

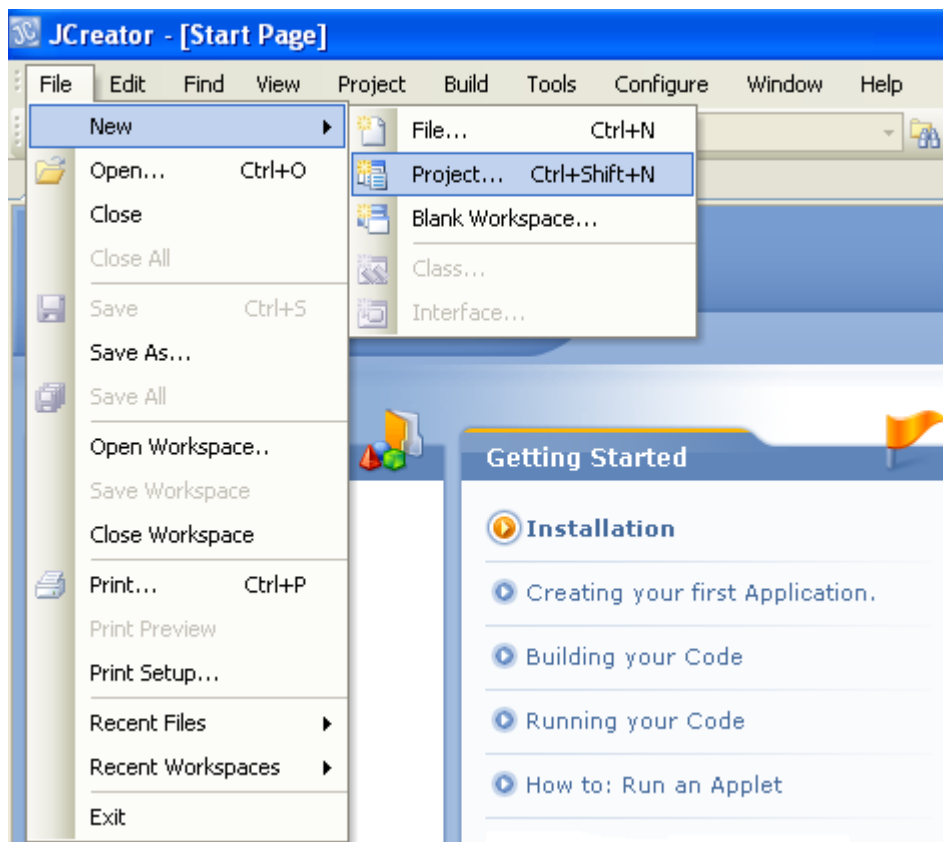
Guida all'uso dell'ambiente di sviluppo¹ integrato o IDE JCreator LE

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:

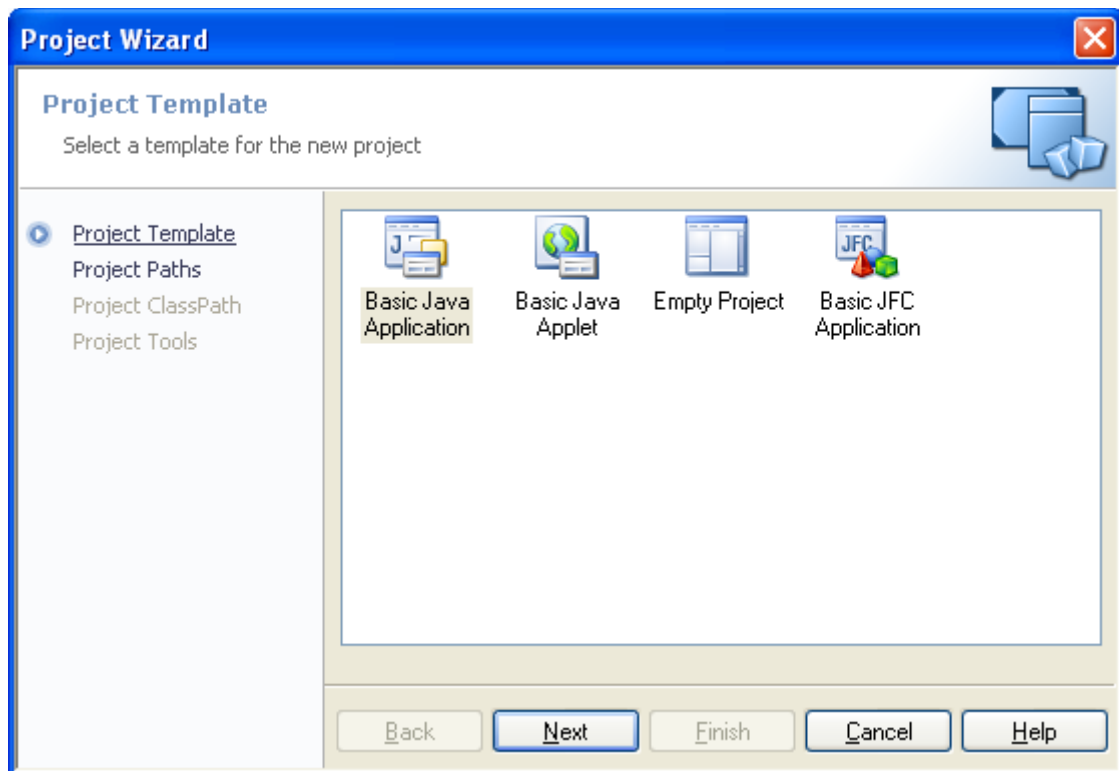


- Selezionando una prima volta i comandi **File** → **New** si scelga **Project**

