

SUPSI

Movie Stat

Studente/i

Michea Colautti
Julian Cummaudo

Corso di laurea

Ingegneria informatica

Modulo / Codice Progetto

Ingegneria Software

Anno

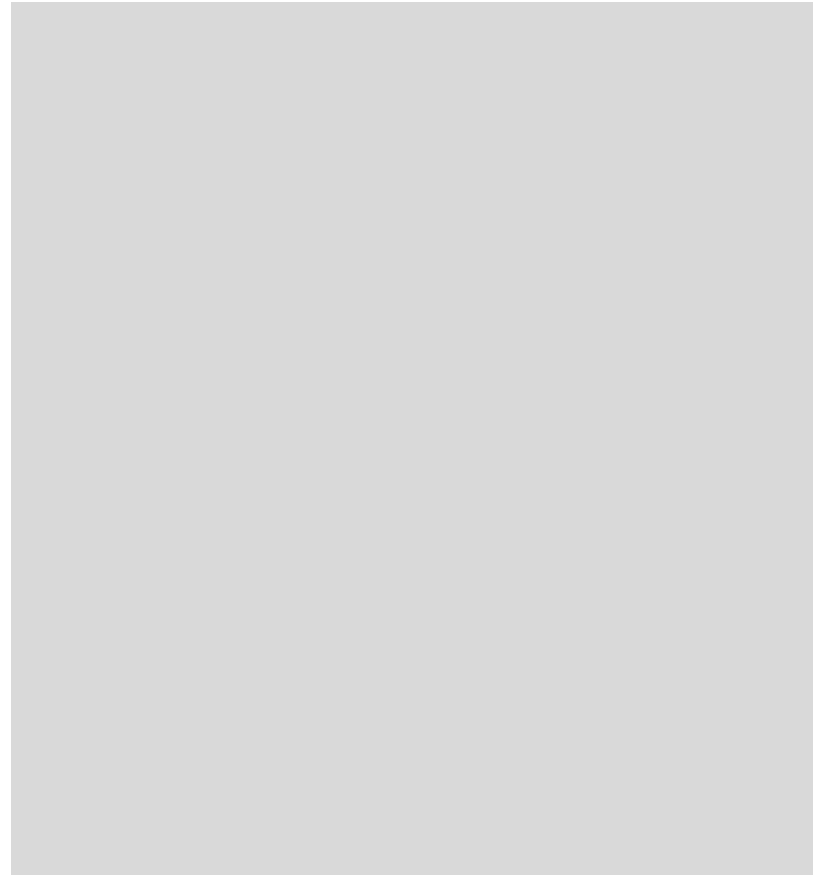
2025

Data

17.03.2025

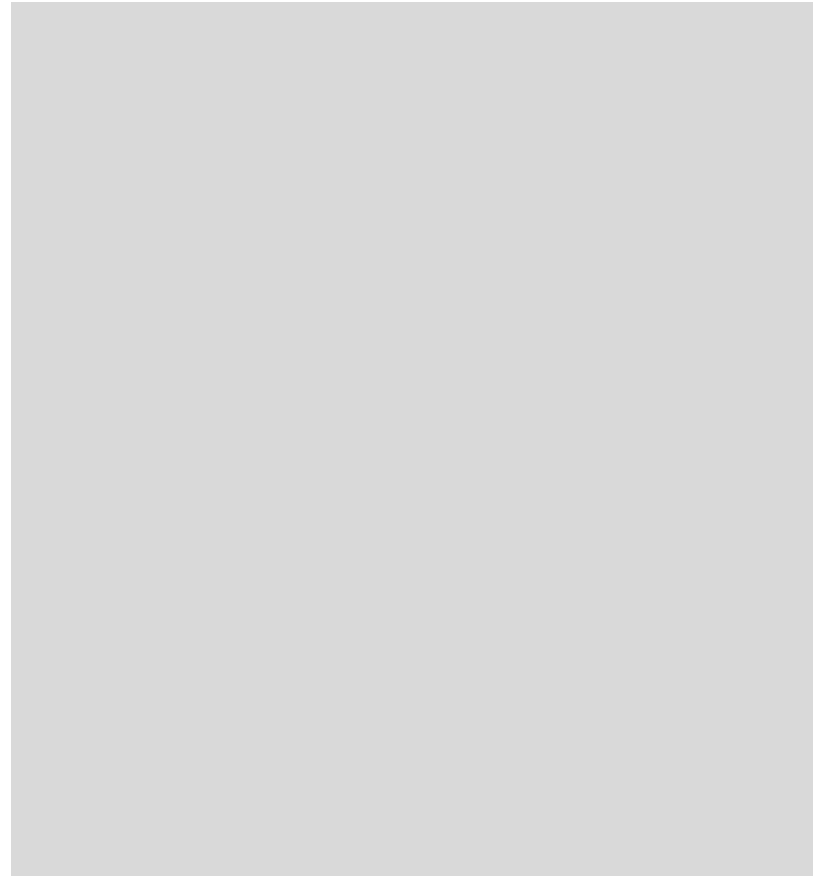
Indice

- **Informazioni generali**
- **Struttura dei file**
- **Funzioni principali**
- **Conclusioni**



Informazioni generali

- **Sviluppo su IntelliJIdea**
- **Utilizzo di Git**
- **Divisione delle funzioni**
- **Revisione assieme**



Struttura dei file

- **CsvTool**
 - Leggere e scrivere un file CSV
- **Movie**
