



Alma Mater Studiorum-Università di Bologna

Scuola di Ingegneria

Fondamenti di Informatica T2

Frazioni: seconda parte

Corso di Laurea in Ingegneria Informatica

Anno accademico 2021/2022

Prof. ROBERTA CALEGARI

Prof. AMBRA MOLESINI

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



Agenda Esercitazione

- Realizzare la versione estesa del componente **Frazione**
- **Frazione**
 - tipo di dato astratto (ADT)
 - progettato come tipo “valore” (oggetto immutabile: una volta “nato” il suo valore è **costante**)

Tutto adeguatamente e costantemente collaudato
→ test, test, test!



Frazione: puntata precedente

- **Costruire** una nuova frazione partendo da numeratore e denominatore
- **Ottenere** numeratore e denominatore della frazione
- **Verificare** se la frazione è uguale ad una data (***equals***)
- **Ottenere** una nuova frazione che rappresenti lo stesso valore in forma ***ridotta ai minimi termini***
- **Stampare** la Frazione sulla console (***toString***)



Frazione: versione estesa

- **Sommare** la frazione con un'altra: si ottiene una nuova frazione che rappresenta il risultato ***ridotto ai minimi termini***
- **Moltiplicare** la frazione con un'altra: si ottiene una nuova frazione che rappresenta il risultato ***ridotto ai minimi termini***
- **Sottrarre** la frazione con un'altra: si ottiene una nuova frazione che rappresenta il risultato ***ridotto ai minimi termini***
- **Dividere** la frazione con un'altra: si ottiene una nuova frazione che rappresenta il risultato ***ridotto ai minimi termini***
- **Invertire** la frazione: si ottiene una nuova frazione che rappresenta il reciproco della frazione data e che è ***già ridotta ai minimi termini***

....



Frazione: versione estesa

- **Confrontare** la frazione con un'altra (compareTo): si ottiene un intero con valori
 - *0 se le due frazioni sono uguali*
 - *1 se la Frazione è maggiore della Frazione ricevuta in ingresso*
 - *-1 se la Frazione è minore della Frazione ricevuta in ingresso*
- **Il valore double** associato: viene restituito un **valore di tipo double** che rappresenta il numero reale equivalente della funzione

ANALISI REQUISITI: *sum*

Metodo classico, sfruttando *mcm*:

$$8/5 + 6/15 = (24 + 6) / 15 = 2/1$$

```
public Frazione sumWithMcm (Frazione f) {  
    int mcm = MyMath.mcm(f.den, this.den);  
    ...  
}
```

Usare il metodo **mcm**
dalla libreria **MyMath**

Metodo alternativo

$$8/5 + 6/15 = (8*15 + 6*5) / (15*5) = (120+30) / 75 = 150/75 = 2/1$$

```
public Frazione sum(Frazione f) {  
    int n = num * f.den + den * f.num;  
    int d = den * f.den;  
    Frazione fSum = new Frazione(n, d);  
    return fSum.minTerm();  
}
```

Analogamente si procede
per le altre operazioni

ANALISI REQUISITI: *il modello*

Frazione
<ul style="list-style-type: none">- den: int- num: int
<ul style="list-style-type: none">+ compareTo(f: Frazione): int+ div(f: Frazione): Frazione+ equals(f: Frazione): boolean+ Frazione(num: int, den: int)+ Frazione(num: int)+ getDen(): int+ getDouble(): double+ getNum(): int+ minTerm(): Frazione+ mul(f: Frazione): Frazione+ reciprocal(): Frazione+ sub(f: Frazione): Frazione+ sum(f: Frazione): Frazione+ sumWithMcm(f: Frazione): Frazione+ toString(): String



E Ora... Al Lavoro!

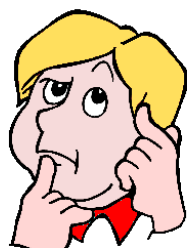
Completare l'ADT con i metodi della versione estesa

- Metodo `sum` e `sumWithMcm` per sommare la frazione con un'altra,
- Metodo `sub` per sottrarre la frazione con un'altra
- Metodo `mul` per moltiplicare la frazione con un'altra
- Metodo `div` per dividere
- Metodo `reciprocal` per invertire la frazione
- Metodo `compareTo` per confrontare la frazione con un'altra
- Metodo `getDouble` per restituire il valore reale equivalente

Sempre → collaudo continuo e sistematico!