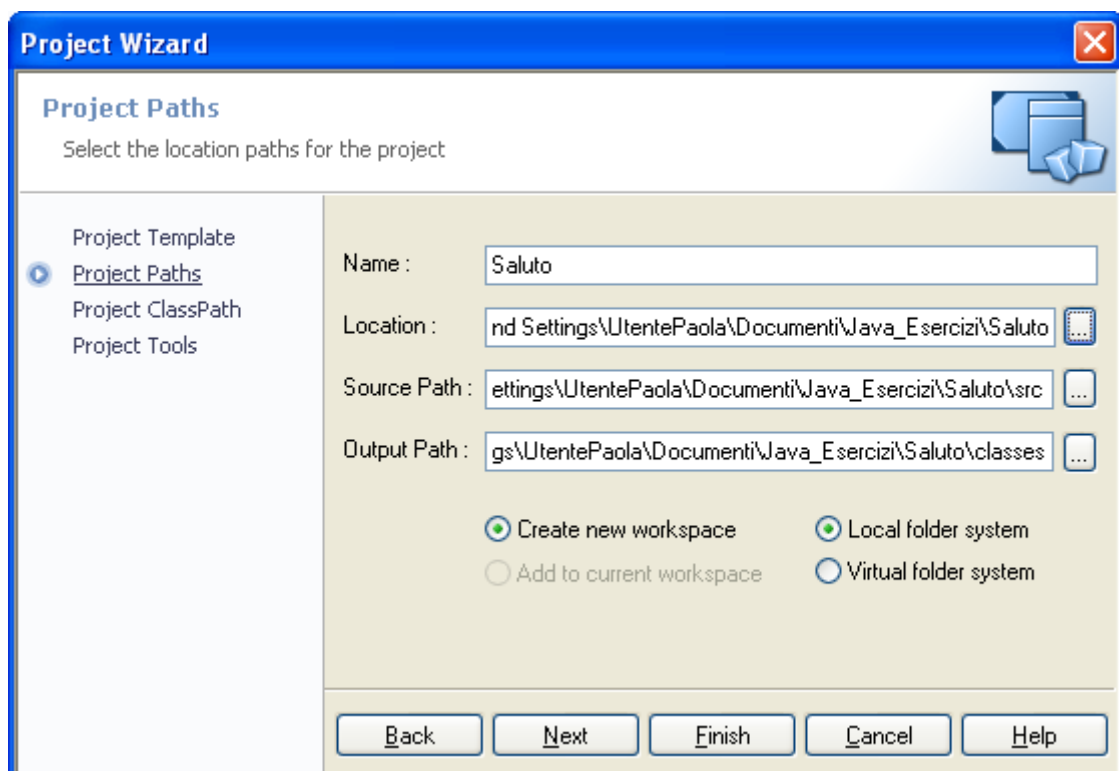


¹ Tra i livelli di funzionalità di un calcolatore si definisce **software di sviluppo** quell'insieme di programmi che consentono di sviluppare le applicazioni definite dall'utente: editor, compilatori, interpreti e linker-loader cioè programmi capaci di tradurre sequenze di istruzioni formulate in linguaggio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

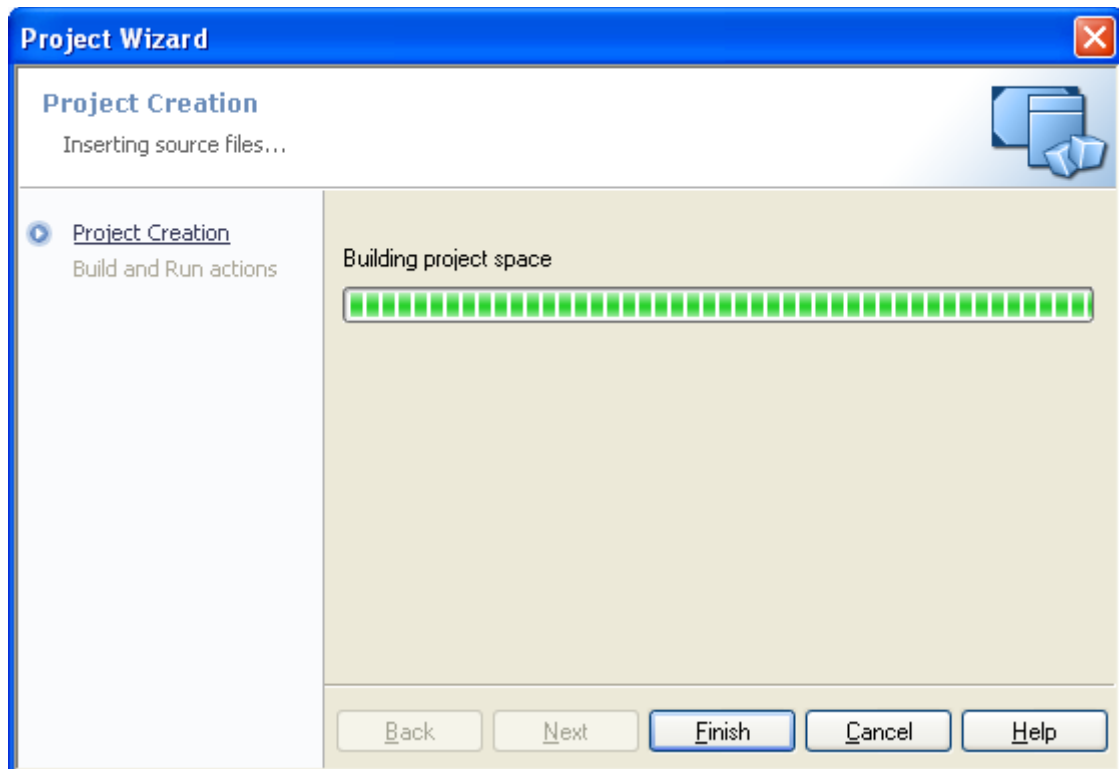
- si crei il progetto di tipo **Basic Java Application**



