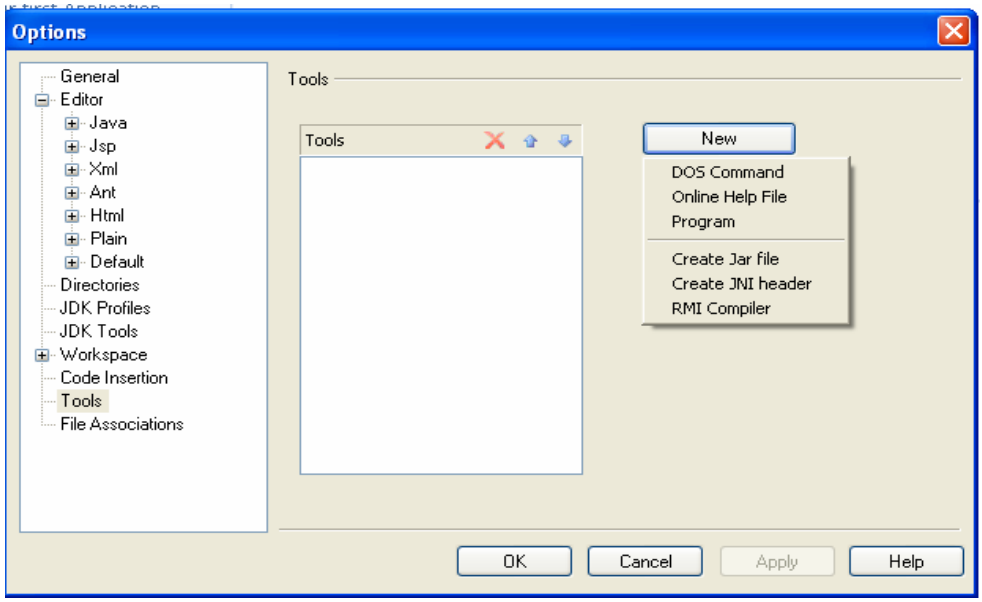
Aggiungere Tools

Con percorso : **Configure → Options**

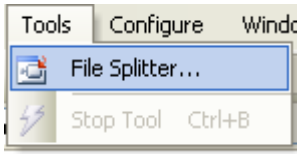


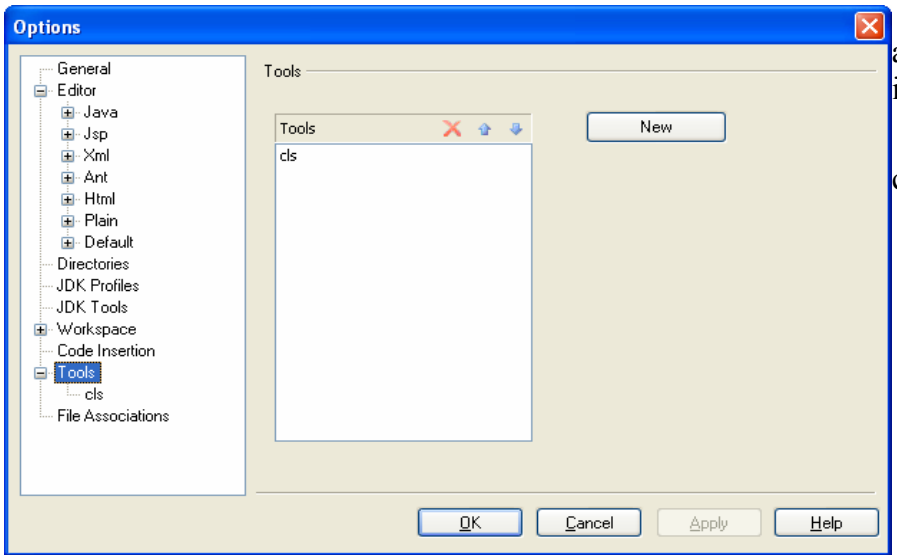
→ Workspace

Tools



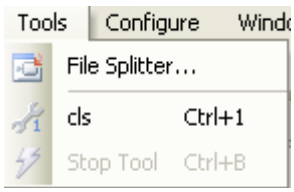
Senza nessuna aggiunta



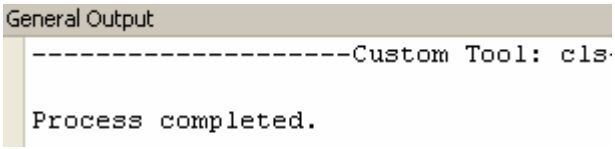


aggiungendo il comando DOS

cls

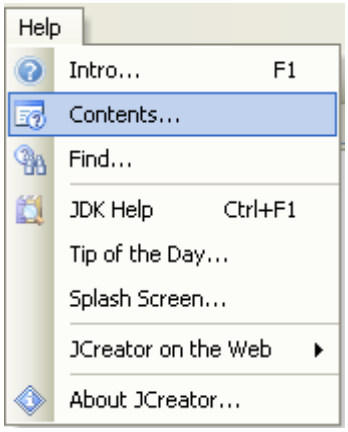