Startkit

- Fondamentale imparare ad importare ed usare gli startkit..
- .. In quasi tutte le esercitazioni vi forniremo noi parte del codice...
- ..**anche all'esame avrete uno startkit**



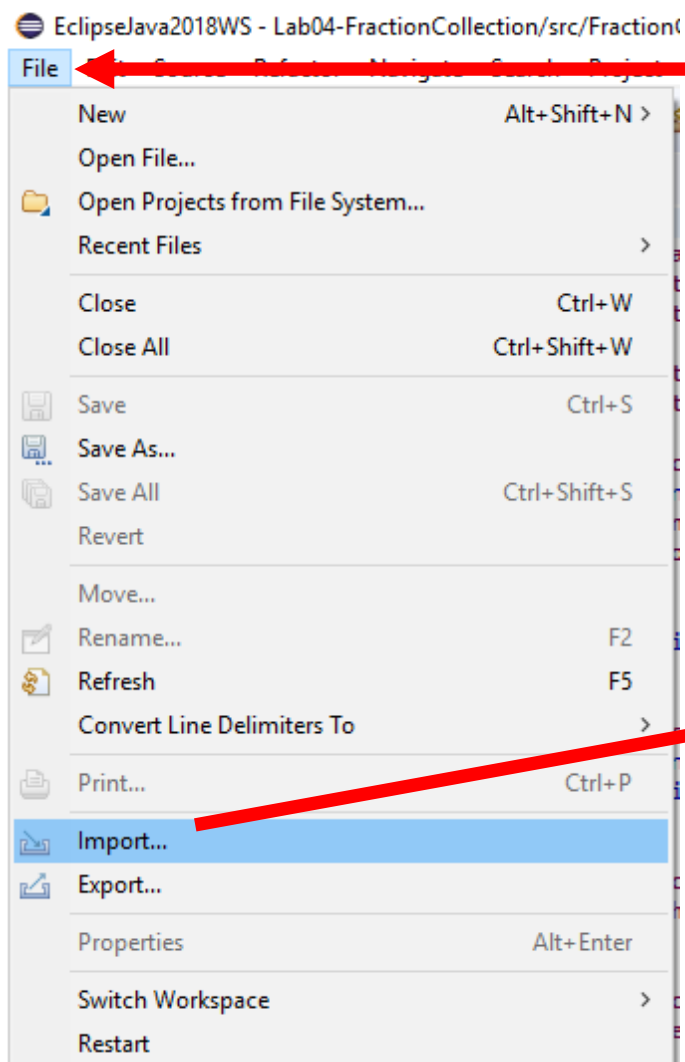
**Ma come lo uso in
Eclipse???**

Startkit

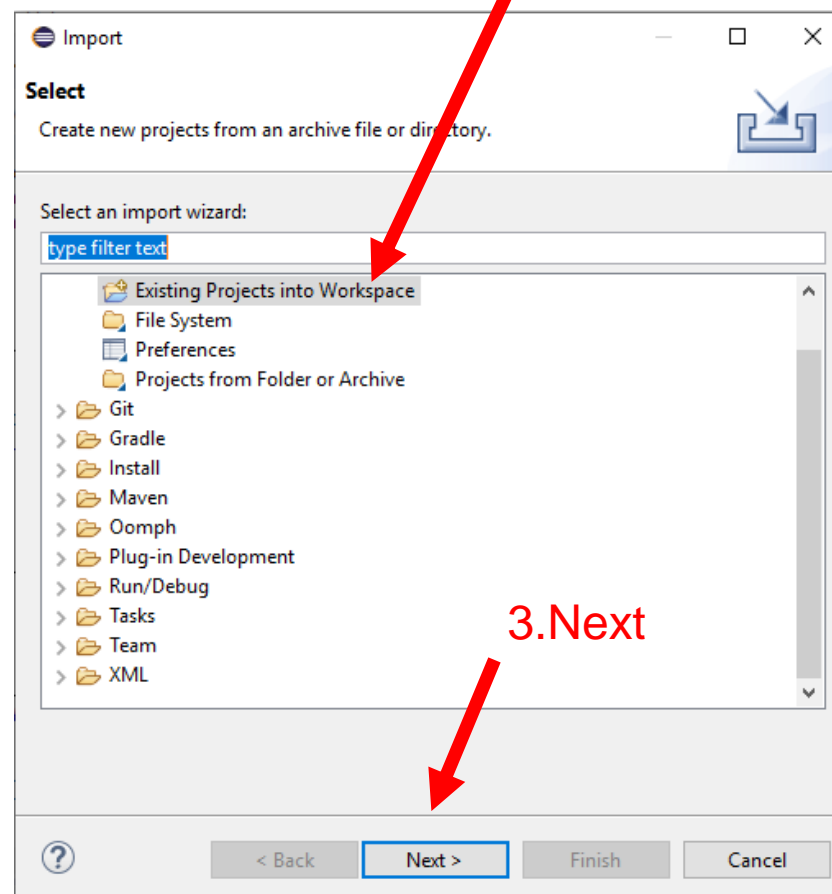
- Startkit = zip che contiene un progetto Eclipse
- **Passi per l'uso dello startkit**
 1. Scarico lo startkit da Virtuale (da eol se siamo all'esame)
 2. Decompatto lo startkit e rinomino la cartella sul FileSystem
 3. Apro Eclipse
 4. Importo la cartella rinominata in Eclipse
 5. Rinomino il Progetto dentro Eclipse

