

Integrated Development Environment (IDE)

- Gli IDE sono **ambienti / sistemi** che includono:
 - Editor di testo con supporto alla scrittura del codice
 - Compilatore
 - Debugger
 - Testing
 - Deployment
 - Gestione di Progetti
- Per **progetto** si intende una collezione di risorse, sorgenti, librerie e documentazione

1

Eclipse

Eclipse è realizzato da una **community** focalizzata sullo sviluppo di strumenti per sviluppatori

<http://www.eclipse.org/>

- È **opensource** e scritto in Java
- È **multiplatforma** (disponibile per Windows, Mac Os, Linux)
- Sono disponibili numerosi **plugin** opensource e commerciali
- **Supporto per più linguaggi** (Java, C, C++, XML, HTML, PHP, ecc.)



2

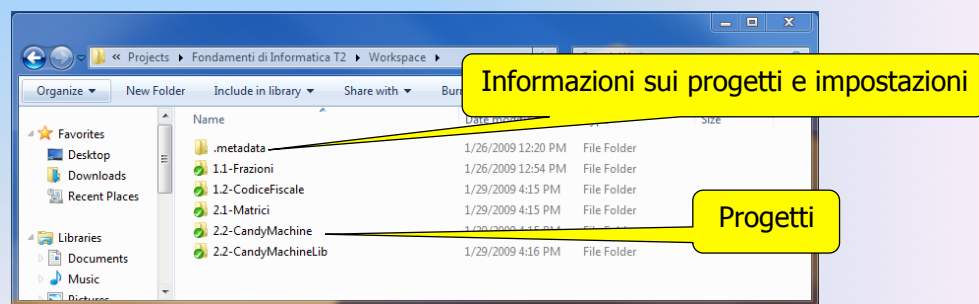
Installazione

- **Scaricare** Eclipse dal sito www.eclipse.org per la piattaforma utilizzata (file compresso)
(*N.B. Se la JDK installata su Windows è a 64 bit anche Eclipse deve essere a 64 bit, altrimenti non funziona!!!*)
- **Decomprimere** il contenuto in una cartella e l'installazione è completata. Eclipse è contenuto interamente all'interno della cartella decompressa. (Es. *C:\Programmi\Eclipse*)

3

Workspace

- È l'**ambiente di lavoro** in cui vengono creati e mantenuti i progetti.
- Ogni **progetto** è all'interno di una directory con lo stesso nome del progetto
- All'interno del workspace sono salvate anche le **impostazioni** di Eclipse

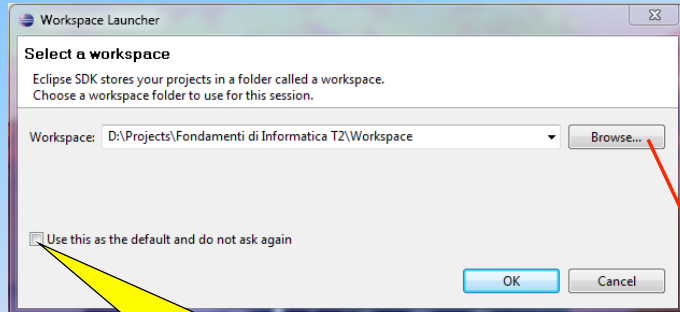


ATTENZIONE!! Cambiare workspace significa anche cambiare le impostazioni di Eclipse

4

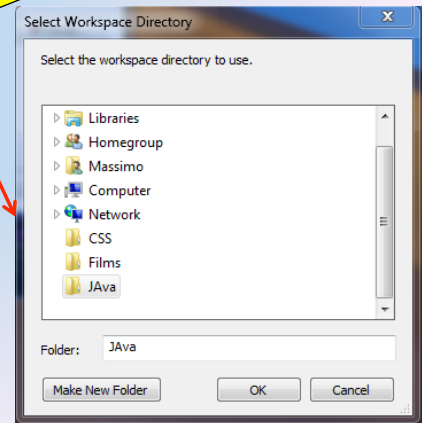
Primo Avvio

- Al primo avvio verrà visualizzata questa schermata



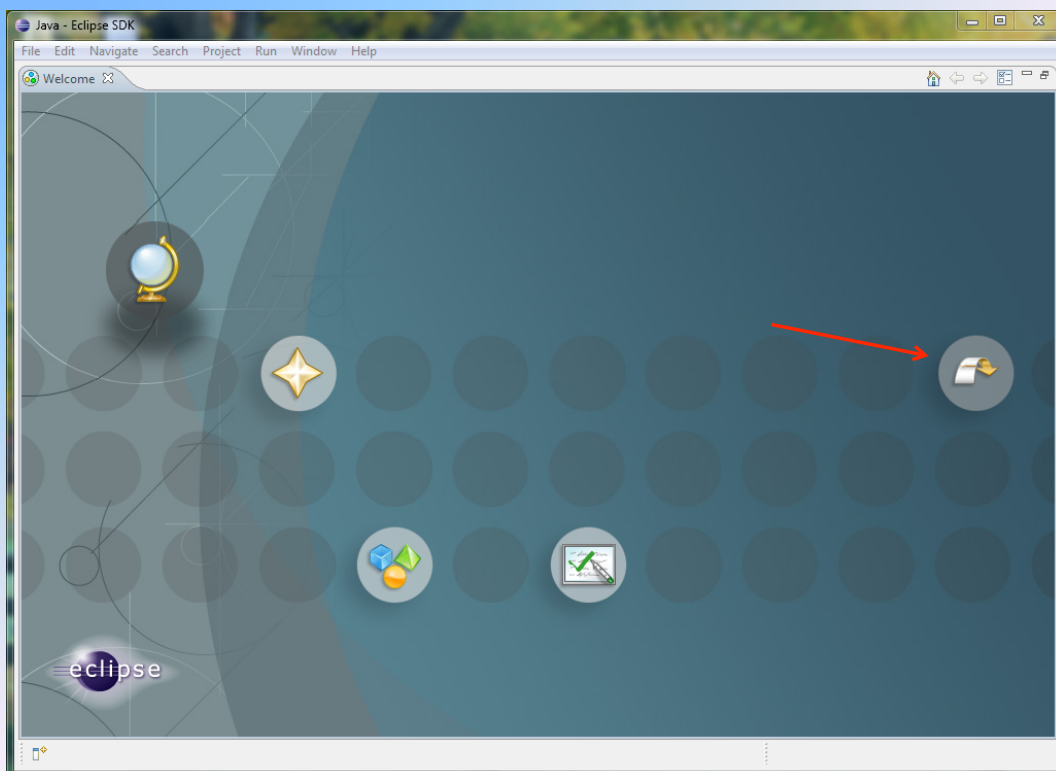
Impostare la cartella su cui salvare il workspace.

