

# Programmazione a oggetti - Java

Esercitazione 1

Contatti:

Prof. Angelo Gargantini – [angelo.gargantini@unibg.it](mailto:angelo.gargantini@unibg.it)

Dott.ssa Silvia Bonfanti – [silvia.bonfanti@unibg.it](mailto:silvia.bonfanti@unibg.it)

# Ambiente Java

# Cosa contiene

- **JRE** (Java Runtime Environment): è un'implementazione della Macchina virtuale Java, specifica per un S.O. ed un'architettura hardware, che esegue effettivamente i programmi Java.
- **JDK** (Java Development Kit): è una estensione della JRE, ed è inoltre un insieme di software che potete utilizzare per sviluppare le applicazioni basate su Java. Contiene un compilatore Java per trasformare i codici in bytecode, cioè in file di tipo .class. I file vengono successivamente eseguiti.

# Installazione

Java SE Development Kit 8u144		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	<a href="#">jdk-8u144-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u144-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	164.65 MB	<a href="#">jdk-8u144-linux-i586.rpm</a>
Linux x86	179.44 MB	<a href="#">jdk-8u144-linux-i586.tar.gz</a>
Linux x64	162.1 MB	<a href="#">jdk-8u144-linux-x64.rpm</a>
Linux x64	176.92 MB	<a href="#">jdk-8u144-linux-x64.tar.gz</a>
Mac OS X	226.6 MB	<a href="#">jdk-8u144-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.87 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.18 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.gz</a>
Solaris x64	140.51 MB	<a href="#">jdk-8u144-solaris-x64.tar.Z</a>
Solaris x64	96.99 MB	<a href="#">jdk-8u144-solaris-x64.tar.gz</a>
Windows x86	190.94 MB	<a href="#">jdk-8u144-windows-i586.exe</a>
Windows x64	197.78 MB	<a href="#">jdk-8u144-windows-x64.exe</a>

- Collegatevi al sito <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Selezionate «Accept License Agreement»
- Scaricate la versione per il vostro PC
- Installate JDK sul PC (Seguite la procedura guidata)

# Ambiente di sviluppo: Eclipse



# Eclipse

- Eclipse è un ambiente di sviluppo integrato (IDE: Integrated Development Environment). In fase di programmazione aiuta il programmatore nello sviluppo del codice supportandolo con una serie di strumenti e funzionalità (es: segnala gli errori di sintassi).
- Eclipse è anche una comunità open source i cui progetti sono focalizzati a costruire una piattaforma di sviluppo estensibile.

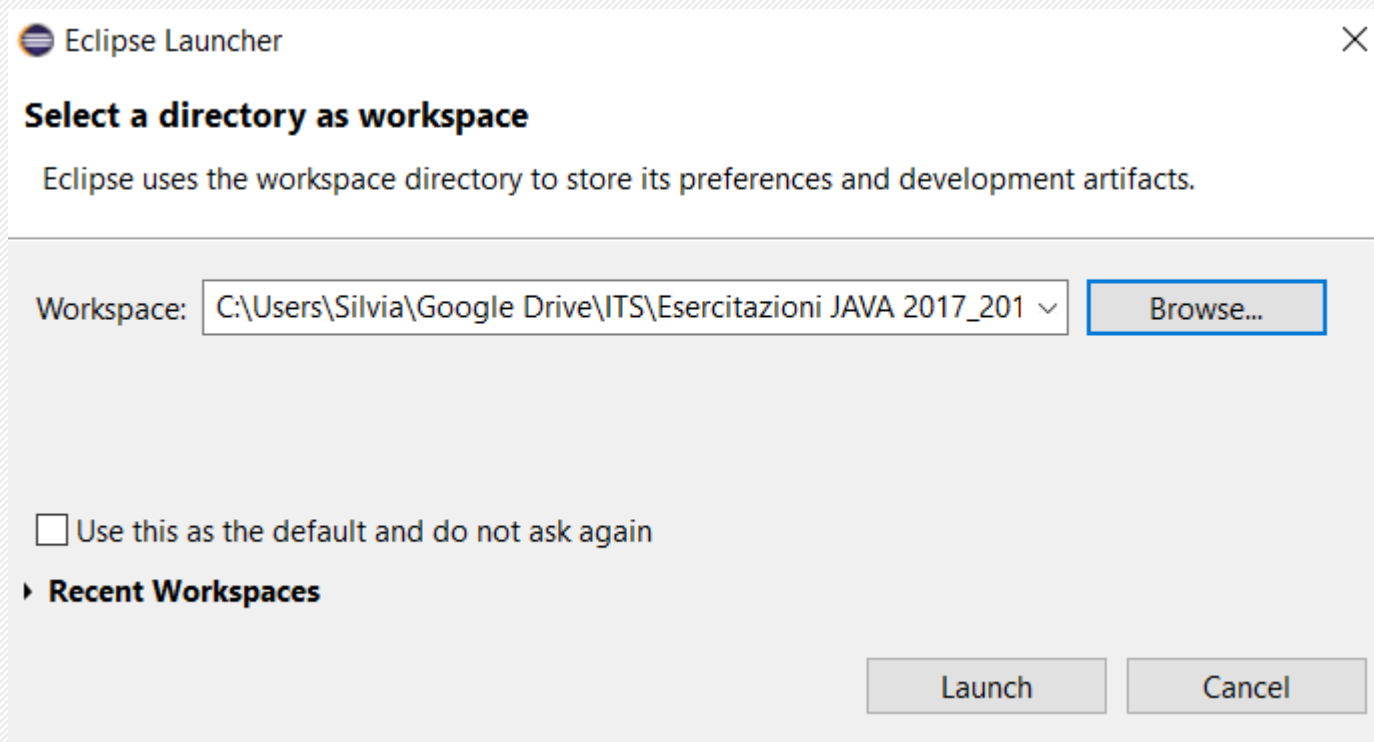
Sui PC dei laboratori c'è già!!!