Attenzione: Fondamentale imparare questa sequenza di attività per l'esame se non volete perdere 1 punto subito!!!

Importo Startkit in Eclipse



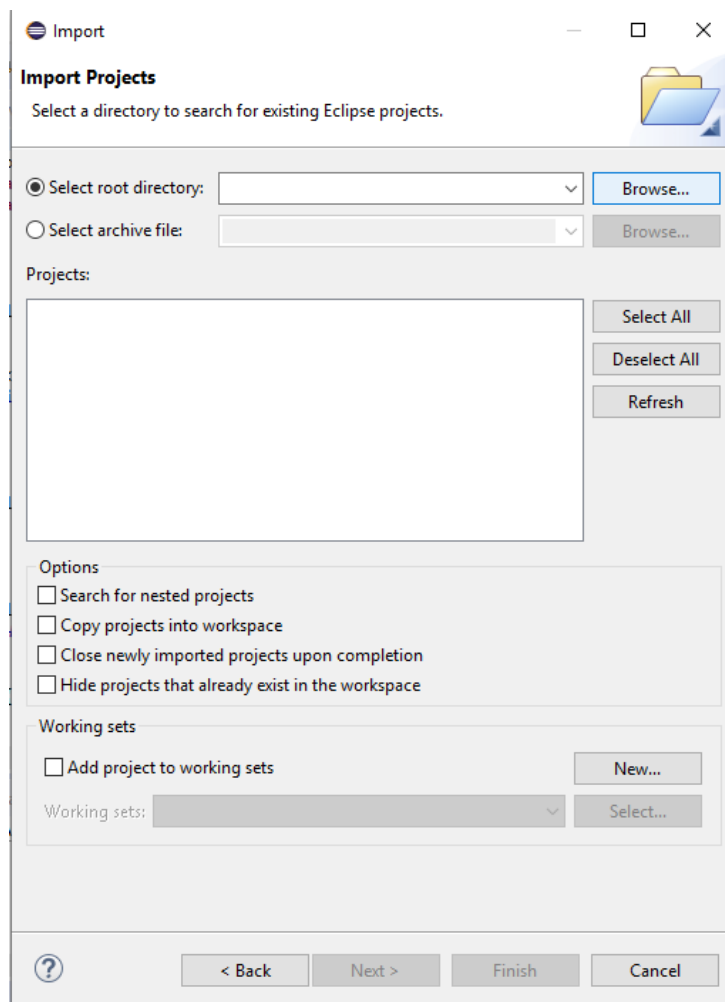
1. Dal menù File
scegliere «Import»