Verrà memorizzata la directory del workspace scelta e impostata come default. Al prossimo avvio non verrà più presentata questa schermata.



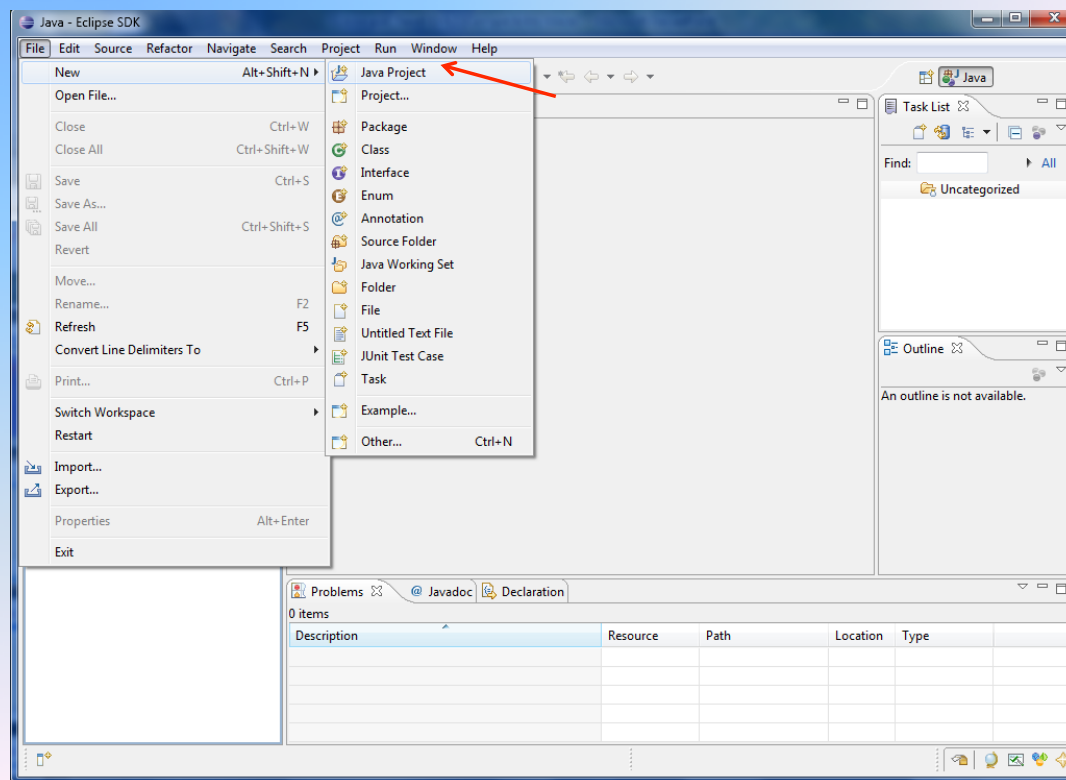
5

Primo Avvio