Effetto: cancella nella sezione General Output

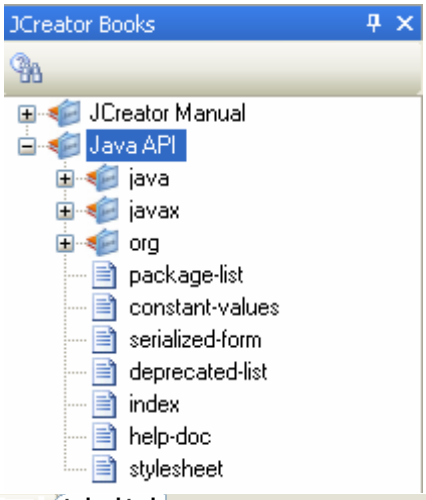


Visualizzare la **documentazione completa: JCreator Books**

con percorso **Help → Contents**

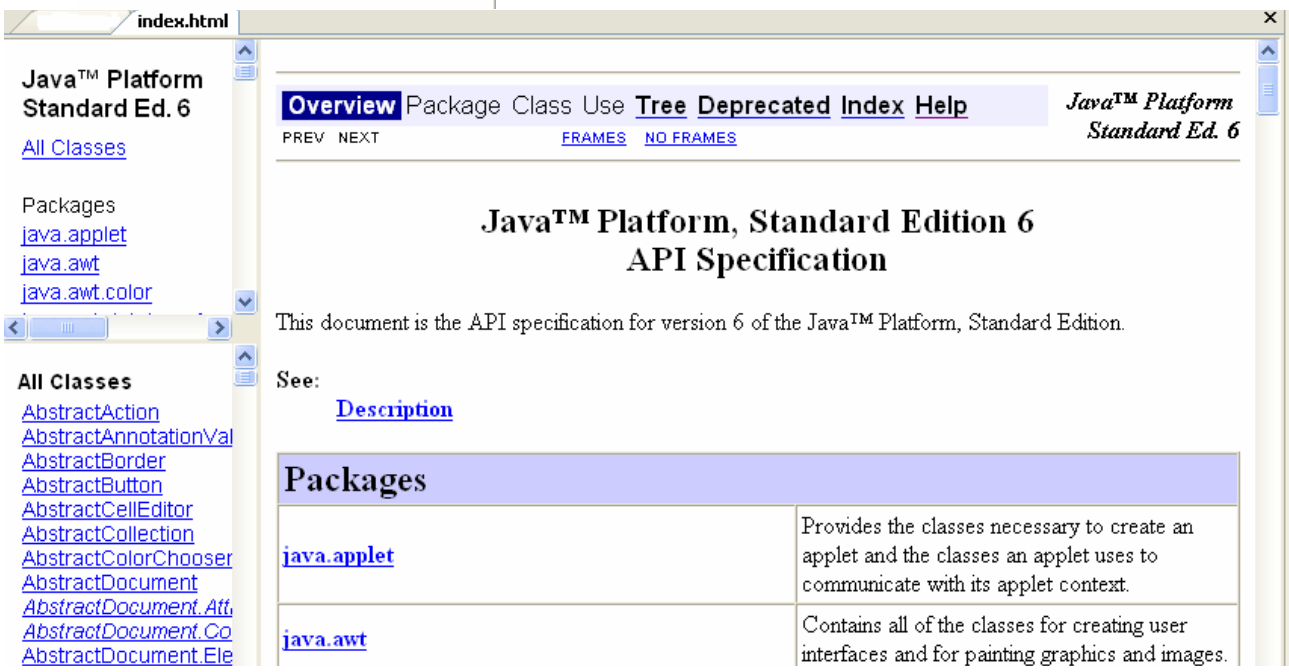


avendo configurato[4] visibile
la documentazione del JDK,
espandendone la visualizzazione



con

doppio click su **index** si aggiunge come scheda la documentazione delle Java **API**