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



- si preme Finish



- si preme Finish nuovamente

Prima FASE:

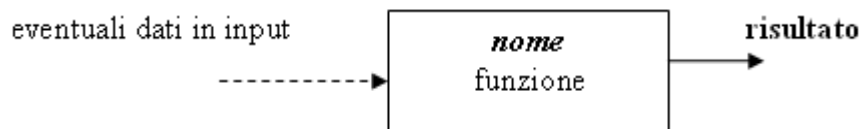
uso dell'**editor integrato** per modificare il *file sorgente* che ha lo stesso nome della classe (ogni applicazione contiene la definizione di almeno una *classe*: la descrizione generale delle caratteristiche di un *esecutore* ad esempio per visualizzare un Saluto) ed estensione *.java*

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 (*Class View*), sia l’Output per informazioni sull’esito delle varie fasi nella costruzione del programma, sia il foglio in cui scrivere.

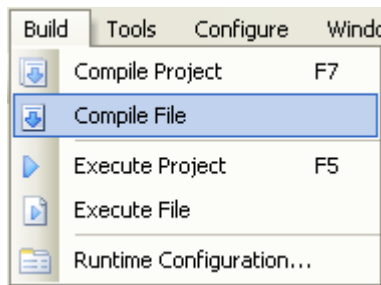
Scoperta: se si aggiungono nel progetto più classi (ciascuna con metodo *main*) [cosa succede ?](#)

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.

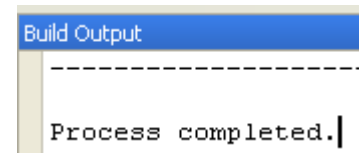
Seconda FASE:

compilazione (traduzione in **bytecode** 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** → **Compile File**



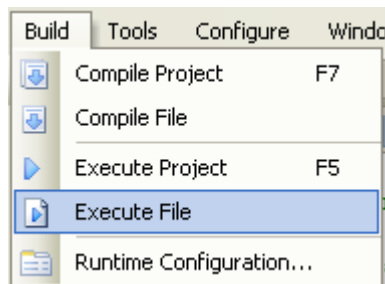
in assenza di errori, compare avviso di elaborazione completata



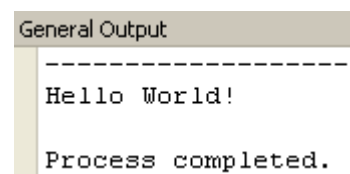
Terza FASE

Interpretazione del **bytecode** (meta-codice universale) da parte della **Java Virtual Machine** cioè generazione del codice specifico per la piattaforma su cui viene eseguito il programma caricato in RAM

- Selezionando il comando **Build** → **Execute File**



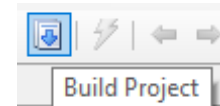
in assenza di errori, compare il risultato:
la visualizzazione della stringa *Hello World!*
su video nella sezione General Output



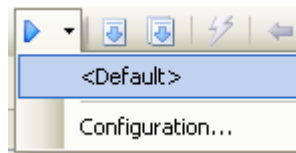
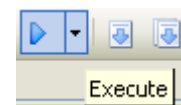
Si possono, in alternativa, usare le opportune³ icone:



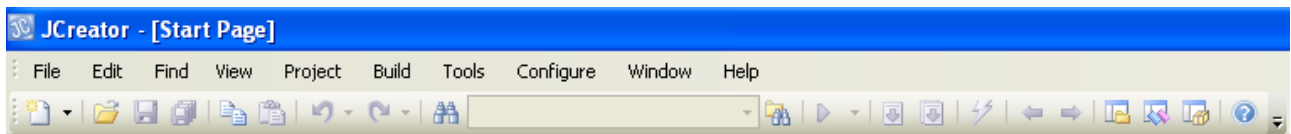
- per realizzare la fase di COMPILAZIONE del File
- per realizzare la fase di COMPILAZIONE del Progetto