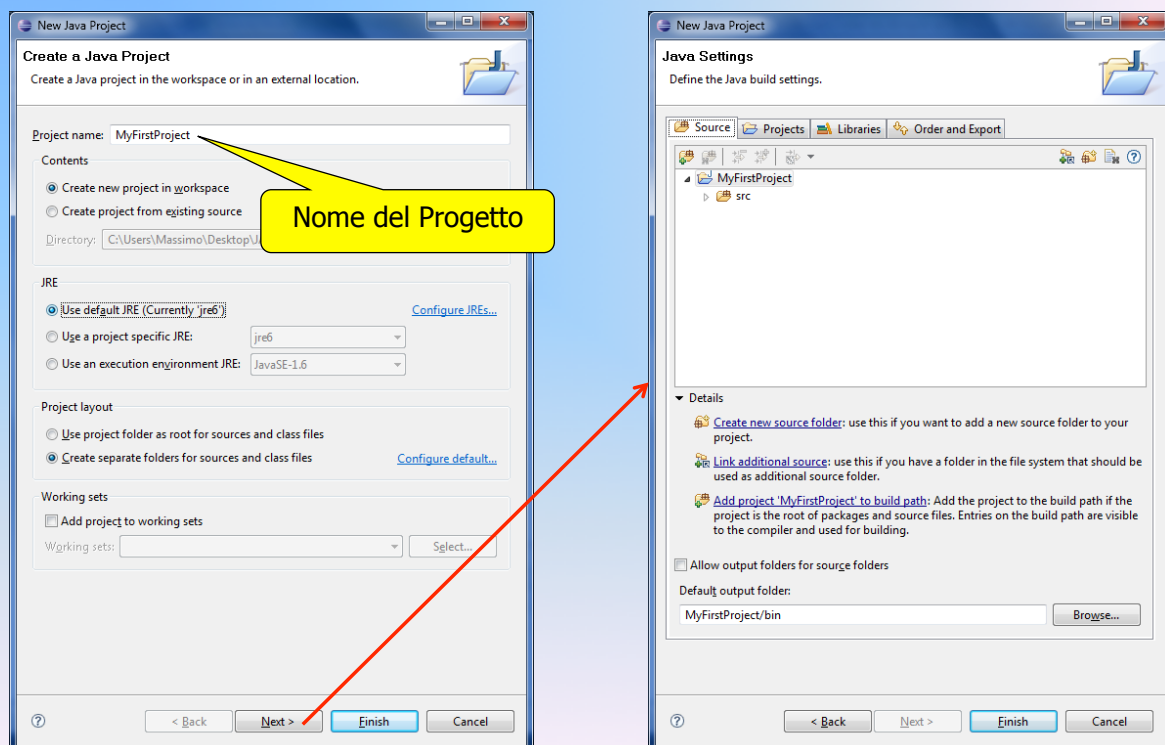
6

Creazione di un Progetto



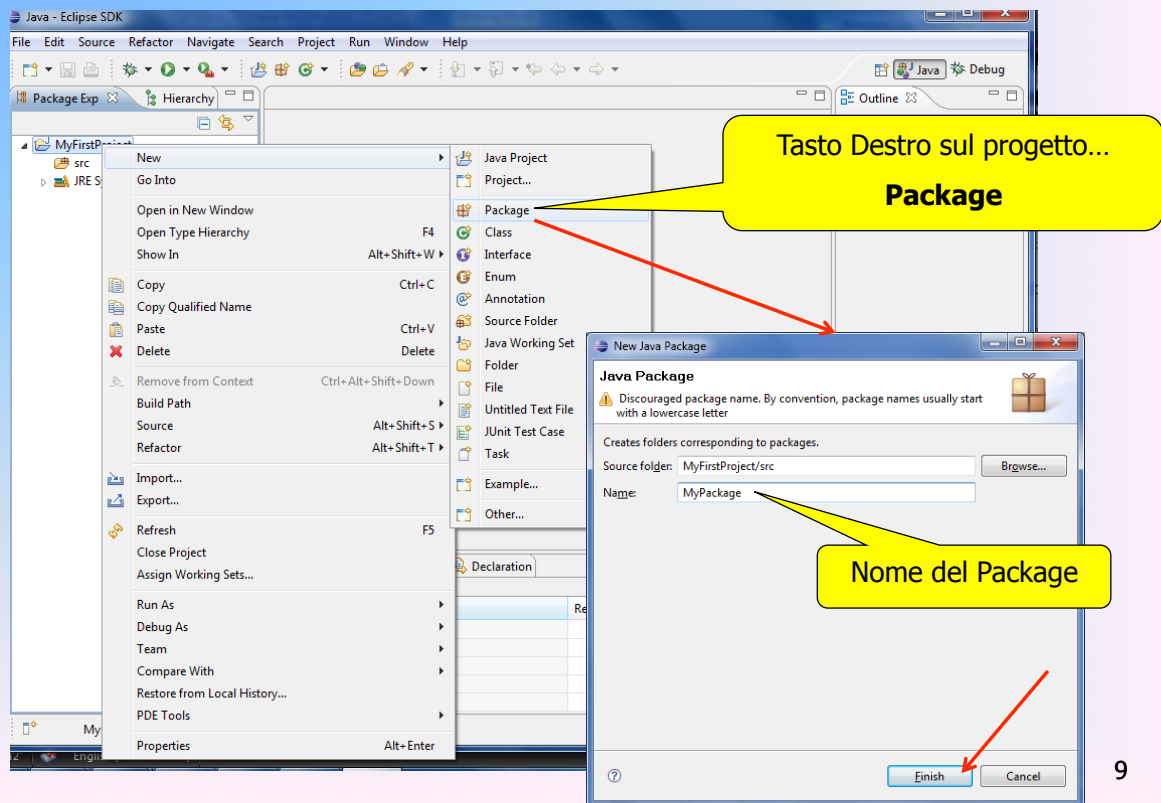
7

Creazione di un Progetto

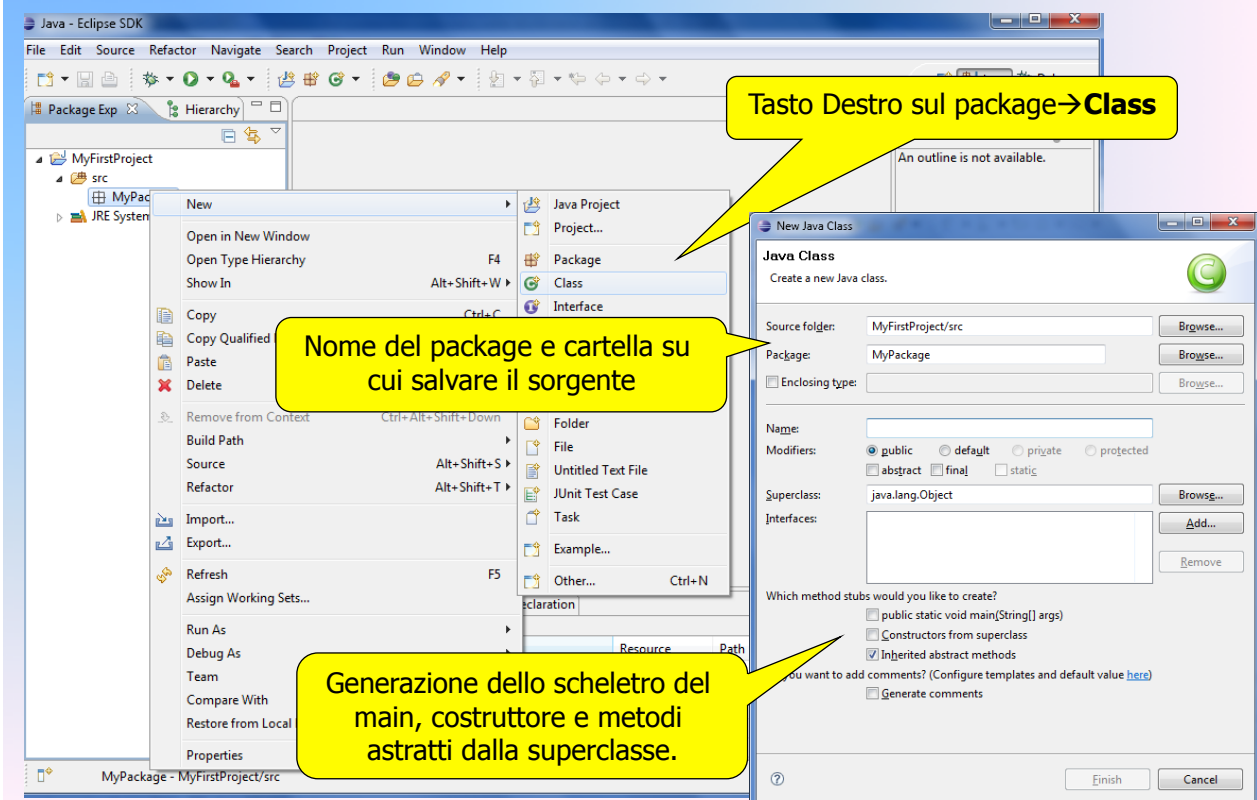


8