2. Scegliere

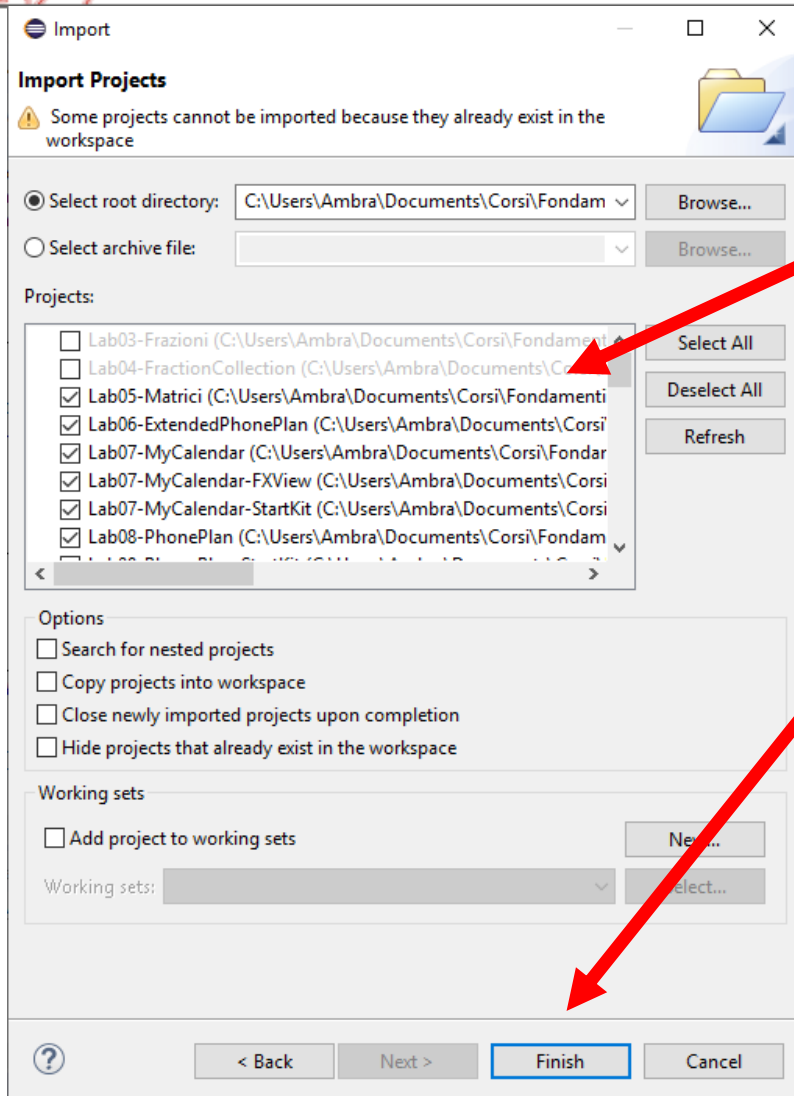
3. Next

Importo Startkit in Eclipse



4. Con Browse scegliere la cartella dove avete salvato lo startkit

Importo Startkit in Eclipse

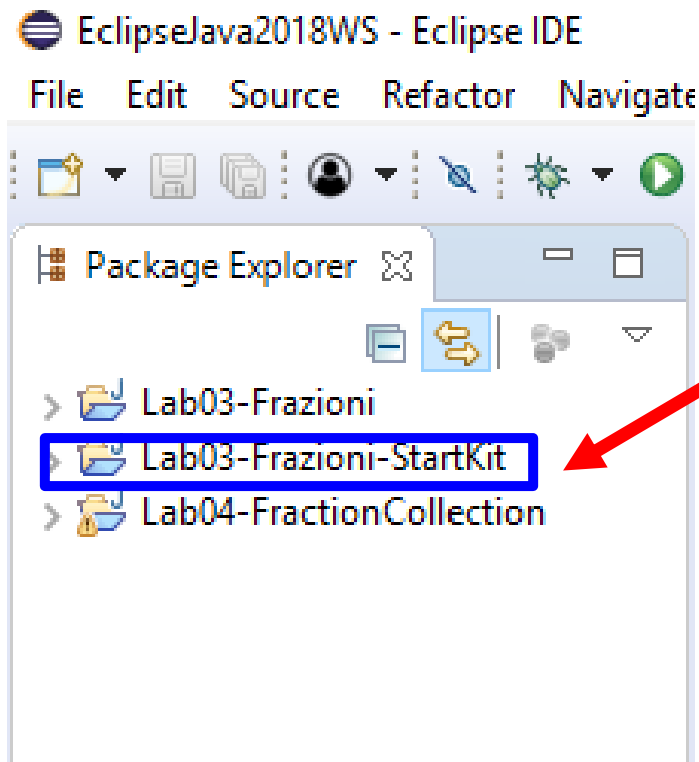


5. Eclipse vi elencherà tutti i progetti presenti nella cartella che avete scelto

