

# Lezione 7 - Prima Parte

Federico Zanardo

3-12 Giugno 2024

## Contents

<b>1</b>	<b>Input/Output e File</b>	<b>2</b>
1.1	Copiare contenuto da un file a un altro . . . . .	2
1.2	Analisi del log di sistema . . . . .	2
1.3	Gestione di un inventario . . . . .	2
1.4	Statistiche dei voti degli studenti . . . . .	2
1.5	Esercizio bonus Lezione 4 . . . . .	3

# 1 Input/Output e File

## 1.1 Copiare contenuto da un file a un altro

Scrivere un programma che:

- Legga il contenuto di un file di testo chiamato `source.txt`
- Scriva il contenuto letto in un nuovo file di testo chiamato `destination.txt`

Il contenuto del file di input è costituito da stringhe a piacere.

## 1.2 Analisi del log di sistema

Scrivere un programma che:

- Legga un file di log chiamato `system.log`
- Analizzi il file per trovare tutte le occorrenze di errori (linee che contengono la parola `ERROR`)
- Scriva tutte le linee contenenti errori in un nuovo file chiamato `error.log`

Suggerimento: utilizzare `BufferedReader` per leggere il file di log e `BufferedWriter` per scrivere il file di errori.

## 1.3 Gestione di un inventario

Scrivere un programma che:

- Legga un file di testo chiamato `inventory.txt` che contiene informazioni su un inventario (nome del prodotto, quantità, prezzo per unità, i.e., `Orange,15,0.30`)
- Permetta all'utente di eseguire operazioni sull'inventario:
  - Aggiungere un nuovo prodotto
  - Aggiornare la quantità di un prodotto esistente
  - Rimuovere un prodotto
  - Salvi l'inventario aggiornato in un file chiamato `inventory_updated.txt`

## 1.4 Statistiche dei voti degli studenti

Scrivere un programma che:

- Legga un file di testo chiamato `grades.txt` contenente i voti degli studenti (nome, voto)
- Calcoli la media, il voto massimo e il voto minimo
- Scriva i risultati in un file chiamato `statistics.txt`

## 1.5 Esercizio bonus Lezione 4

Aggiungere delle API nel controller che permettano di aggiungere un nuovo utente, un nuovo prodotto, e un nuovo ordine. Rispettando l'architettura proposta nell'esercizio 4, i rispettivi oggetti dovranno essere salvati in un file dedicato. Quindi, vi saranno un file per gli utenti (`users.txt`), per i prodotti (`products.txt`) e per gli ordini (`orders.txt`). Il metodo per l'aggiunta di un nuovo ordine deve seguire il seguente esempio:

```
public Order addOrder(int userId, int productId) {
    // metodo per cercare un utente dato il suo identificativo
    User user = ...

    // metodo per cercare un prodotto dato il suo identificativo
    Product product = ...

    Order order = new Order(/* valorizzare correttamente il costruttore */);

    return order;
}
```