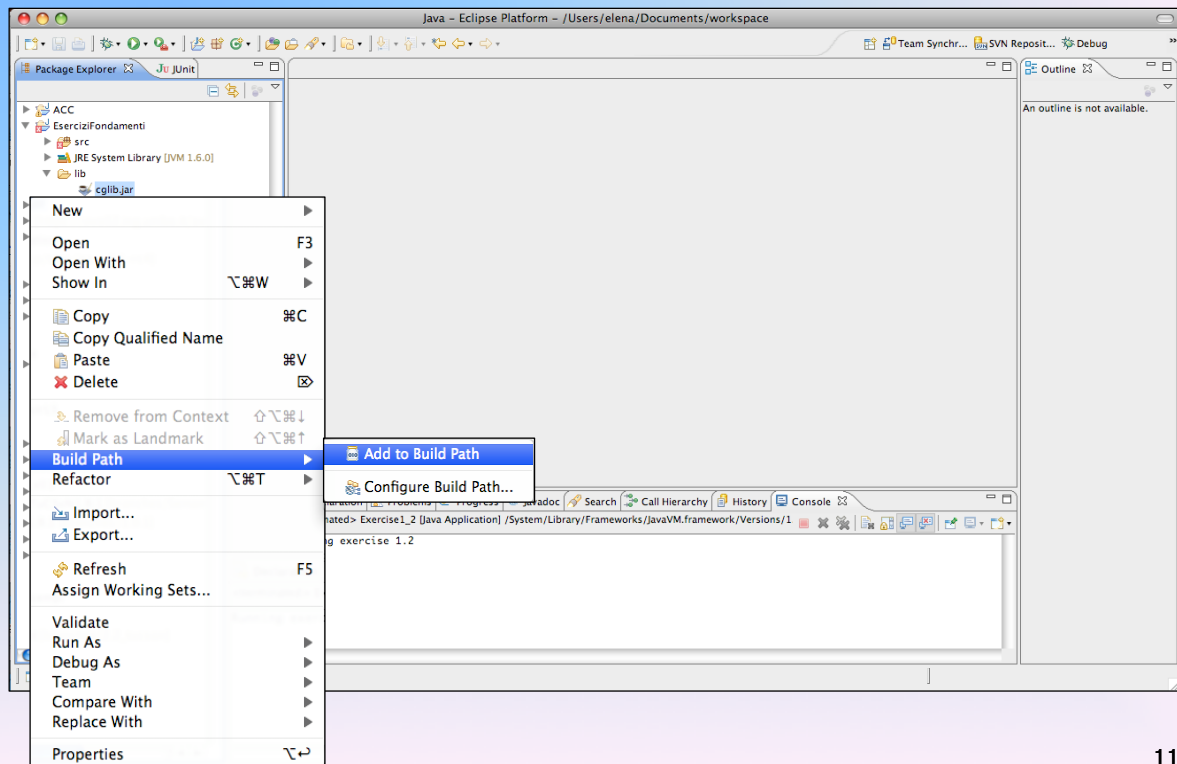
Creazione di un Package



Creazione di una Classe

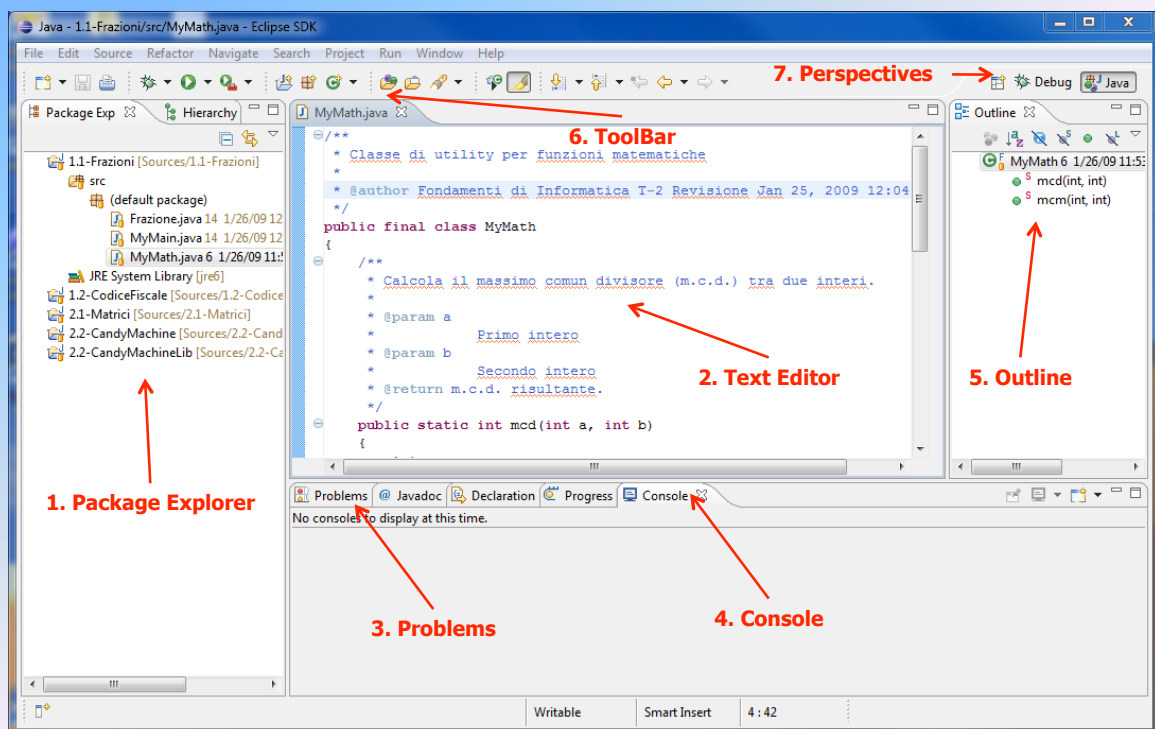


Aggiunta di una Libreria



11

Ambiente di Lavoro (1/7)