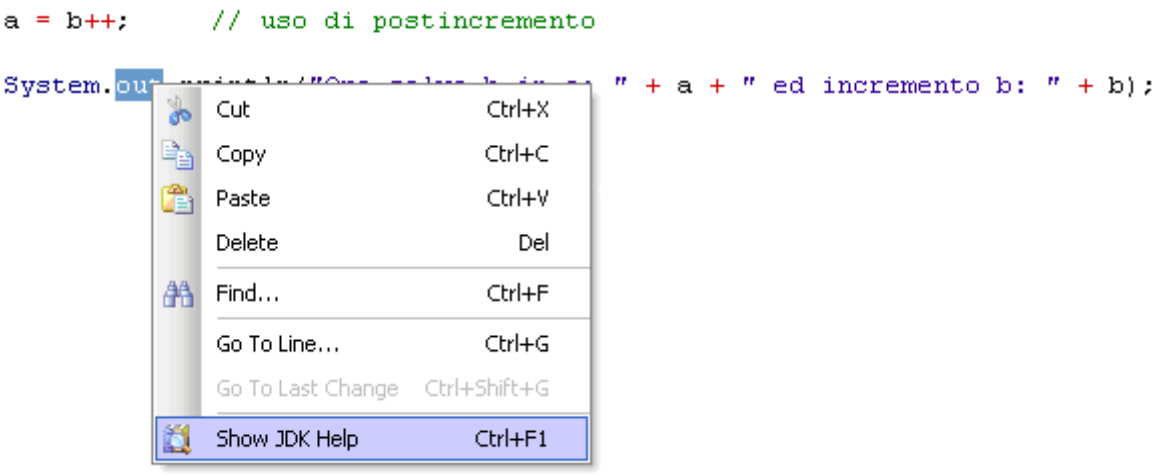


[4] Altre guide passo-passo volendo eseguire la procedura di installazione o modificare le opzioni di configurazione in ambiente JCreator.

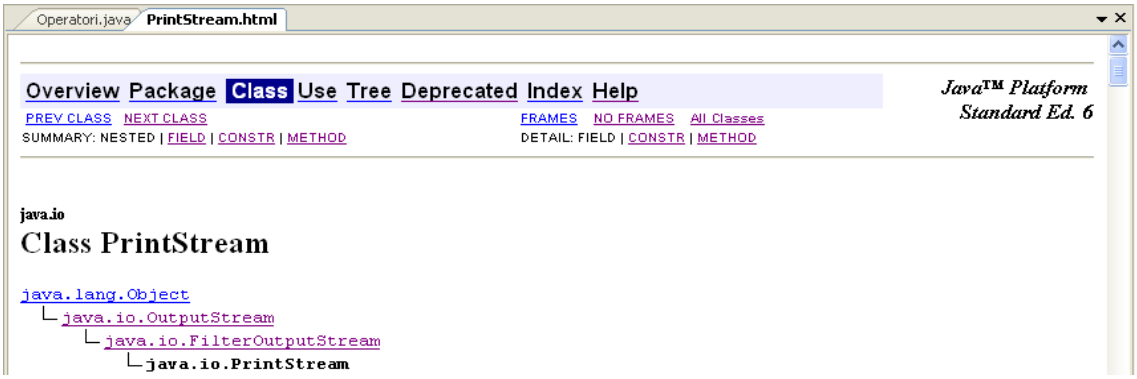
Esempio d'uso di documentazione JDK

Selezionato l'oggetto **out**, con tasto destro del mouse si apre menu contestuale e cliccando su