6. Selezionate il progetto (o i progetti) a cui siete interessati e poi premete Finish

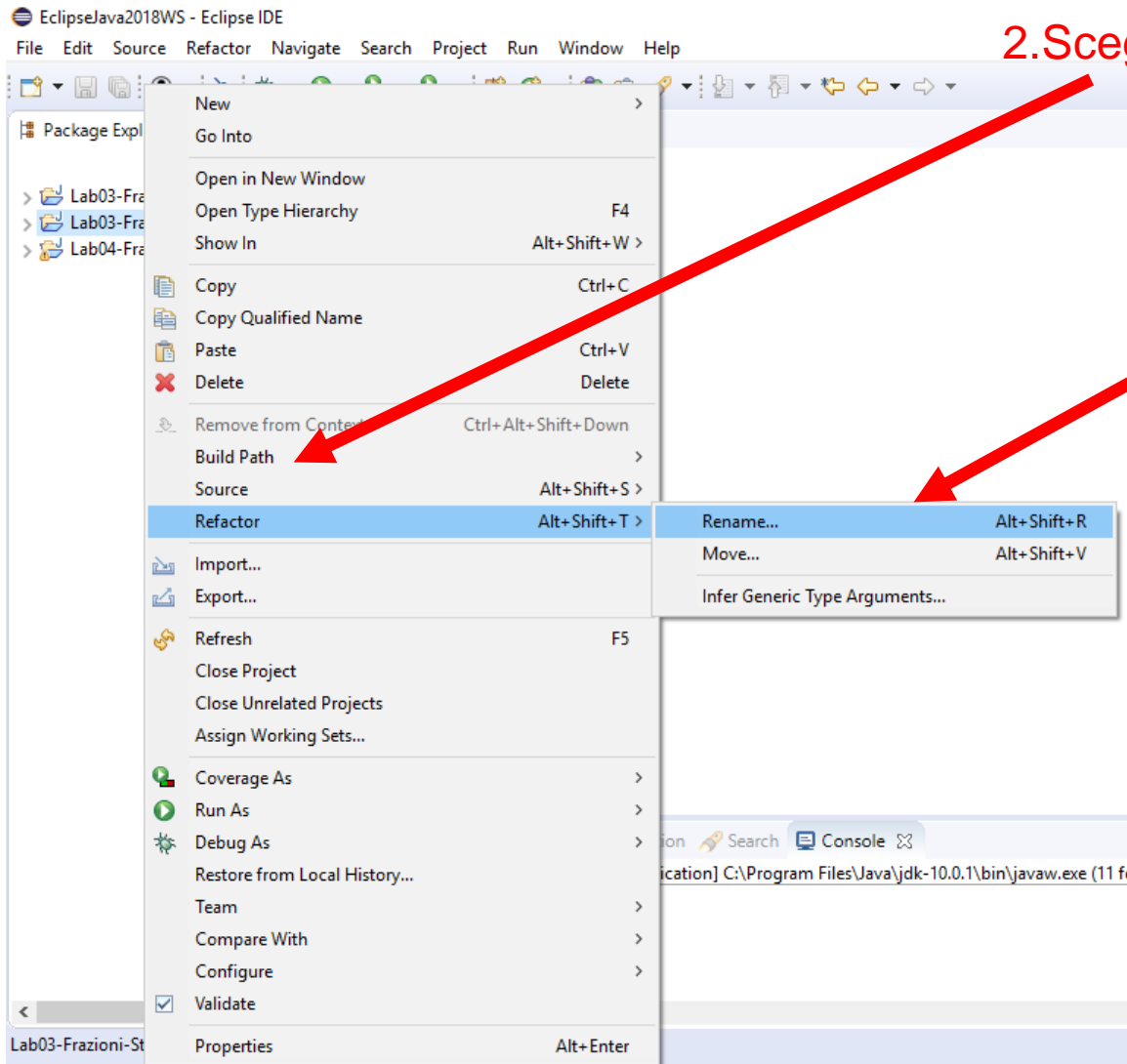
7. Pazientate un attimo e vedrete comparire tutti i progetti: Eclipse ricompila il workspace e se avete scelto molti progetti potrebbe metterci un po'