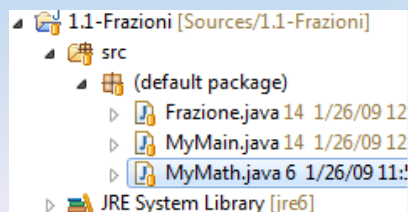


12

Ambiente di Lavoro (2/7)

1. Package Explorer

- Permette di **visualizzare i progetti** all'interno del workspace
- Raggruppa per ogni progetto in **package**
- Visualizza le **dipendenze** ad altre **librerie** (Es. JSR System Library)
- I **sorgenti** di un progetto sono inseriti nella cartella **src**



13

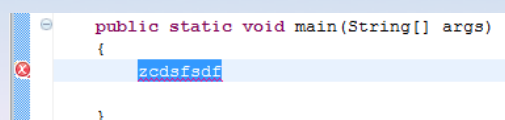
Ambiente di Lavoro (3/7)

2. Text Editor

- Permette di scrivere il codice sorgente
- Fornisce un supporto grafico evidenziano con colori differenti le parole chiave (Highlight)

```
/**
 * @param args
 */
public static void main(String[] args)
```

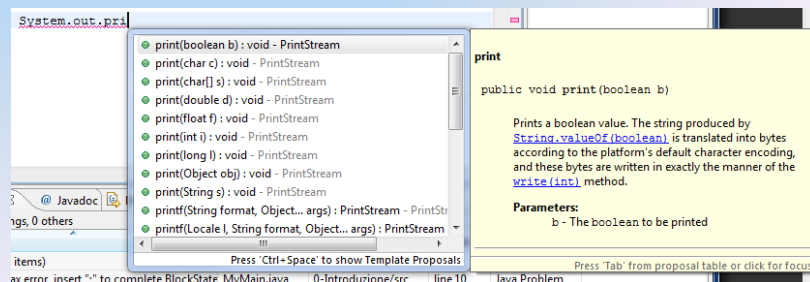
- Eclipse esegue una **compilazione** continua **in background** evidenziando eventuali errori (sottolineatura rossa ondulata)



14

Ambiente di Lavoro (4/7)

- In caso di errori/warning cliccando sull'icona rossa/gialla sul lato sinistro del codice in errore/warning viene visualizzato un elenco di **possibili soluzioni** all'errore
(**ATTENZIONE!!!** Eclipse non è un oracolo!!! Fidatevi il giusto... spesso non dice la verità...)
- Fornisce un supporto all'uso dei metodi di una classe con riquadro di Help (javadoc) del metodo selezionato.



N.B. Per forzare l'auto completamento → tasti CTRL+SPACE

15

Ambiente di Lavoro (5/7)

3. Problems

- Visualizza eventuali **warning** ed **errori** di compilazione

The screenshot shows the Eclipse Problems view. It displays three errors: 'Syntax error, insert ";" to complete BlockState', 'Syntax error, insert "AssignmentOperator Exp"', and 'zcdfsdf cannot be resolved'. Each error is associated with the file 'MyMain.java' at line 10.

Description	Resource	Path	Location	Type
Errors (3 items)				
Syntax error, insert ";" to complete BlockState	MyMain.java	0-Introduzione/src	line 10	Java Problem
Syntax error, insert "AssignmentOperator Exp"	MyMain.java	0-Introduzione/src	line 10	Java Problem
zcdfsdf cannot be resolved	MyMain.java	0-Introduzione/src	line 10	Java Problem

5. Console

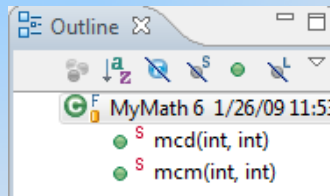
- Incapsula la console dell'applicazione includendo standard output, input ed error.

16

Ambiente di Lavoro (6/7)

5. Outline

- Visualizza la struttura della classe attualmente visualizzata nel Text Editor attraverso una struttura ad albero



6. ToolBar

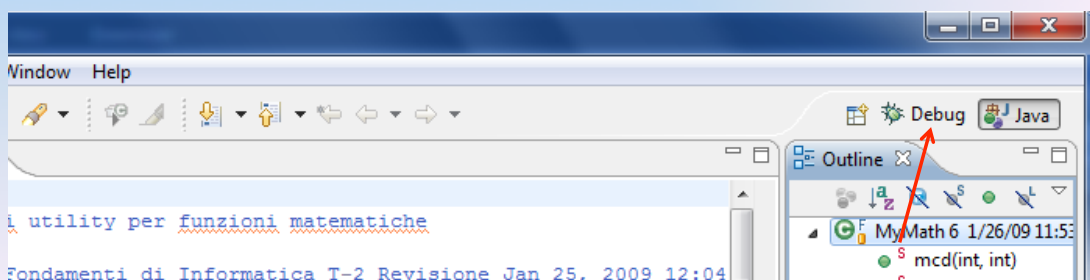
- È la classica barra con le operazioni di uso più comune (Apri, Salva, Compila, Esegui, Debug...) personalizzabile

17

Ambiente di Lavoro (7/7)

7. Perspectives

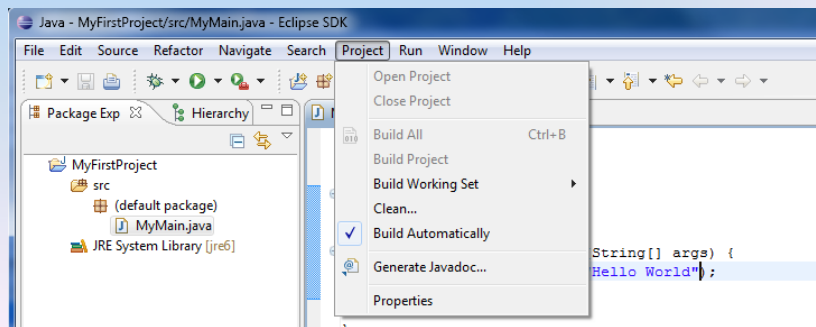
- Eclipse prevede più “prospettive” di **visualizzazione**
- Ogni prospettiva ha una serie di **viste** ed **editor**
- Esiste la prospettiva di programmazione java, di debug java, altre dipendenti dai plugin installati (*la prospettiva chiamata “Java” è quella mostrata nelle slide precedenti*)
- Per passare da una prospettiva all'altra si utilizzano i pulsanti