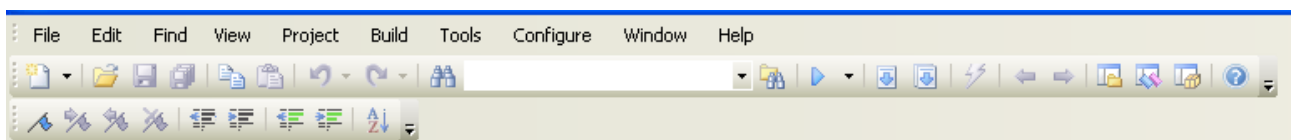
- per realizzare la fase di **interpretazione** al *run time* del File:
- in configurazione di default
- impostando una diversa configurazione passando argomenti al *run time*



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



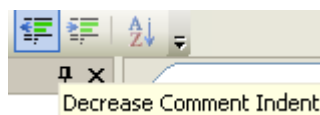
anche la barra degli strumenti per l'**Edit** che consente di :



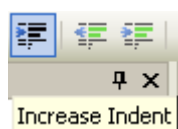
- inserire a commento righe di codice



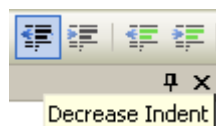
oppure eliminare il commento ad inizio riga //



- **indentare** aumentando il margine rientrato



o diminuendolo

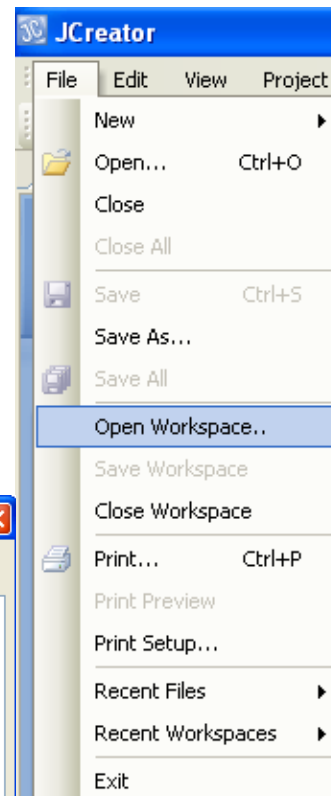
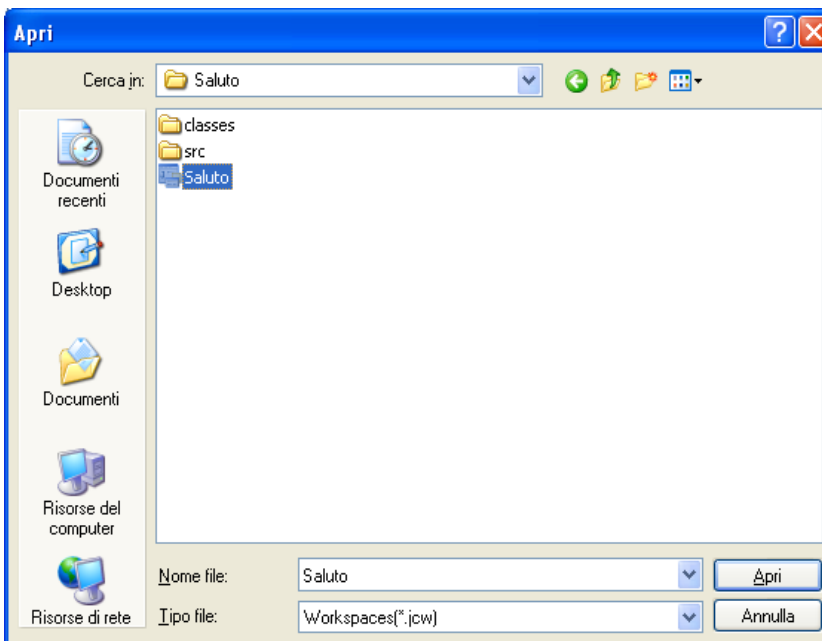


³ Nella versione Professional è prevista opzione di debugging **Build** → **Start Debug**

Sviluppare programmi nell'ambiente di lavoro

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

selezionando l'opportuno **Workspace**
(equivale ad aprire il file.jcw)