# Installazione

- Collegatevi al sito <http://www.eclipse.org/downloads/packages/release/Oxygen/M2> e scaricate la versione **Eclipse IDE for Java EE Developers**. In questo modo troverete già l'ambiente di sviluppo configurato per sviluppare programmi Java.
- Il programma è contenuto in un file .zip
- Estraiete il contenuto
- Per avviare Eclipse è sufficiente eseguire il file eclipse.exe

# Primo avvio (1/3)

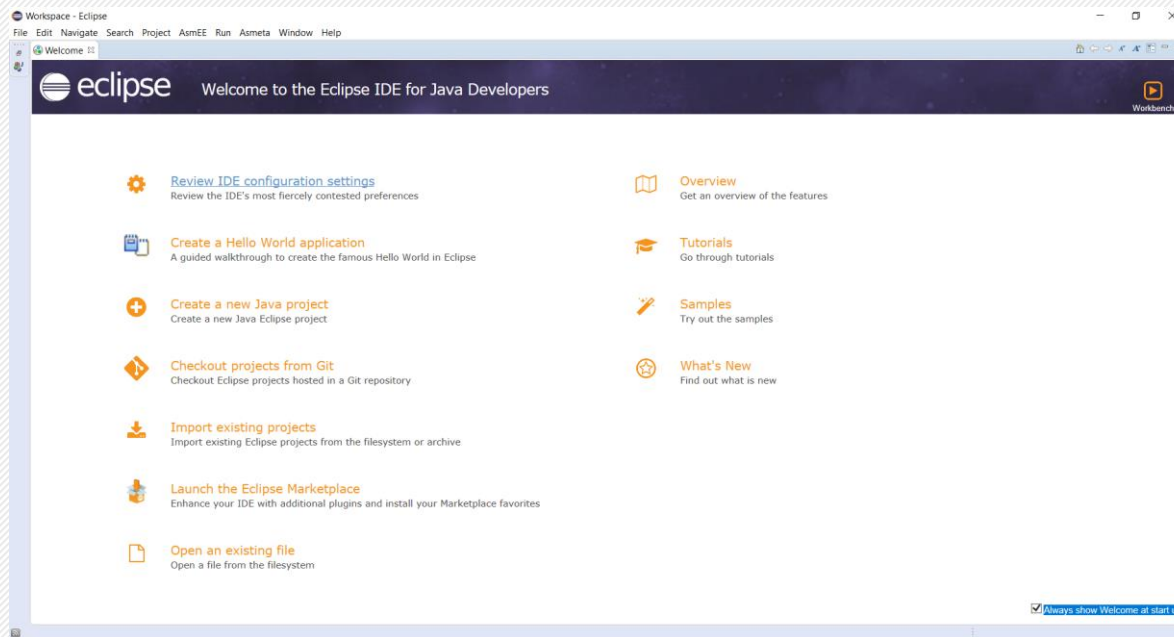
- Selezionare il **Workspace** (spazio di lavoro): è una cartella dove verranno salvati i file





# Primo avvio (2/3)

- Chiudete la schermata di benvenuto e togliete il flag dall'opzione «Always show Welcome at start up»