18

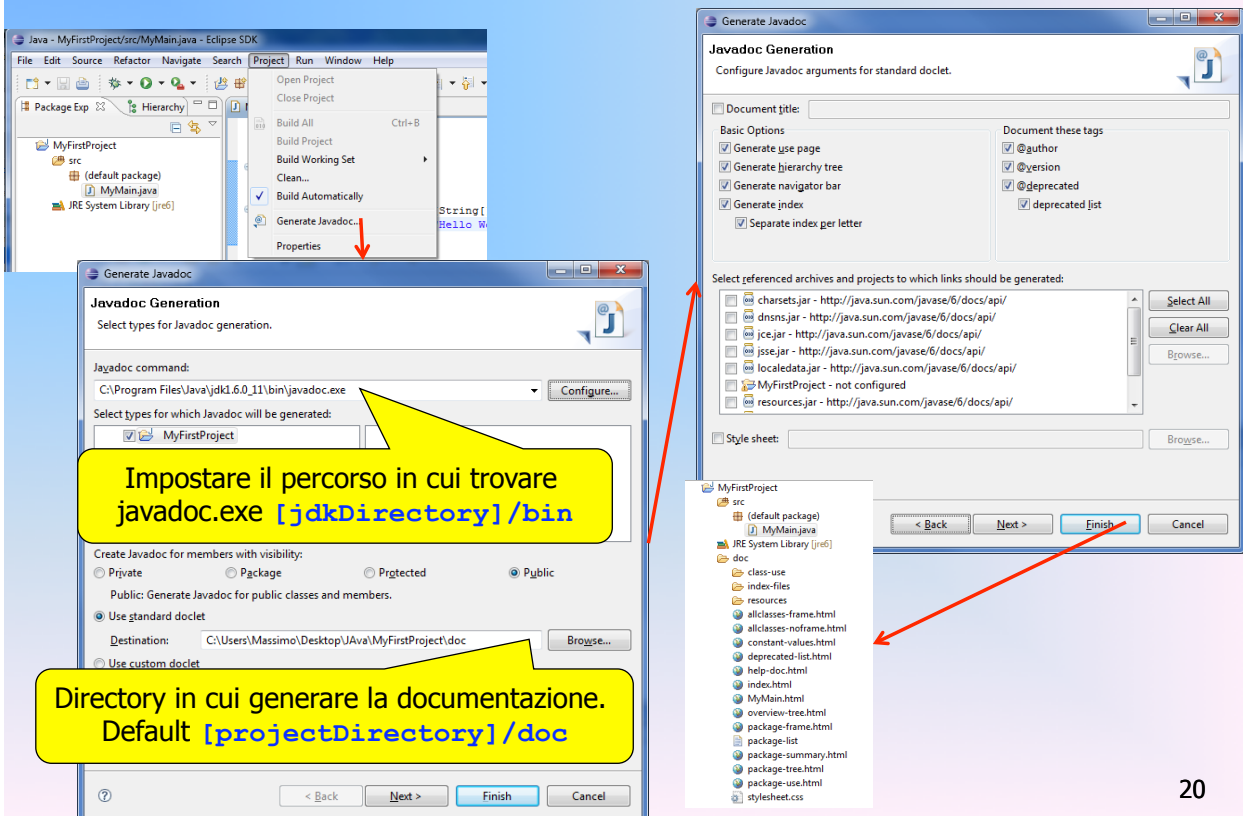
Compilazione

- Eclipse **compila** in background ad **ogni salvataggio** di un file sorgente se abilitata l'impostazione **Build Automatically** (Impostazione di default)
- Il comando **Clean** permette di eliminare i file .class generati per una ricompilazione completa
- Nel caso si desideri compilare solo “al bisogno” è possibile disattivare Build Automatically e compilare tramite comando **Build All**