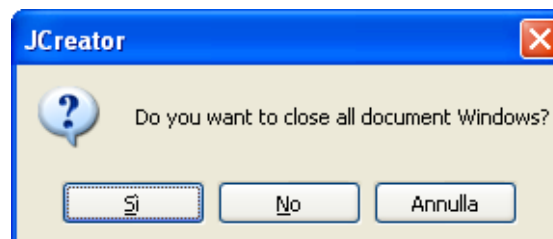
- **Salvare il Workspace**

Save Workspace

- **Chiudere il Workspace**

Close Workspace

accettando di chiudere tutti i documenti
Windows

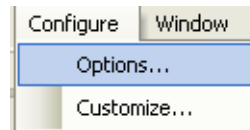


- potendo visualizzare i più **recenti Workspaces**

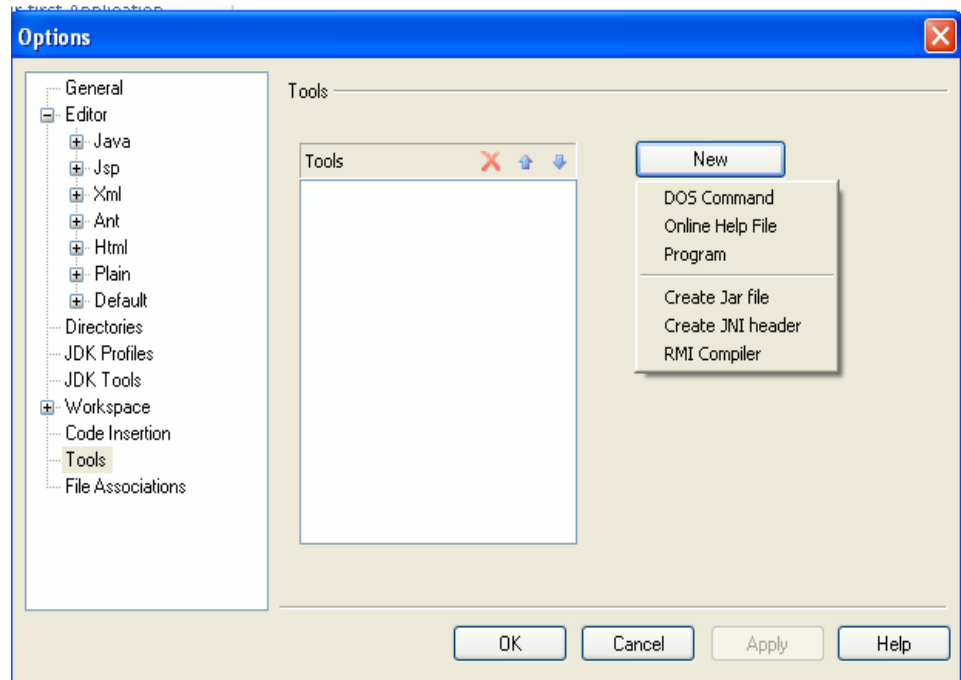
Recent Workspaces ▶

Aggiungere Tools

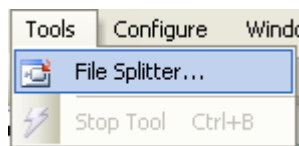
Con percorso : **Configure** → **Options**