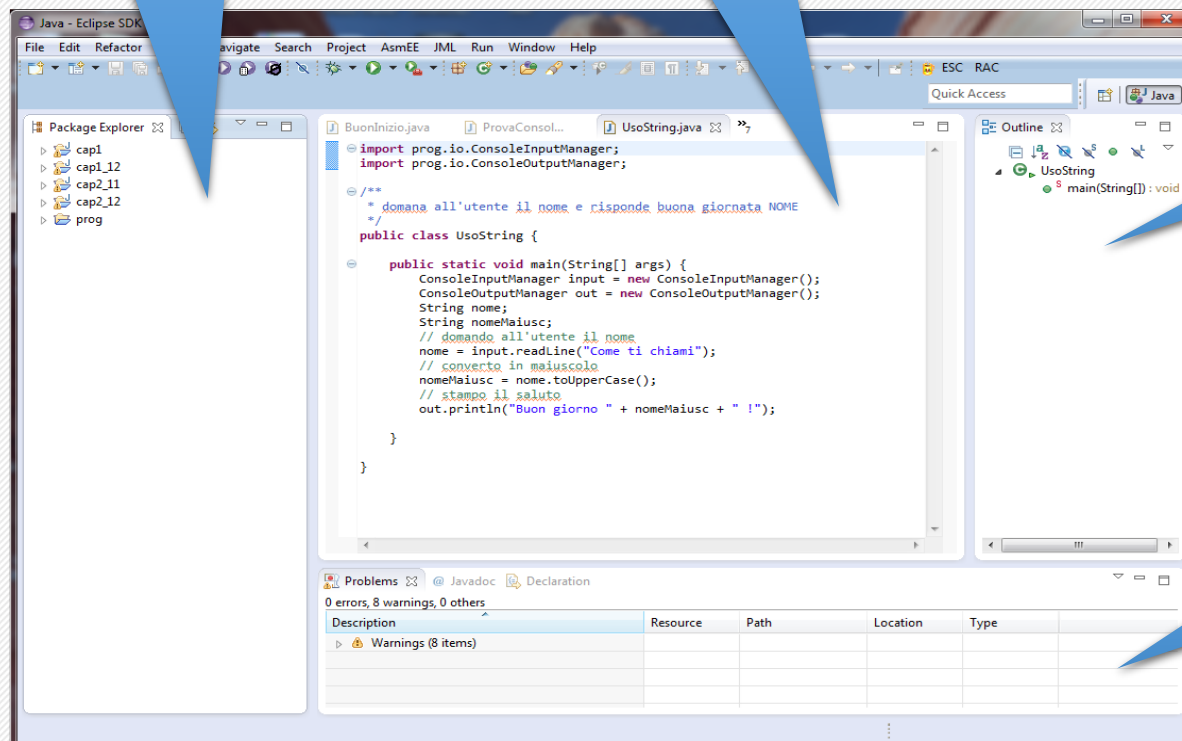
## Primo avvio (3/3)

# Progetti

# Programa .java

# Outline

## Console Problems list

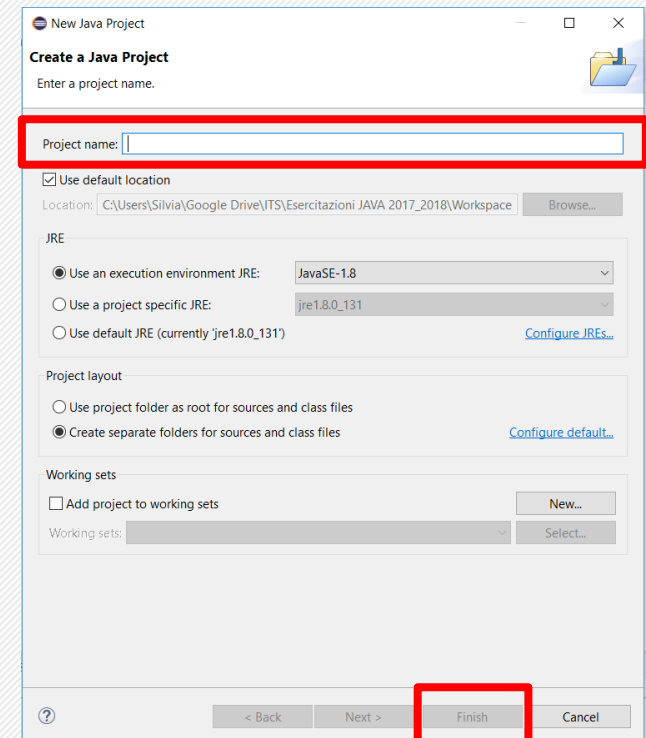


# Progetto Java

- Prima di iniziare a creare il programma Java, è necessario creare un progetto che conterrà i vostri file .java

File -> New -> Java Project

- Inserite il nome del progetto (primoProgetto)
- Cliccate su «Finish»
- Nel vostro workspace trovate un progetto vuoto



# Classe Java

- Per iniziare a scrivere il vostro primo programma dovete aggiungere all'interno del progetto un file di tipo .java