 - Classe che rappresenta un Film
- **MovieStat**
 - Avendo una lista di Film, calcolare statistiche
- **Test**
 - Classe con metodo Main

```
OPEN CSVReader with inputFile  
READ all records from CSV into a list
```

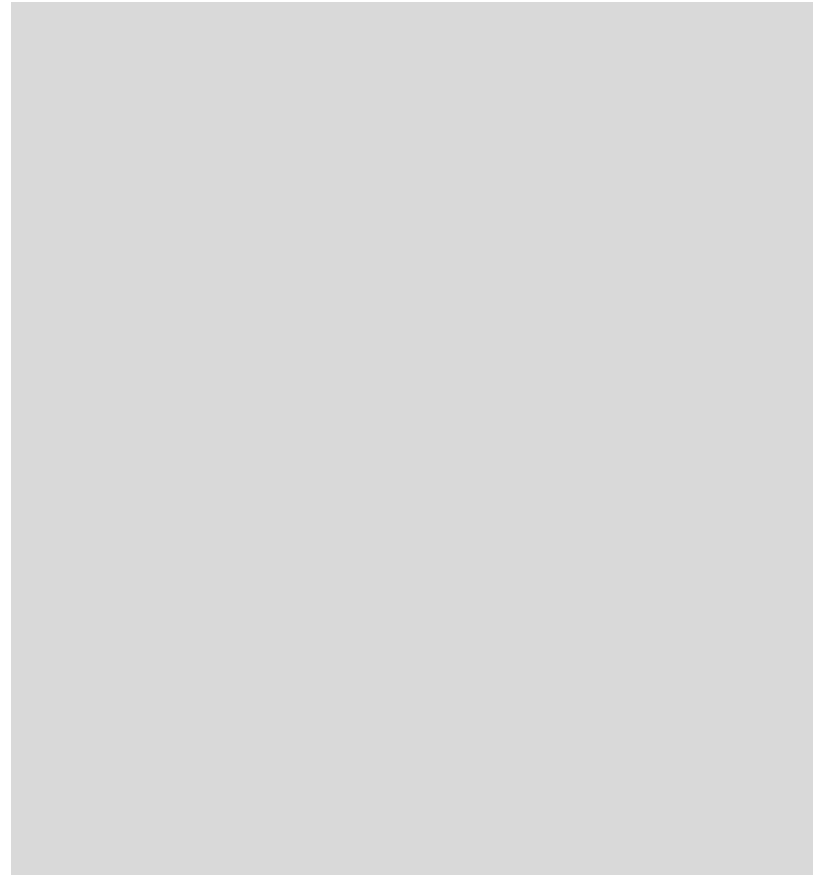
```
FOR each record (skipping header)
```

```
    SET row to current record  
    IF row has less than 14 columns  
        CONTINUE to next iteration  
    ELSE  
        EXTRACT values from row:  
        PARSE values  
        CREATE new Movie with values  
        ADD movie to movies list
```

```
RETURN movies
```