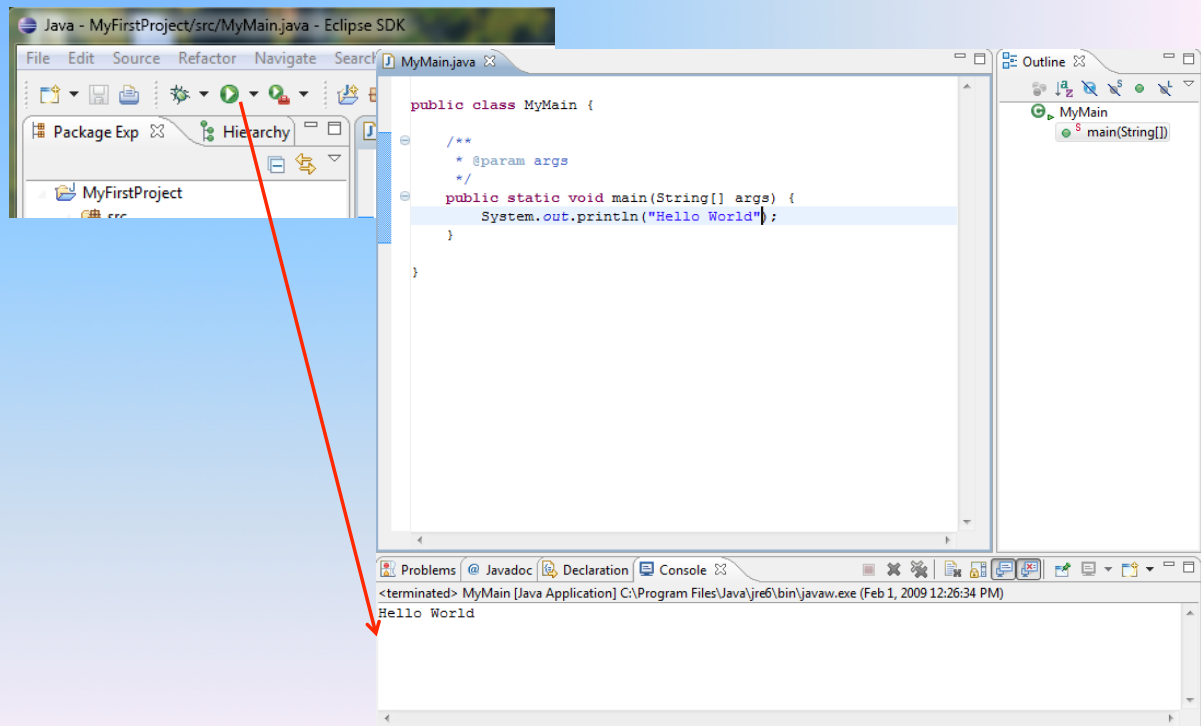
19

Generazione della Documentazione



20

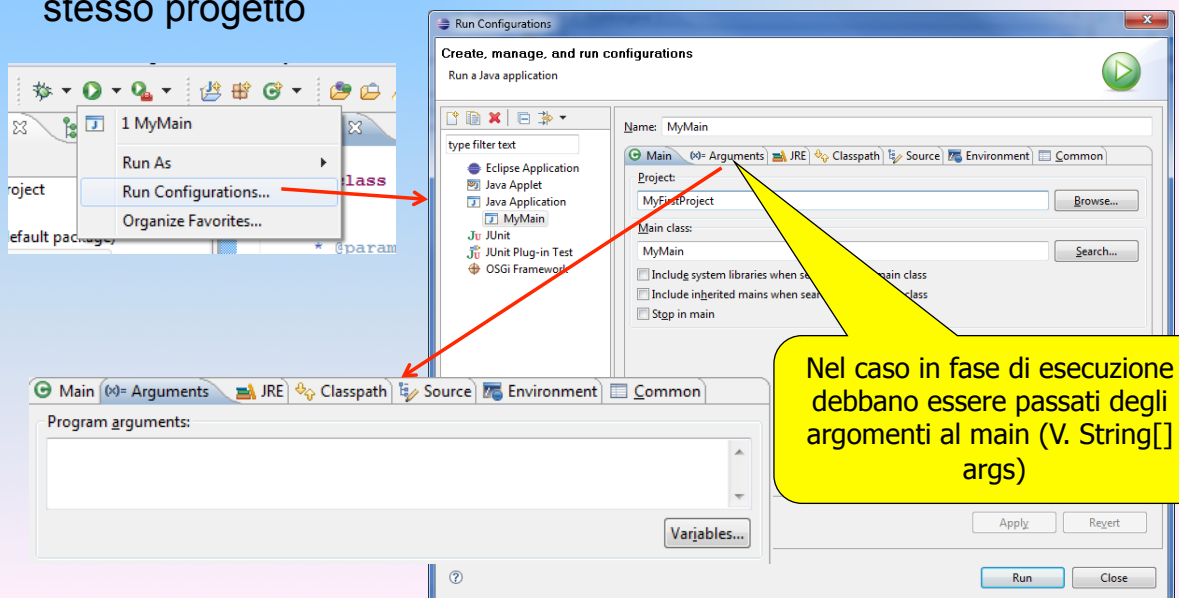
Esecuzione (1/2)



21

Esecuzione (2/2)

- **Run Configurations** permette di impostare parametri di ingresso, runtime, classpath ecc.
- È possibile impostare **più impostazioni** di esecuzione per uno stesso progetto



Debug (1/4)

L'esecuzione dell'applicazione in debug permette di:

- eseguire **passo-a-passo**;
- controllare il valore delle **variabili** e vederle evolvere durante l'esecuzione;
- impostare **breakpoint**;
- visualizzare la catena delle chiamate a metodi.



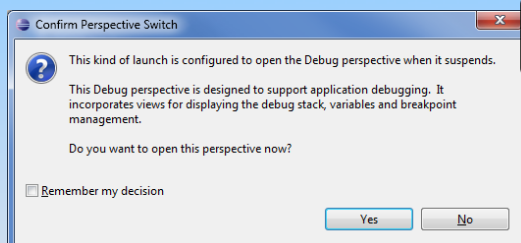
23

Debug (2/4)

Nel codice sorgente è possibile impostare dei **breakpoint**.

Durante l'esecuzione il debug si fermerà in quel punto.

```
public class MyMain {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        int i = 0;  
        i++;  
        System.out.println(i);  
    }  
}
```



Impostazione/disattivazione del breakpoint tramite doppio click in questa zona

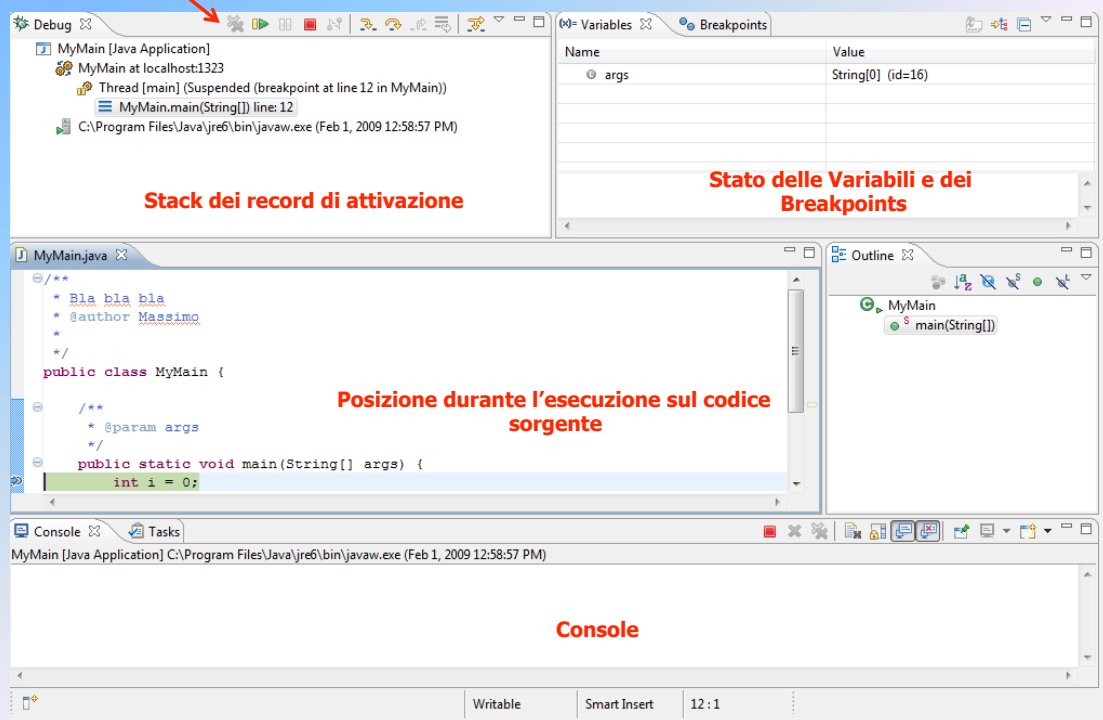
All'avvio del debug verrà richiesto se passare alla **perspective** di debug (...anche sì 😊)