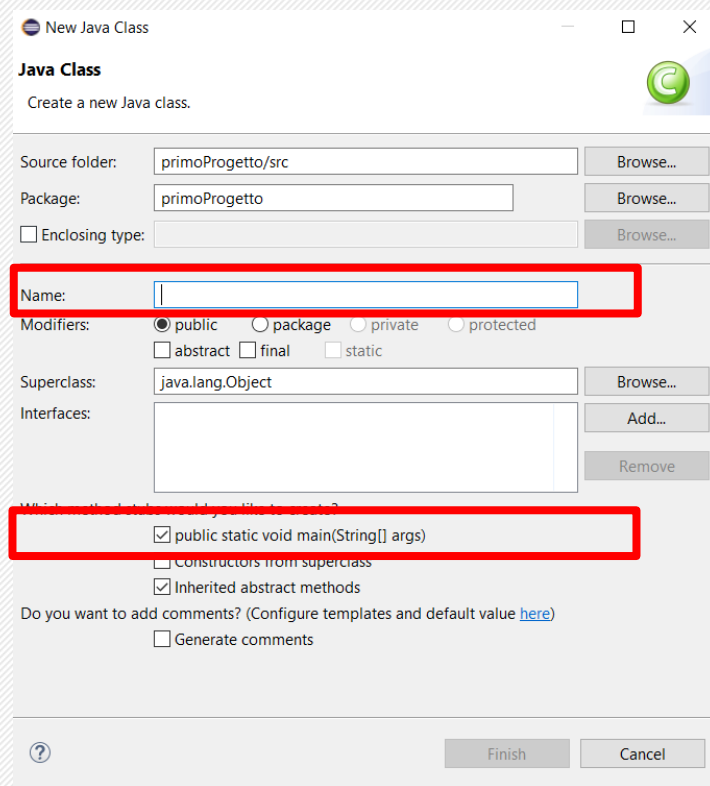
File -> New -> Class

- Inserite il nome del file (es: HelloWorld)
- **Aggiungete il metodo MAIN!!**

Mettete il flag sull'opzione «public static void main (String[] args)»

- Cliccate su «Finish»

- Per convenzione il nome delle classi inizia con la lettera maiuscola
- Le classi vengono create in:  
nomeprogetto/src/nomepackage



# Primo programma

Scrivete il seguente codice all'interno del vostro file:

```
package primoProgetto;  
public class HelloWorld {  
  public static void main(String[] args)  
  {  
    System.out.println("Hello World");  
  }  
}
```

# Compiliamo ed eseguiamo il programma

- Cliccate con il tasto destro sulla classe o sul progetto

*Run As -> Java Application*


Oppure aprite il file java ed utilizzate questo pulsante nella barra in alto



- Il risultato viene visualizzato all'interno della «Console»

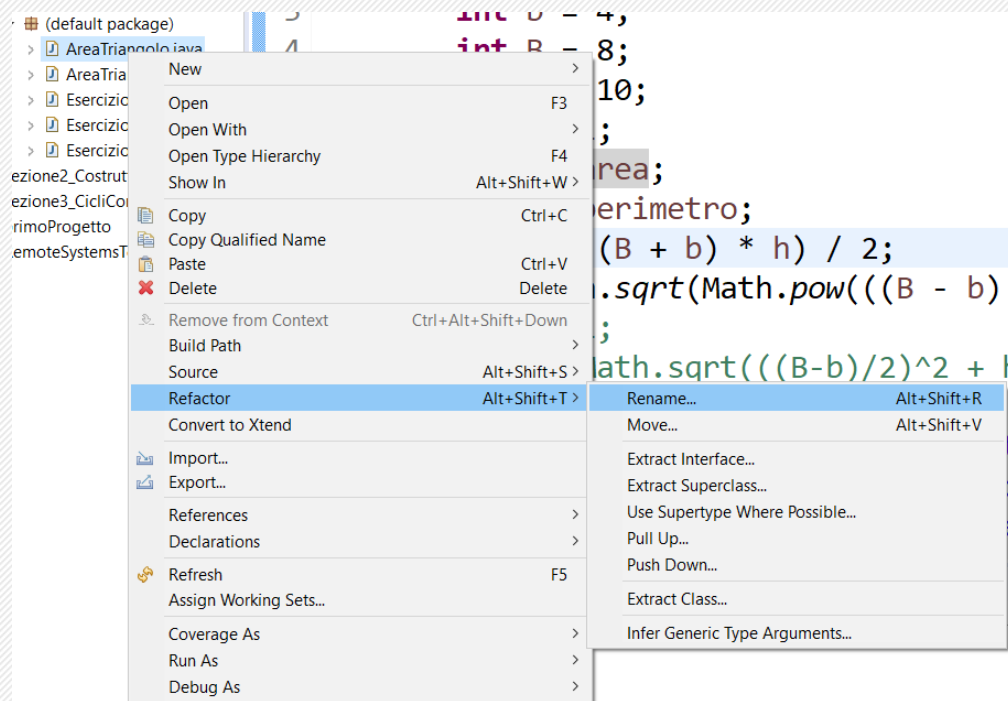
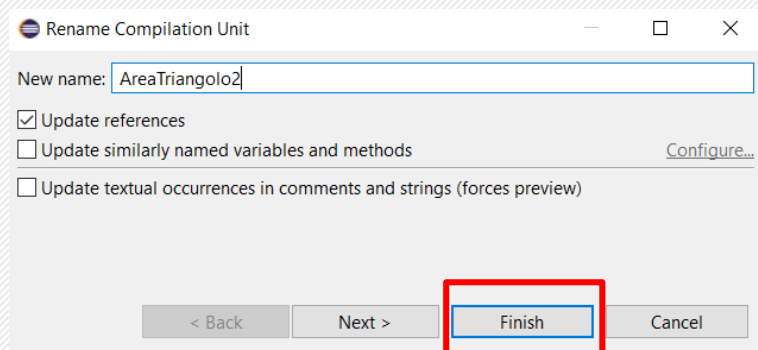


