# Fondamenti di Informatica T2
# Frazioni – Prima 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 prima versione di un componente **Frazione**
  - come **tipo di dato astratto (ADT)**
  - progettato come tipo “valore” (oggetto immutabile: una volta “nato” il suo valore è **costante**)
- … usiamo ancora la linea di comando per questa volta….

# Frazione: versione base

- **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***)

# Analisi Requisiti: *Costruttore*

- Come rappresentare internamente una frazione **negativa**?
  - Il segno è mantenuto dal numeratore, dal denominatore, da entrambi o da una terza entità?
  - SCELTA: manteniamo le cose semplici → dal solo numeratore
- Relativamente ai **dati in ingresso** al **costruttore**:
  - vanno *normalizzati* in modo che il segno sia mantenuto dal solo numeratore, come stabilito
  - Quanti e quali costruttori?

# Analisi Requisiti: *Costruttore*

**Frazione.java**

```
public class Frazione {
  private int num, den;

  public Frazione(int n, int d){

 // normalizzo num e den per il segno…
       num = n; den = d;
  }
  // segue il resto della classe…
}
```

Costruttore PRIMARIO a due argomenti

È il costruttore principale, che gestisce il caso generale.

- Caso particolare: i valori interi si possono esprimere come *particolari frazioni* con denominatore 1.

```
f8 = new Frazione(8,1);
```

- Se questa esigenza è frequente, si può sfruttare la possibilità di definire più costruttori per predisporre un *costruttore ausiliario* che gestisca questo caso particolare.

# Analisi Requisiti: *Costruttore*

**Frazione.java**

```
public class Frazione {
  private int num, den;

  public Frazione(int n, int d){
   // normalizzo num e den per il segno…

    num = n; den = d;
  }
  public Frazione(int n){
        num = n; den = 1;
  }
  // segue il resto della classe…
}
```

Costruttore PRIMARIO a due argomenti

Costruttore AUSILIARIO a un solo argomento

Non indispensabile, ma utile per gestire il caso particolare.

- Grazie al costruttore ausiliario, le frazioni che esprimono valori interi si possono esprimere più efficacemente:

```
f8 = new Frazione(8);
```

# Analisi Requisiti: *Accessor*

- Una volta che i valori sono “dentro” l’entità, occorre poterli “vedere” da fuori
  - Servono due ***accessor*** pubblici (**`get*`**) che consentano di recuperare numeratore (con segno) e denominatore

- NON servono invece i metodi **`set*`** perché Frazione è un oggetto immutabile e quindi non può essere cambiato dall’esterno

# Analisi Requisiti: *Equals*

- Test di uguaglianza/equivalenza (metodo **`equals`**):
  - consente di esprimere un concetto di uguaglianza *personalizzato*
  - prende in ingresso *un'altra frazione* e verifica se sia *uguale* a quella su cui è stato invocato il metodo
  - nel caso specifico ha senso che "uguale" significhi "equivalente"
- Due generiche frazioni **n / m** e **p / q** sono **equivalenti** se vale la seguente uguaglianza: **n * q = m * p**.

  ESEMPIO: 1/2 è equivalente a 3/6 o a 9/18 e a molte altre

# Analisi Requisiti: *MinTerm*

- Riduzione ai minimi termini (metodo **`minTerm`**)
  - restituisce anch'essa una *nuova* frazione senza alterare quella su cui è stato invocato il metodo
  - si effettua dividendo numeratore e denominatore per il loro *Massimo Comun Divisore (M.C.D.)*

    ESEMPIO: mcd(14,8) = 2 → minTerms(14/8) = 7/4
  - OCCHIO: i valori negativi come si gestiscono..?

Sfruttare libreria **MyMath** realizzata nella scorsa esercitazione, in particolare metodo statico **mcd**

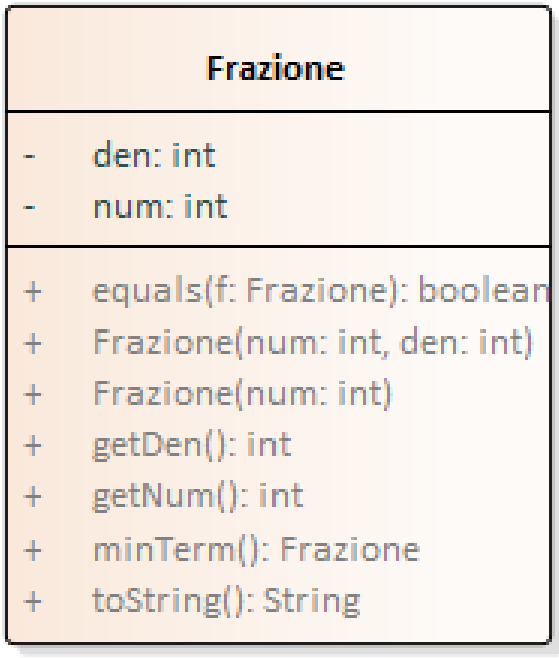
**ATTENZIONE: mcd è calcolato con l'algoritmo di Euclide e prevede che i dati in ingresso siano numeri NATURALI (no numeri negativi e attenzione che il numeratore non sia 0 –> rischio di divisione per 0)**

# Analisi Requisiti: *toString*

- Il metodo toString permette di ottenere una rappresentazione della **`Frazione`** sotto forma di stringa che permette di stampare su console la **`Frazione`**

- La  stringa di uscita dovrebbe avere la forma Num/Den

# Analisi Requisiti: *il Modello*




Rappresentazione dell'ADT Frazione attraverso il linguaggio di modellazione UML

# Collaudo di Frazione

- Per collaudare l'ADT **`Frazione`** vi forniamo noi del codice pronto in un opportuno «pacchetto» chiamato Lab02-Frazione-PrimaParte-SK
- Nel pacchetto troverete:
  - la classe **`MainFrazione`** che contiene un main di prova
  - la classe **`FrazioneTest`** che contiene il codice per il collaudo della classe **`Frazione`**
- **Non troverete** MyMath
  → dovete usare la vostra libreria sviluppata nell'esercitazione precedente

Ricordare di aggiungere la direttiva **–ea**

# Esempio di Run

Prompt fondamenti

```
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab>cd Lab02-Frazione-PrimaParte

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>javac MyMath.java

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>javac MyMath.java

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>javac Frazione.java


J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>javac MainFrazione.
java

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>java MainFrazione
Creata la frazione 3/12
Creata la frazione 1/4
Creata la frazione 1/8
Creata la frazione 4
Creata la frazione -1/8
Creata la frazione -2/3
Creata la frazione 5/7

Le frazioni 3/12 e 1/4 sono equivalenti
Le frazioni 3/12 e 1/8 non sono equivalenti
Le frazioni 3/12 e -2/3 non sono equivalenti

La frazione 3/12 ridotta ai minimi termini diventa 1/4

La frazione -2/3 ridotta ai minimi termini diventa -2/3


J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab02-Frazione-PrimaParte>
```