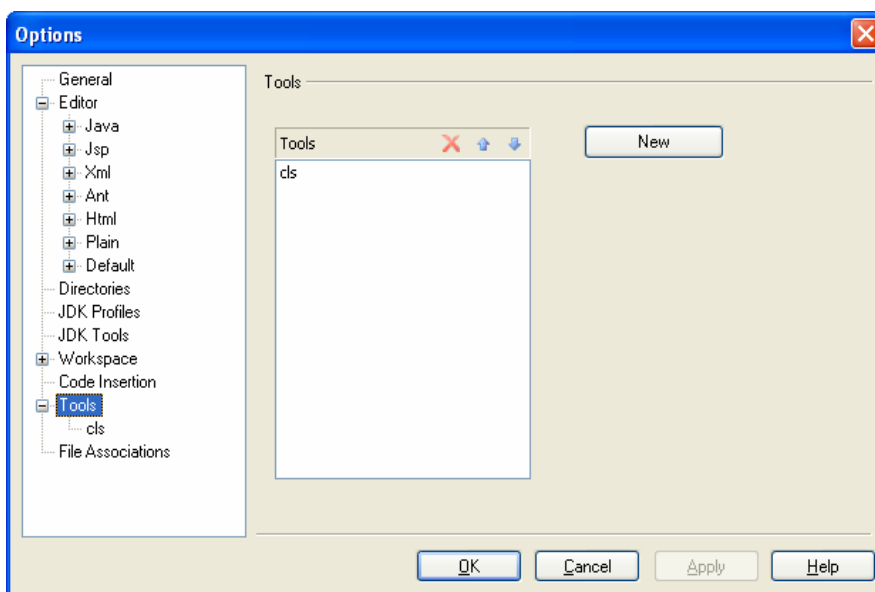
→ **Workspace Tools**



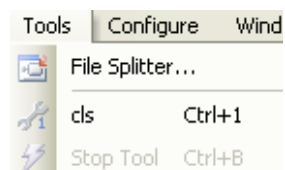
Senza nessuna aggiunta



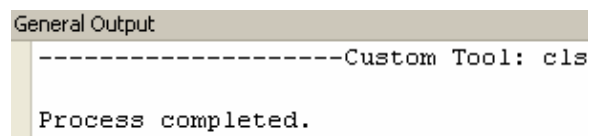
aggiungendo
il comando DOS



cls

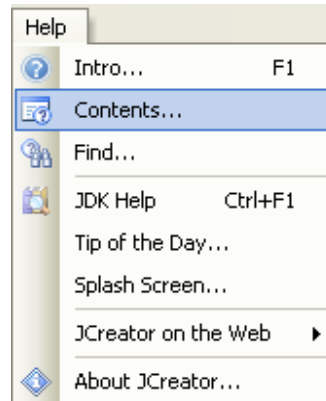


Effetto: cancella nella sezione
General Output

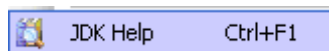


Visualizzare la documentazione completa: JCreator Books

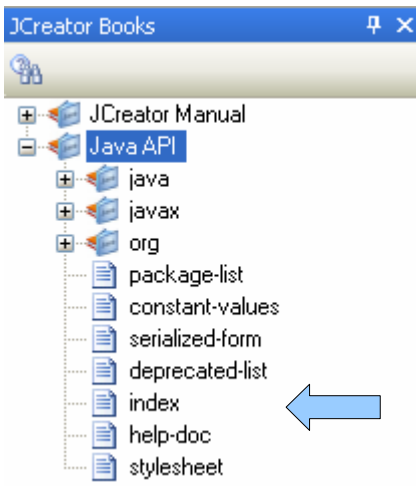
con percorso **Help** → **Contents**



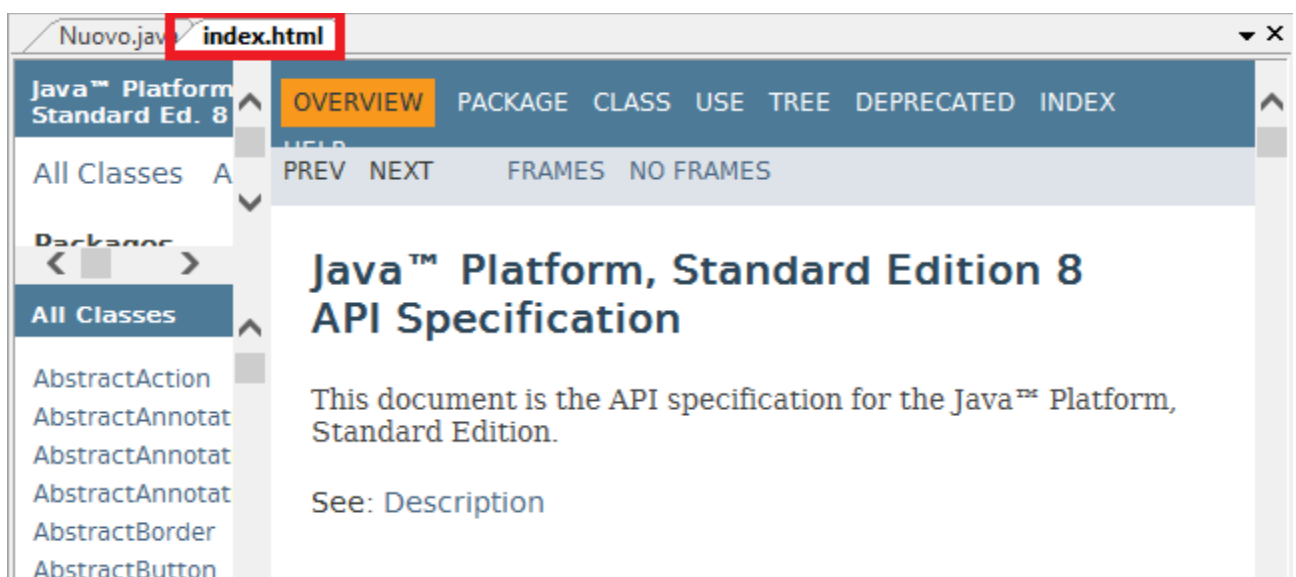
avendo [configurato](#)⁴ visibile la documentazione del JDK,



espandendone la visualizzazione



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



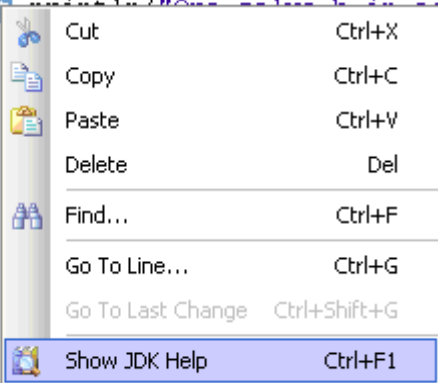
⁴ Altre guide passo-passo volendo eseguire la procedura di [installazione](#) o [modificare](#) le opzioni di configurazione in ambiente JCreator.

Esempio d'uso di documentazione JDK

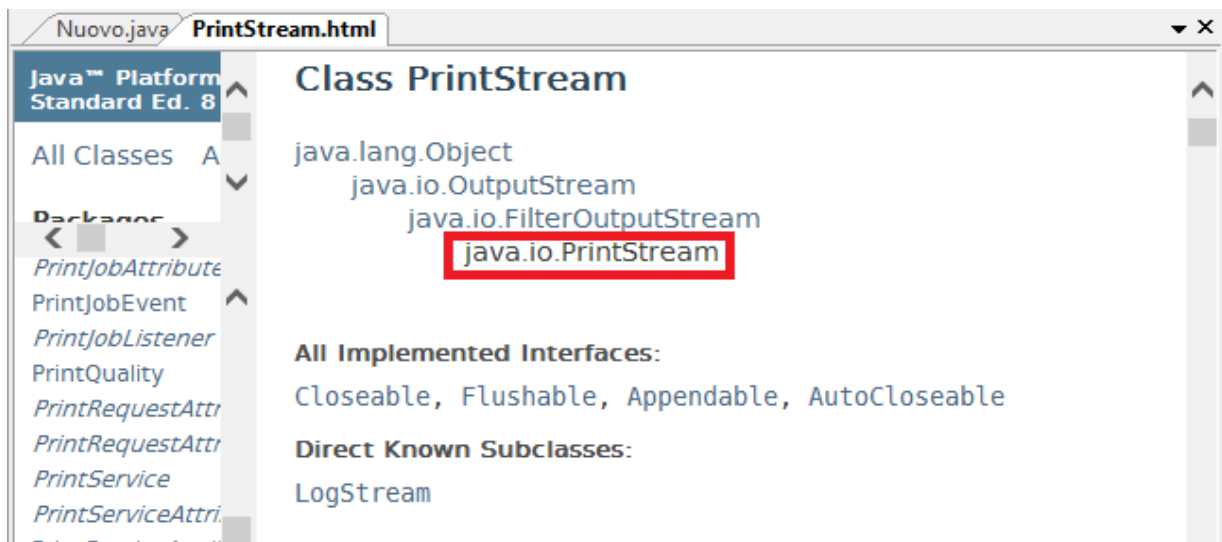
Selezionato l'oggetto **out**, con tasto destro del mouse si apre menu contestuale e cliccando su