# Note:

- Un progetto java per poter essere eseguito deve contenere almeno una classe in cui è implementato il metodo **main**!
- Un progetto java può contenere più classi contenenti il metodo main: in questo caso per eseguire una specifica classe cliccare con il tasto destro sulla classe, poi *Run As -> Java Application* oppure apri la classe e utilizza il pulsante 
- Se il progetto contiene una sola classe con il metodo main puoi eseguire la classe cliccando con il tasto destro sul progetto, poi *Run As -> Java Application*

# Come rinominare un file .java?

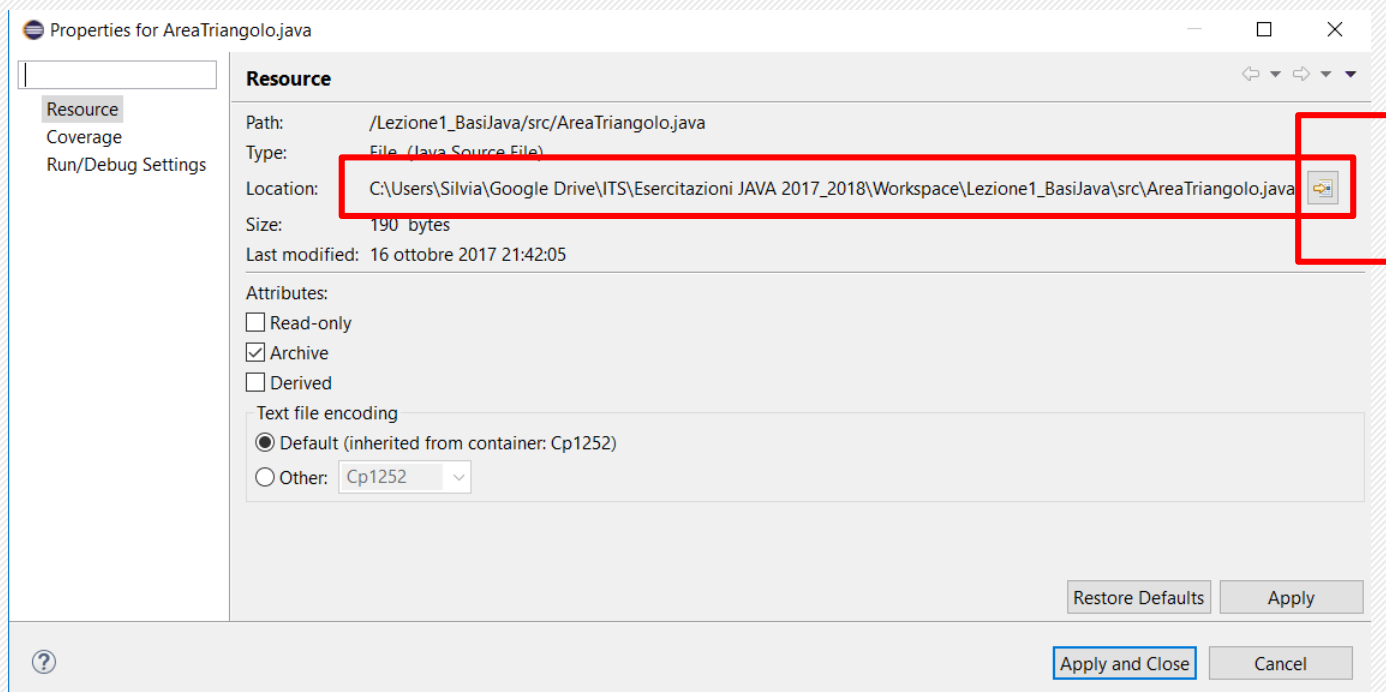
- Click con il tasto destro sulla classe da rinominare -> Refactor -> Rename