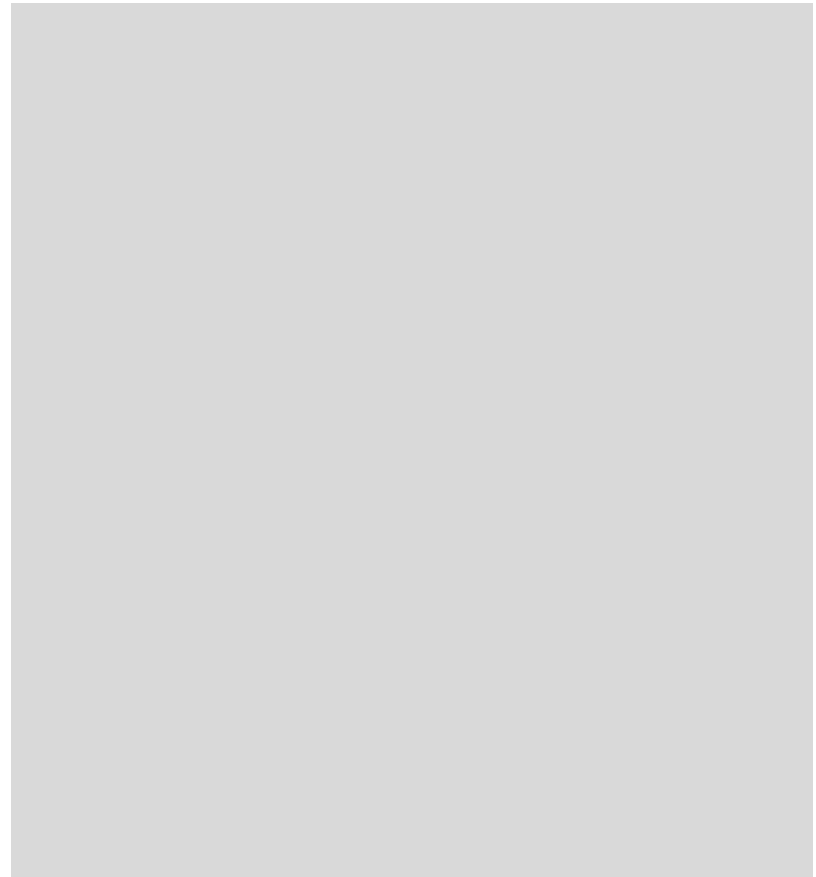
Leggere un file CSV

- **Utilizzo della classe CSVReader**
 - Classe testata e stabile
- **Percorrere il file**
- **Uso di FileReader**
- **Estrarre i dati e creare istanza di Movie**



Scrivere un file CSV

- Aprire il file di config con `FileInputStream`
- Ricavare le proprietà, salvare la path del file nello stato di `CSVTool`
- `CSVWriter` scrive le intestazioni e i dati da degli array



Creazione delle statistiche

- **Metodo compute**

```
public void compute() {  
    int moviesNumber = getMoviesNumber();  
    double avgRuntime = getAvgRuntime();  
    String bestDirector = getBestDirector();  
    String mostPresentS = getMostPresentStar();  
    int mostProdYear = getMostProdYear();  
    csvTool.writeCsv(moviesNumber, avgRuntime, bestDirector, mostPresentS, mostProdYear);  
}
```

- **Utilizzo di Stream**

```
public double getAvgRuntime() {  
    return movieList.stream() Stream<Movie>  
        .mapToDouble(Movie::getDuration) DoubleStream  
        .average() OptionalDouble  
        .orElse( other: 0.0);  
}
```

Conclusioni

- **Progetto semplice ma interessante**
- **Discussioni settimanali**