Rinomino Startkit in Eclipse



1. Tasto destro su Lab03-Frazioni-StartKit

Rinomino Startkit in Eclipse

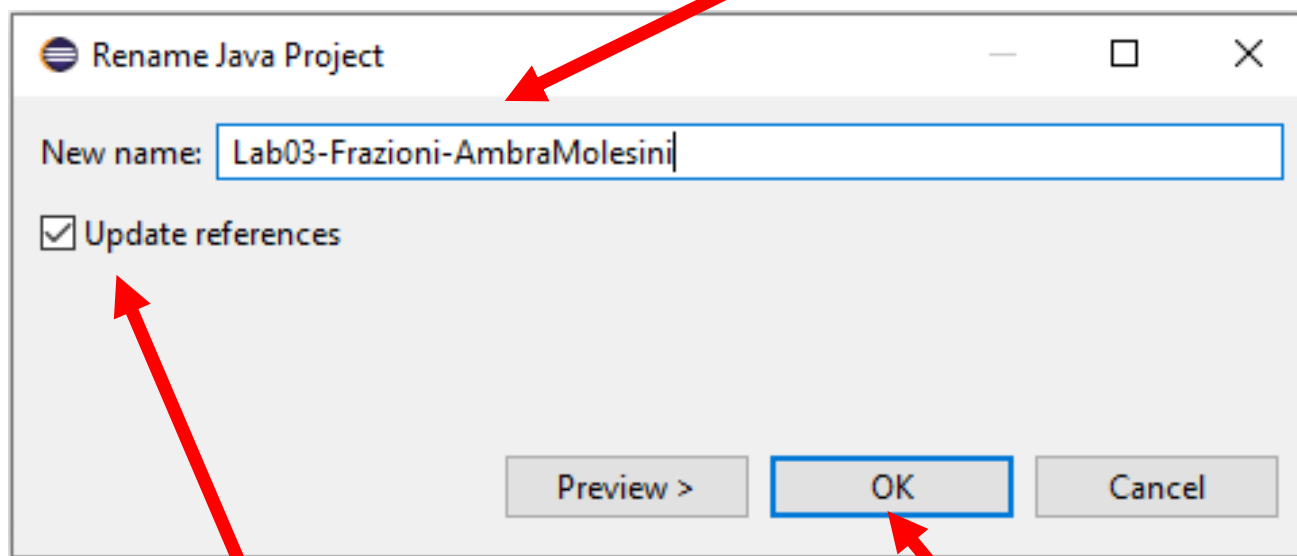


2. Scegliere Refactor

3. Scegliere Rename

Rinomino Startkit in Eclipse

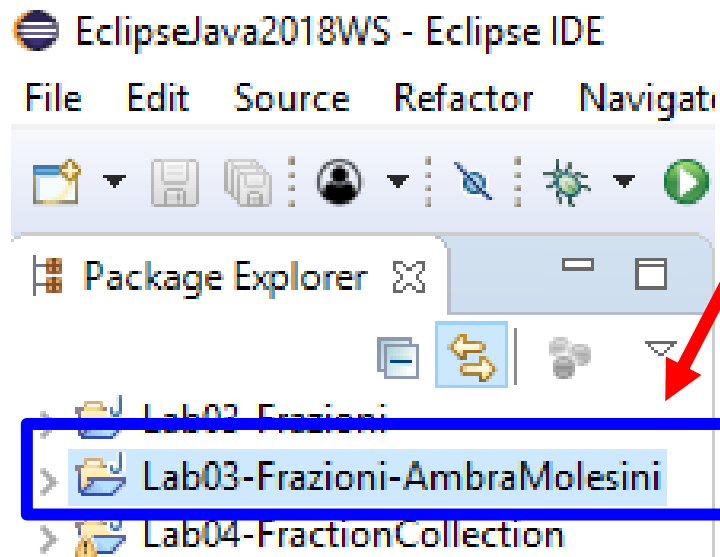
4. Dare un nuovo nome al progetto



5. Attenzione che sia selezionato →
Eclipse aggiorna per voi tutte le
dipendenze incrociate con altri progetti

6. Premere OK

Rinomino Startkit in Eclipse



Il progetto è stato rinominato

**Cruciale che lo
impariate a fare!!!
All'esame**

DOVRETE FARLO
**(-1 punto se progetto
non rinominato)**

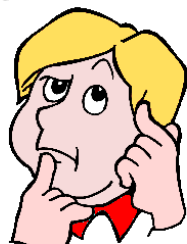
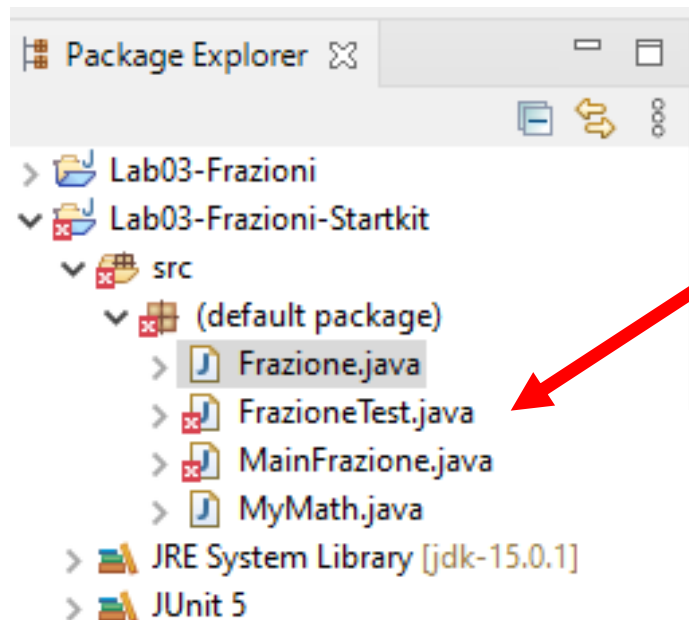
ESAME DI FONDAMENTI DI INFORMATICA T-2 del 4/2/2021