# Dove vengono salvati il progetto e i file .java?

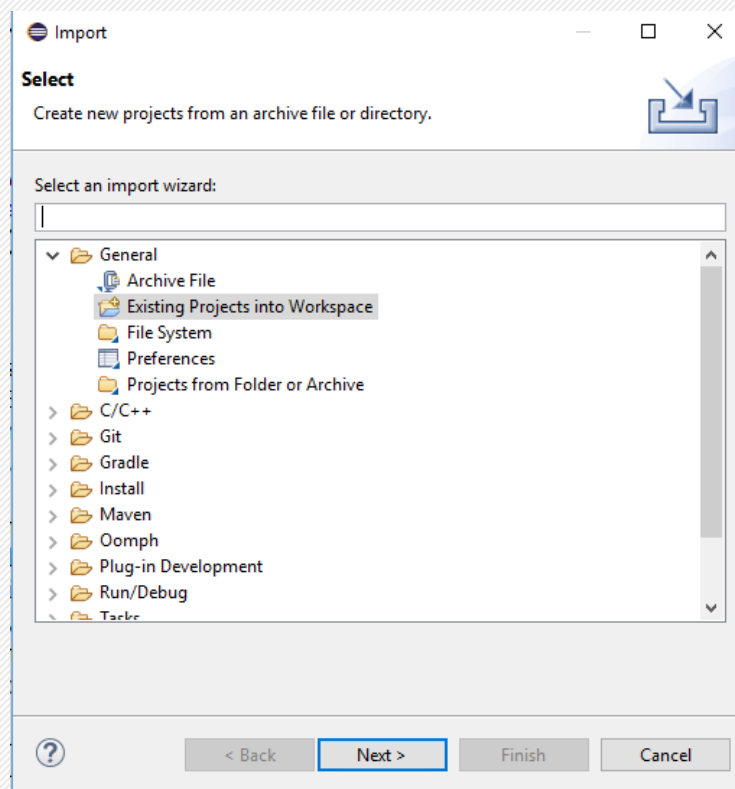
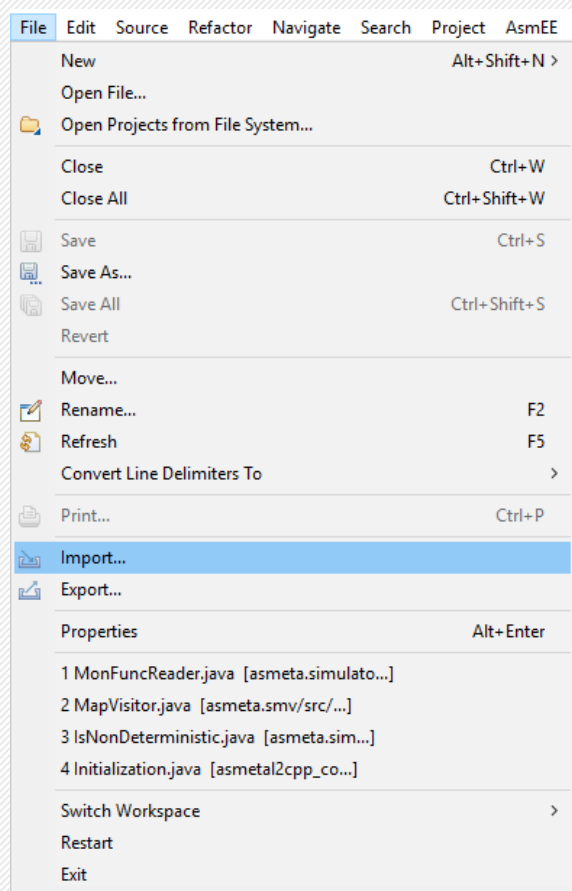
- Click con il tasto destro sulla classe/progetto di cui si vuole conoscere il percorso -> Properties



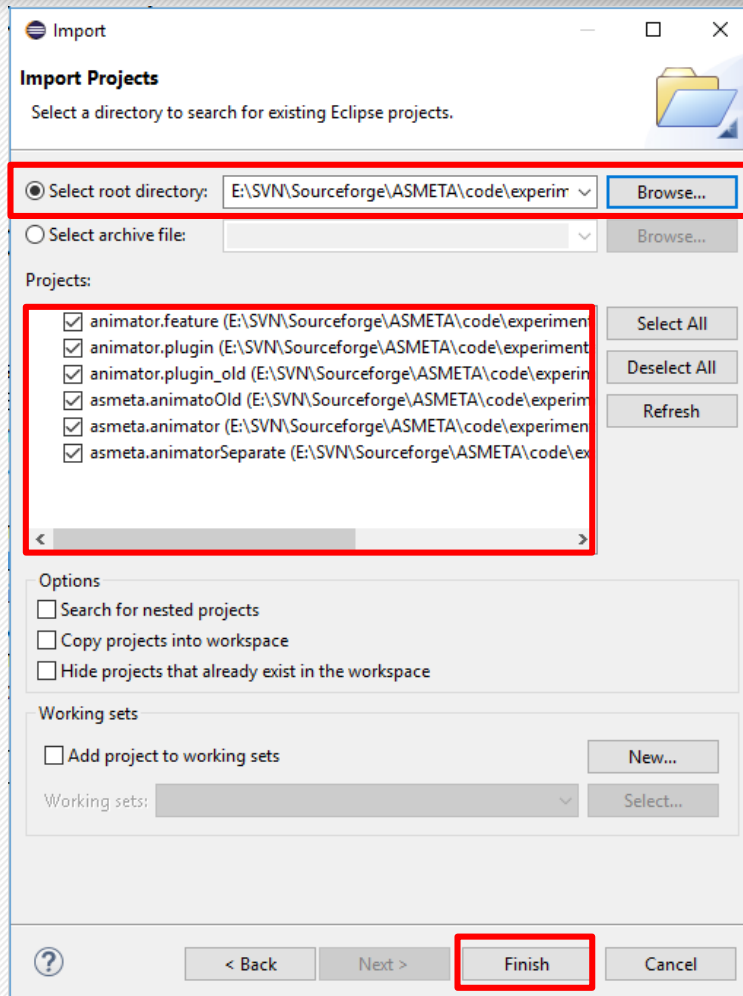
# Come esportare un progetto esistente

- Apri la cartella in cui è contenuto il progetto e copia la cartella del progetto su una chiavetta/..