ATTENZIONE!!! Terminato il debug bisognerà ritornare **manualmente** alla perspective Java cliccando sulla corrispondente bottone

24

Debug (3/4) - Prospective

Menu di Esecuzione



25

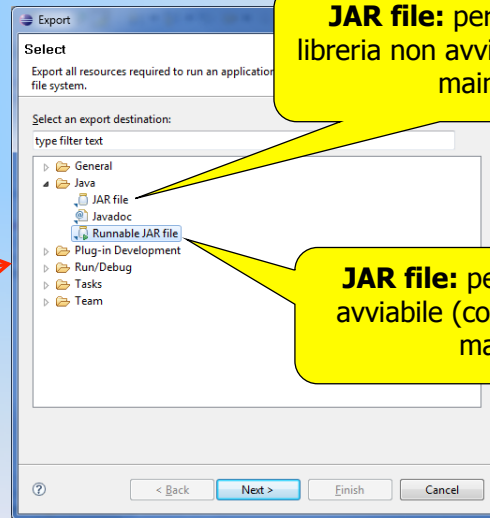
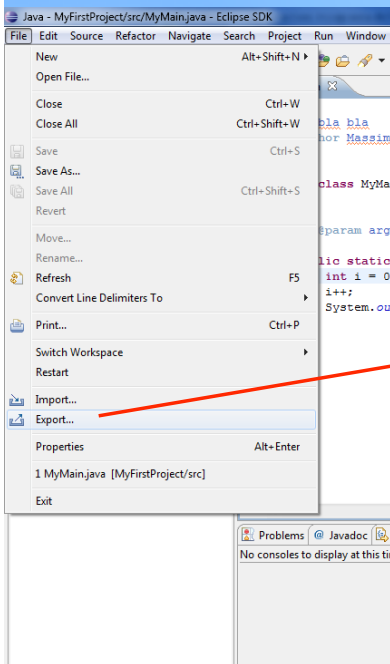
Debug (4/4)

Menu Di Esecuzione

-  → Continua l'esecuzione fino al breakpoint successivo
-  → Mette in pausa l'esecuzione
-  → Ferma L'esecuzione
-  → Entra all'interno del metodo
-  → Prosegue senza entrare nel metodo

26

Creazione Jar

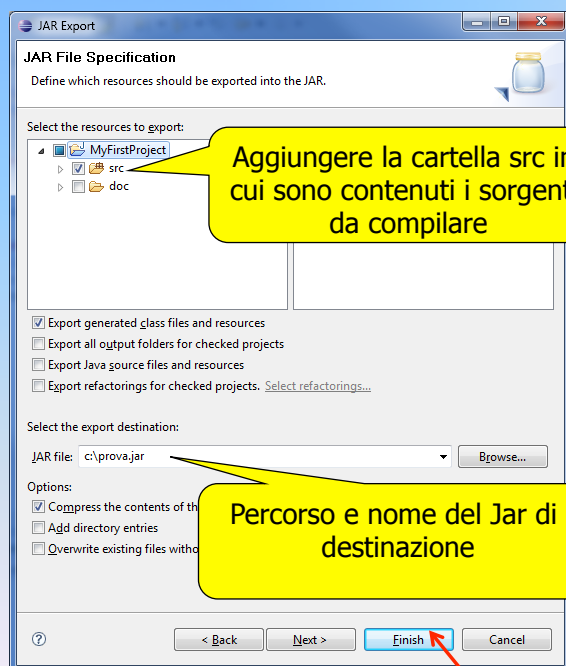


JAR file: per un Jar di libreria non avviabile (senza main)

JAR file: per un file Jar avviabile (con manifest e main)

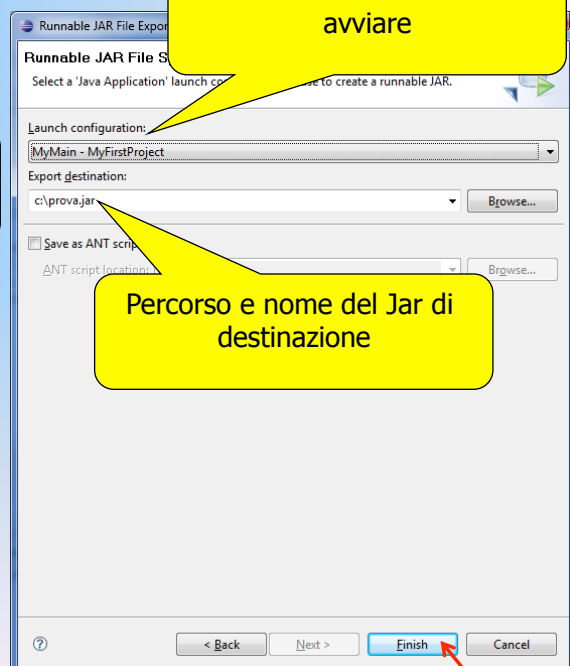
27

Creazione Jar



Aggiungere la cartella src in cui sono contenuti i sorgenti da compilare

Percorso e nome del Jar di destinazione



Classe contenente il Main da avviare

Percorso e nome del Jar di destinazione

28