Proff. E. Denti – R. Calegari – A. Molesini

Tempo a disposizione: 3 ore

NOME PROGETTO ECLIPSE:	CognomeNome-matricola (es. RossiMario-0000123456)
NOME CARTELLA PROGETTO:	CognomeNome-matricola (es. RossiMario-0000123456)
<u>NOME ZIP DA CONSEGNARE:</u>	CognomeNome-matricola.zip (es. RossiMario-0000123456.zip)

Startkit



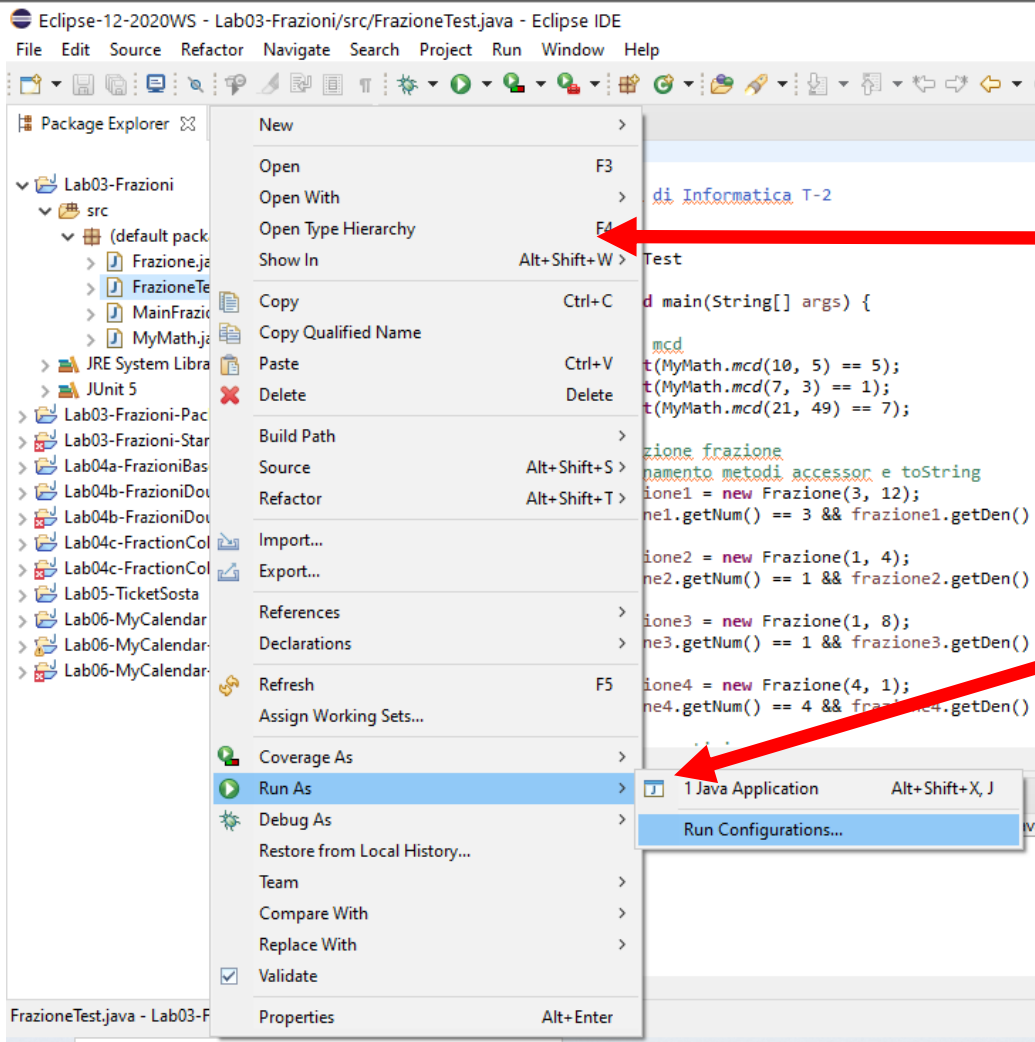
Q: È normale avere le classi con le «X rosse» sopra nello Startkit?



A: Sì è normale, la classe **Frazione** non ha tutti i metodi che dovrebbe avere, mancano quelli che dovete inserire voi oggi

Attenzione: non aspettate di aver scritto tutti i metodi di **Frazione** per collaudare la classe → collaudate metodo per metodo così non vi perderete in «errori in cascata» → basta commentare le parti dei test che non servono e poi man a mano che implementate i metodi togliere i commenti ai rispettivi test