Show JDK Help

```
a = b++; // uso di postincremento
System.out.println("Output: " + a + " ed incremento b: " + b);
```

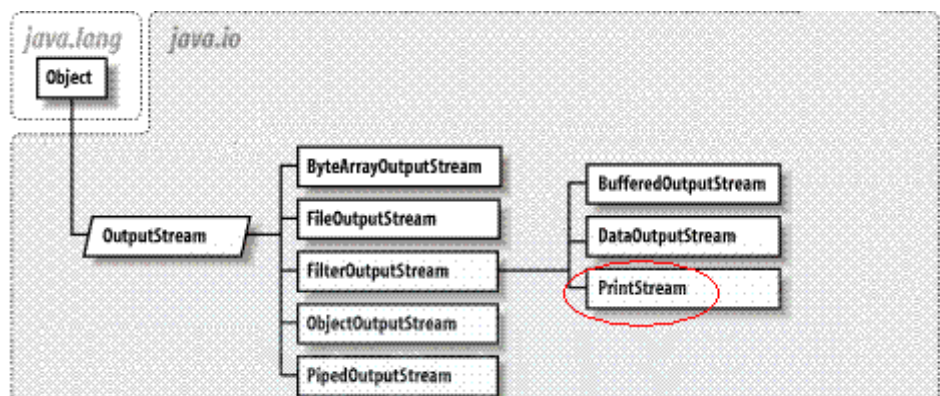


si apre la documentazione del JDK (se non già visualizzata) potendo inserire come *scheda* la pagina che documenta la **classe** (**PrintStream**) di cui **out** è un oggetto:



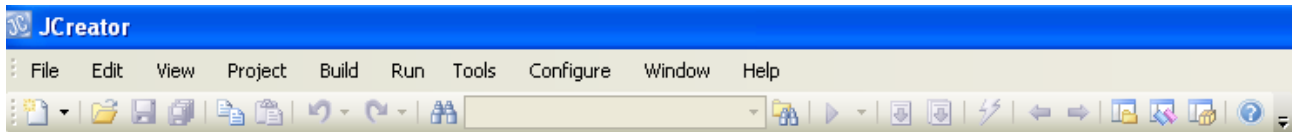
mostrando la *gerarchia* delle **classi** (predefinite e memorizzate all'interno di **package**) da cui *eredita* tutte le caratteristiche, specializzandosi sempre più:

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

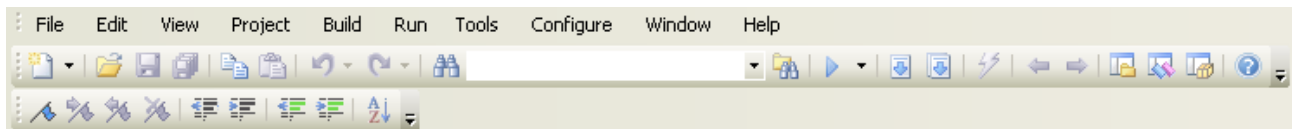


Barre e opzioni di Edit

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



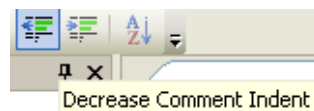
anche la barra degli strumenti per l'Edit che consente di:



· inserire a commento righe di codice

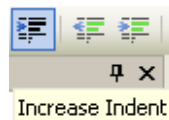


oppure eliminare il commento ad inizio riga //

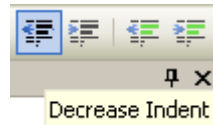


· **indentare**

aumentando il margine rientrato

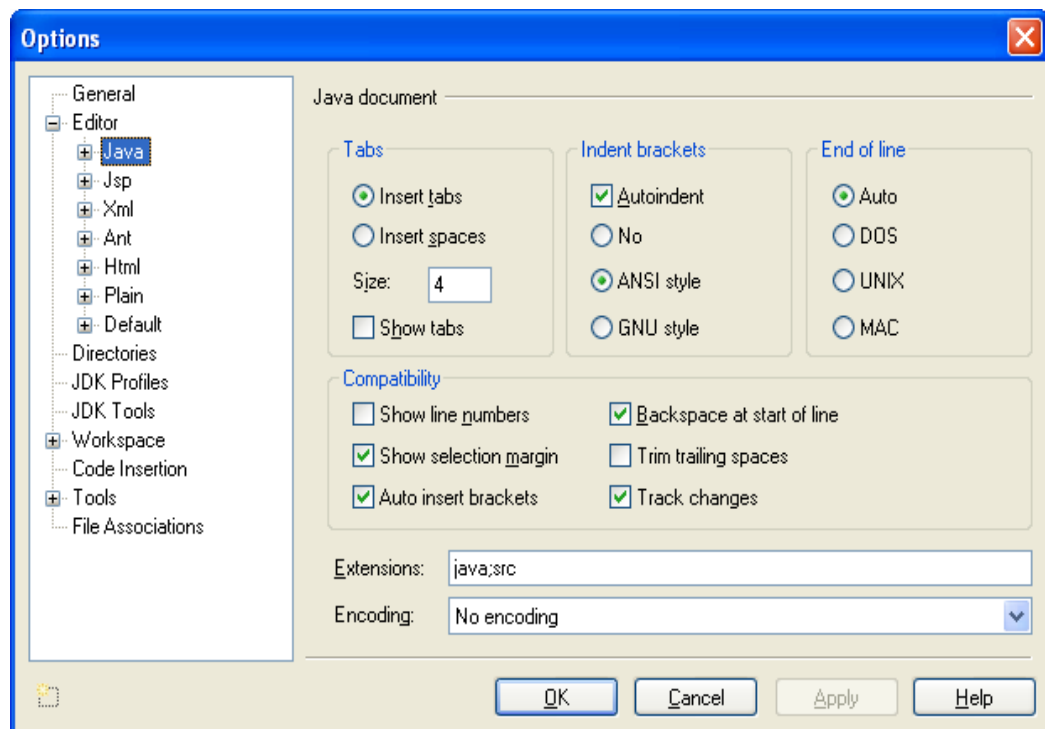


o diminuendolo



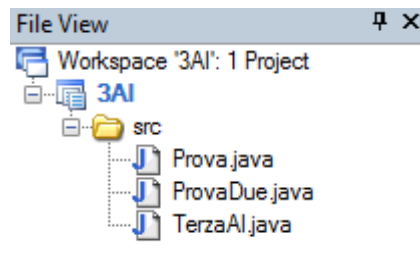
Per impostare un numero di spazi per un rientro del margine diverso da quello impostato di default (un tabulatore) potendo smarcare la possibilità di inserire il numero delle linee e la cancellazione degli spazi inutili (Trim trailing spaces): con percorso **Configure** → **Options**

selezionare Editor – **Java**



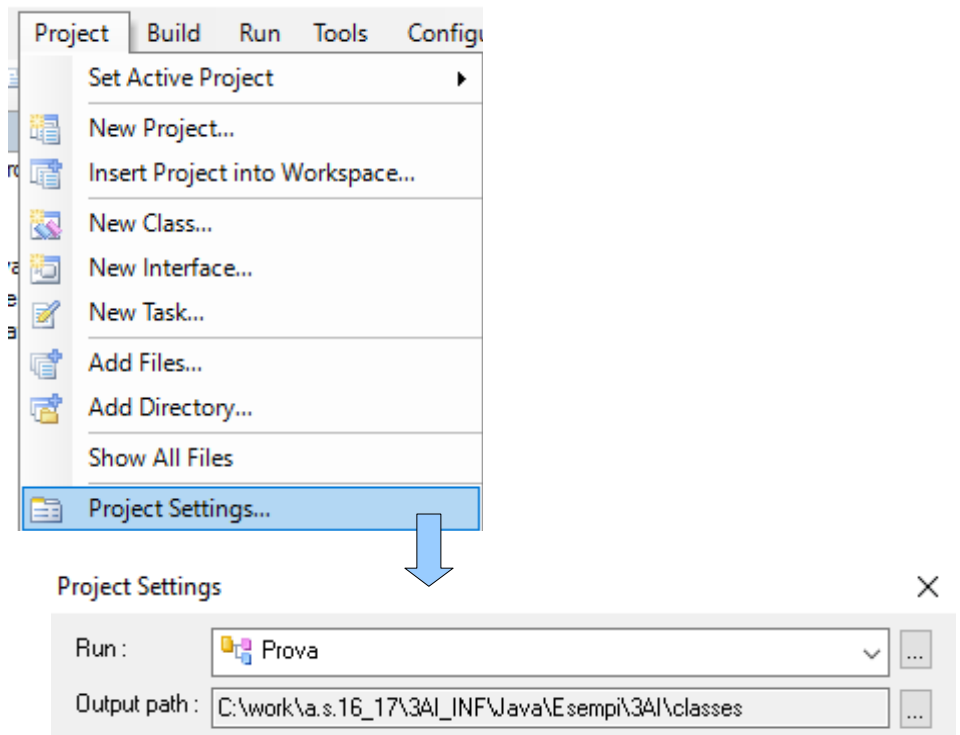
L'ambiente “risolve problemi per noi”

Se si aggiungono nel progetto più classi (ciascuna con metodo main)

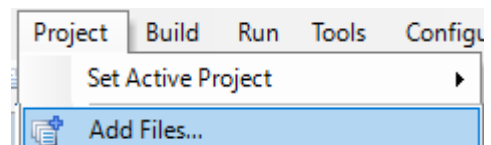


in realtà, in automatico, al momento della costruzione del progetto, è stata settata la **prima classe definita** come quella da eseguire

volendo si può cambiare:



Come si possono aggiungere file sorgente al progetto?



Come creare nel progetto nuovi file ?