# Come importare un progetto esistente (1/2)



# Come importare un progetto esistente (2/2)



- Selezionare la directory in cui è contenuto il progetto da importare
- Selezionare il/i progetto/i da importare
- Cliccare su «Finish»

ConsoleOutputManager  
ConsoleInputManager

# Esercizio 1 - guidato

- Creare un nuovo progetto che stampa sulla console una qualsiasi stringa utilizzando la classe **ConsoleOutputManager**.

# Come usare console output manager

```
package primoProgetto;

import prog.io.ConsoleOutputManager;

public class PrimoProgramma {
    public static void main(String[] args)
    {
        ConsoleOutputManager video= new ConsoleOutputManager();
        video.println("Ecco il mio primo programma!");
    }
}
```

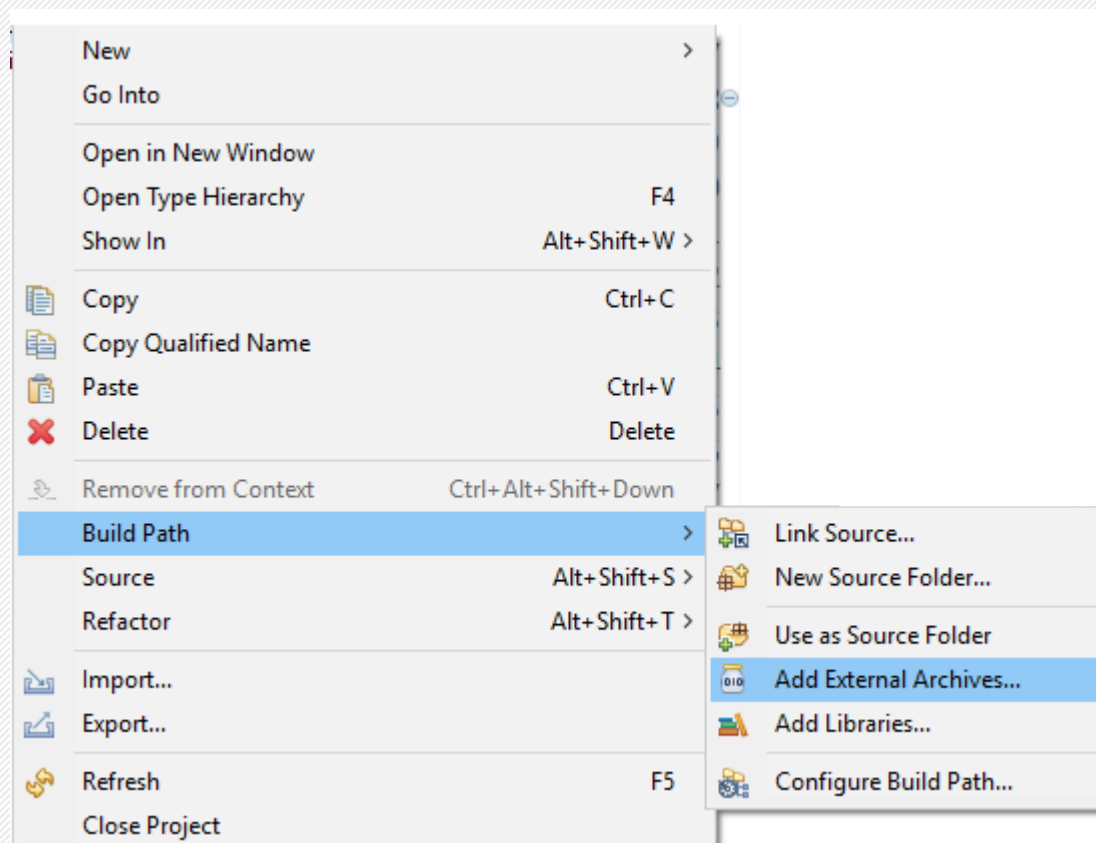
# Come aggiungere la libreria **prog**?

- Come aggiungere prog.zip o progXX.jar al mio progetto eclipse ?
  - Oltre alle librerie standard di Java posso aggiungere nuove librerie (.jar o .zip) che contengono dei file class al mio progetto
  - In questo modo posso usare classi definite da altri o da me in altri progetti oltre alla classi standard
  - Devo modificare build path (per poter compilare) o class path

La libreria progXX.jar la trovate su ilias!!!!



# Aggiungere i jar



Attenzione!!