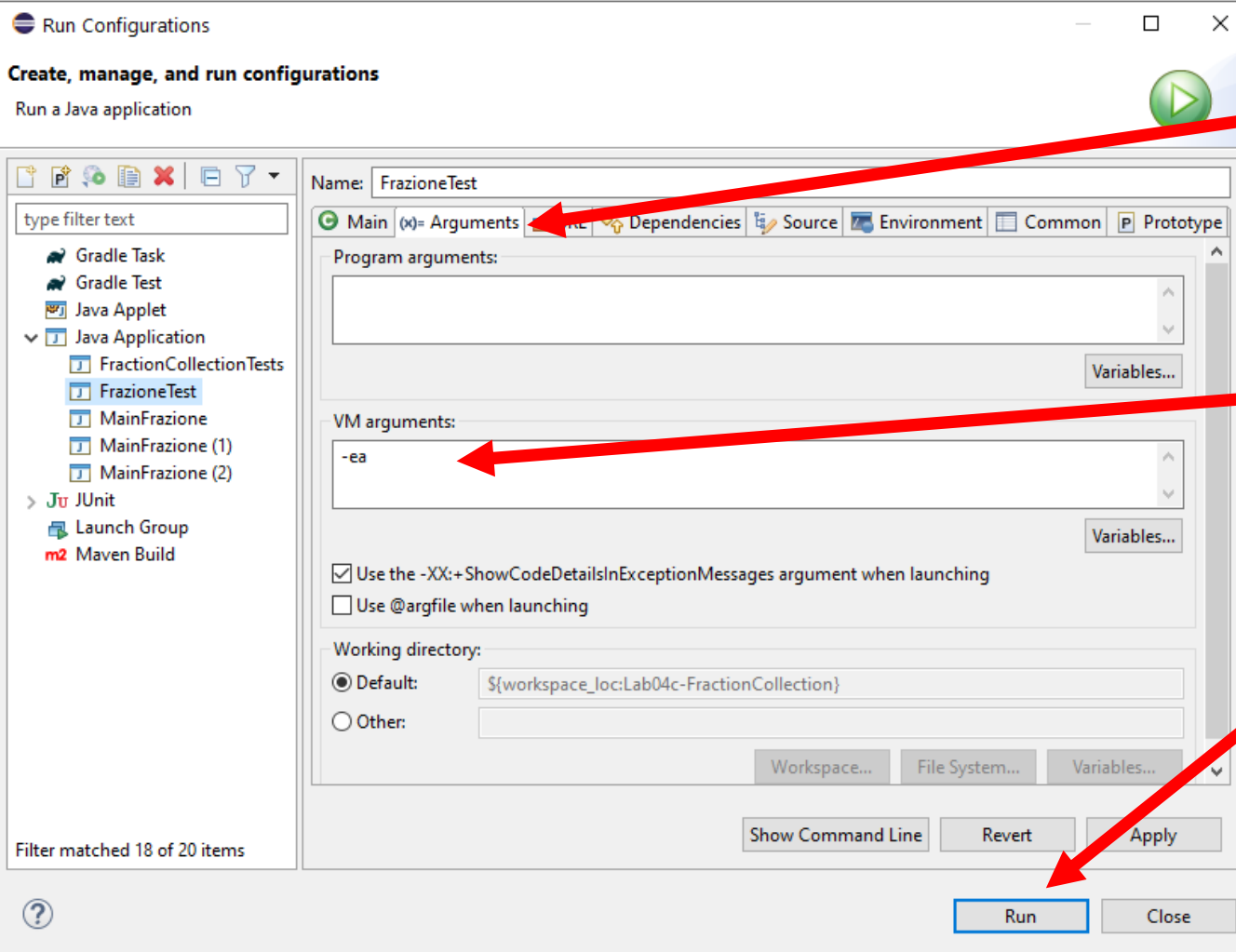
Run dei Test in Eclipse



1. Tasto destro sulla classe di test

2. Run As e successivamente Run Configurations..

Run dei Test in Eclipse



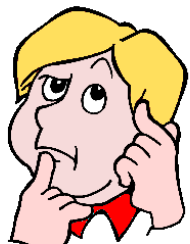
3. Menù Arguments

4. Direttiva -ea
alla VM

5. Run

Trattamento Degli Errori

- L'attuale *contratto di utilizzo* prevede che colui che crea la frazione sia *responsabile del risultato*: creare una frazione con zero al denominatore è possibile... ma prima o poi scoppia tutto!
- Il bug è evidentemente a MONTE della costruzione, ma se ne subiscono le conseguenze a VALLE, quando si usa l'oggetto
- IDEALMENTE, bisognerebbe poter *interrompere l'operazione* e *notificare al chiamante* il problema, ma (per ora!) non sappiamo farlo
- In alternativa, ci vorrebbe qualcuno che almeno verificasse i parametri PRIMA della costruzione....



Idee?