Show JDK Help

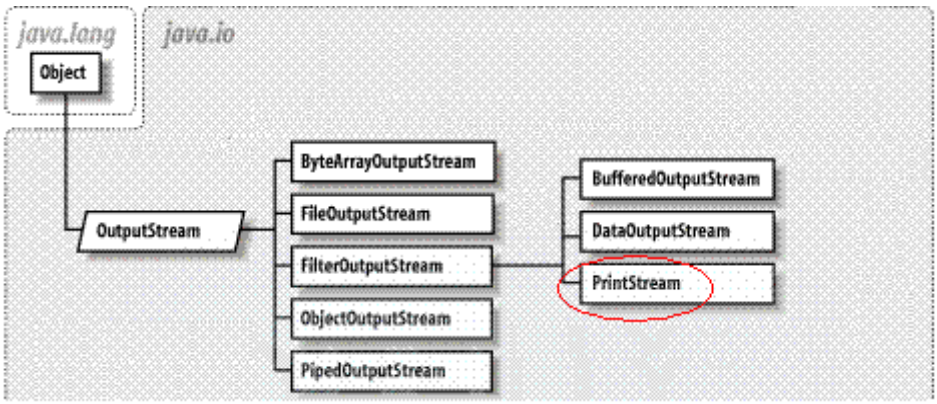


si apre come scheda la pagina che documenta la ***classe*** (**PrintStream**) di cui **out** è un oggetto:



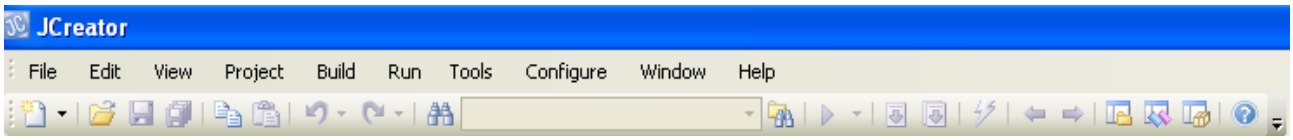
mostrando la *gerarchia* delle ***classi*** (predefinite e memorizzate all'interno di **package**) da cui ***eredita*** tutte le caratteristiche, specializzandosi sempre più:

PrintStream definisce **oggetti** di tipo **flusso** per **scrivere** (*OutputStream*) dati "**filtrati**" come **sequenze di caratteri**

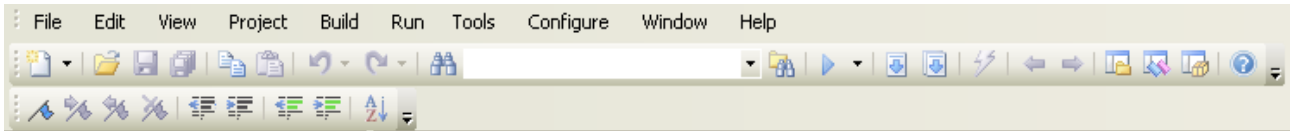


Barre e opzioni di Edit

Può essere utile visualizzare oltre la barra dei menù e la barra degli strumenti standard (di default)

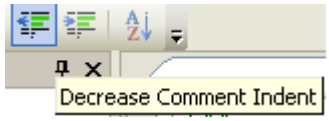


anche la barra degli strumenti per l'**Edit** che consente di :

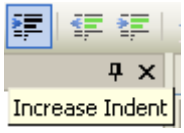


- inserire a commento righe di codice

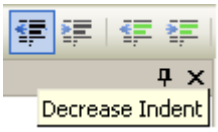
oppure eliminare il commento
ad inizio riga //



- **indentare**
aumentando il margine rientrato



o diminuendolo



Per impostare un numero di spazi per un rientro del margine diverso da quello impostato di default (un tabulatore) potendo smarcare la possibilità di inserire il numero delle linee e la cancellazione degli spazi inutili (Trim trailing spaces):

con percorso **Configure → Options**

selezionare Editor - **Java**