In questo modo scelgo il file jar dal file system e viene memorizzato il percorso assoluto. Quindi se sposto il progetto su un altro pc non trova più il file .jar

Seleziona il file progXX.jar

# Percorso relativo per le librerie

- Se voglio poter spostare tutti i miei progetti da un PC ad un altro (ad esempio su una chiavetta), posso fare così:
  1. metto il prog.jar in una directory del mio progetto (new -> folder)
    - Eseguo la stessa procedura nella slide precedente aggiungendo il file .jar aggiunto al progetto (così però se ho tanti progetti, devo copiare la libreria in tutti i progetti)
  2. creo un progetto esterno in cui metto i miei jar
    - Nel progetto in cui devo utilizzare la libreria faccio tasto destro -> Build Path -> configure Build Path. Nella scheda Libraries clicco su “Add Jar” e seleziono il jar che mi interessa dal progetto che contiene tutti i jar

## Esercizio 2

- A. Rifai l'esercizio 1 sul tuo PC
  - B. Creare un nuovo progetto che chieda all'utente di inserire da console una stringa e stampare a video la stessa.
  - C. Creare un nuovo progetto che chiede due numeri all'utente e ne stampa la somma.
- 
- Gli esercizi sono da fare utilizzando la libreria prog.io fornita dal libro di testo.

## Esercizio 2\_A - passi

1. Creo un nuovo progetto
2. Creo una nuova classe (con main)
3. Importo prog.jar nel buildpath
4. Edito il file
5. Creo un oggetto ConsoleOutputManager (con new)
6. Utilizzo il metodo print/println per stampare  
Per vedere quali sono i metodi disponibili:  
nomevariabile. + (CTRL+Spazio)
7. Eseguo con run

# Esercizio 2\_B 2\_C - suggerimenti

- Per leggere da tastiera i valori/stringhe/... inseriti dall'utente usate

## **ConsoleInputManager**

- Le sue istanze realizzano canali di comunicazione con il dispositivo di input standard, cioè la tastiera.
- Metodi:
  - `readLine` legge una riga di testo
  - `readInt` legge un numero intero
  - ...

# Frazioni

## Esercizio 3 - guidato

- Scrivi un programma che crea due frazioni  $\frac{1}{3}$  e  $\frac{2}{3}$ , calcola e stampa la somma, la differenza, il prodotto e il quoziente

Utilizzate la classe `Frazione` del package `prog.utili`

## Esercizio 4

- Scrivi un programma che domanda all'utente il numeratore e denominatore di due frazioni e come sopra stampa somma, differenza, prodotto e quoziente

Utilizzate la classe `Frazione` del package `prog.utili`



# Esercizio 5

- Si scriva un programma che prenda in input delle frazioni, dopo aver inserito una frazione viene richiesto all'utente se ne vuole inserire un'altra.
- Il programma calcolerà la media tra le frazioni inserite.
- Per calcolare la media basta memorizzare la somma delle frazioni inserite
- Note:
  - Utilizzare un ciclo while che termina quando l'utente inserisce No
  - Per leggere quello che inserisce l'utente puoi usare `readSiNo` per leggere la risposta e poi `readInt` per leggere numeratore e denominatore della frazione, oppure puoi usare `readLine` e se il testo inserito non contiene "no" convertilo in intero usando il metodo `Integer.parseInt(Strings)`

Stringhe

# Esercizio 6

- Scrivere un semplice programma che faccia apparire una cornice di asterischi sopra e sotto lunga quanto la stringa inserita

```
*****
```

```
Ciao, come va?
```

```
*****
```

- Dichiarare qualche variabile di appoggio per memorizzare le stringhe
- Usare substring e length
- Non usare cicli for

# Esercizio 7

- Scrivere un semplice programma che faccia apparire frase inserita incorniciata da asterischi:

```
*****
```

```
*                                     *
```

```
*      Ciao, come va?              *
```

```
*                                     *
```

```
*****
```

- Dichiarare qualche variabile di appoggio per memorizzare la stringa
- Usare `substring`, `length` e `concat`

## Esercizio 8

- Scrivere un programma che domanda una frase e la riscrive senza le vocali
- A. Usa un unico main
  - B. Usa il sottometodo isVocale
  - C. Riscrivi isVocale usando lo switch con case multipli, cioè:

```
switch(c){  
  case 'a':  
  case 'A':  
  ...
```

## Esercizio 9

- Scrivere un programma che domanda una stringa e conta le vocali
  - Usare `charAt` per trovare l'i-esimo carattere di una stringa
- A. Usa un unico `main`
  - B. Metti la funzionalità di conteggio in un sottometodo (`static`) che data una stringa restituisce il numero di vocali
  - C. Scrivi un altro metodo `isVocale` che dato un carattere dice se è vocale o no

# Esercizio 10

- Si scriva un'applicazione che legge da tastiera una stringa di caratteri e che stampa a video il numero di occorrenze di ciascuna lettera nella stringa (solo se occorre).
- Esempio “pippo”
  - i -> 1
  - o -> 1
  - p -> 3

# Esercizio 11

- Si scriva un'applicazione che legge da tastiera una stringa di caratteri e che stampa a video il numero di occorrenze di ciascuna lettera nella stringa (solo se occorre).
- Esempio “pippo”
  - i -> 1
  - o -> 1
  - p -> 3



# Esercizio 11 - suggerimenti

- Usa due for annidati
- Usa un ciclo for con char o int da 'A' a 'z' per il carattere da cercare
- Usa un ciclo interno for per la lunghezza della stringa per scorrere la stringa
- Usa un contatore
- Usa charAt per prendere un carattere in una stringa