

Esercitazione Java: Lettura di un file CSV e Generazione di una Tabella HTML

L'obiettivo è creare un programma Java che legge un file `temperature.csv` contenente dati relativi alle temperature nelle principali città europee e genera un file HTML contenente una tabella per visualizzare queste informazioni.

Requisiti

1. Il file CSV contiene le seguenti colonne:

- **Data:** la data in formato `yyyy-MM-dd`.
- **Città:** il nome della città.
- **Stato:** il paese di appartenenza.
- **Temperatura Minima:** la temperatura minima registrata.
- **Temperatura Massima:** la temperatura massima registrata.

2. Il programma deve:

- Leggere i dati dal file CSV.
 - Generare un file HTML con una tabella che mostri i dati del CSV.
 - Applicare uno stile di base alla tabella HTML.
-

Struttura del file CSV di esempio (`temperature.csv`)

```
Data,Città,Stato,Temperatura Minima,Temperatura Massima
2024-11-10,Roma,Italia,10,18
2024-11-10,Parigi,Francia,7,15
2024-11-10,Berlino,Germania,5,12
2024-11-10,Madrid,Spagna,8,20
2024-11-10,Londra,Regno Unito,6,14
```

Codice Java

Ecco un esempio di implementazione:

```
import java.io.*;
import java.nio.file.*;
import java.util.*;

public class TemperatureCSVtoHTML {

    public static void main(String[] args) {
        String csvFilePath = "temperature.csv"; // Percorso del file CSV
        String htmlFilePath = "temperature.html"; // Percorso del file
        HTML generato
```

```

// Legge i dati dal CSV
List<String[]> data = readCSV(csvFilePath);

// Genera il file HTML
generateHTML(data, htmlFilePath);

System.out.println("File HTML generato con successo: " +
htmlFilePath);
}

// Metodo per leggere i dati dal file CSV
public static List<String[]> readCSV(String filePath) {
    List<String[]> data = new ArrayList<>();
    try (BufferedReader br =
Files.newBufferedReader(Paths.get(filePath))) {
        String line;
        while ((line = br.readLine()) != null) {
            String[] row = line.split(","); // Divide i dati per
colonna
            data.add(row);
        }
    } catch (IOException e) {
        System.err.println("Errore nella lettura del file CSV: " +
e.getMessage());
    }
    return data;
}

// Metodo per generare il file HTML
public static void generateHTML(List<String[]> data, String filePath)
{
    try (BufferedWriter bw =
Files.newBufferedWriter(Paths.get(filePath))) {
        // Scrive l'intestazione dell'HTML
        bw.write("<!DOCTYPE html>\n");
        bw.write("<html lang=\"en\">\n");
        bw.write("<head>\n");
        bw.write("<meta charset=\"UTF-8\">\n");
        bw.write("<meta name=\"viewport\" content=\"width=device-
width, initial-scale=1.0\">\n");
        bw.write("<title>Temperature Città Europee</title>\n");
        bw.write("<style>\n");
        bw.write("table { border-collapse: collapse; width: 100%;
}\n");
        bw.write("th, td { border: 1px solid #ddd; padding: 8px; text-
align: left; }\n");
        bw.write("th { background-color: #f4f4f4; }\n");
        bw.write("</style>\n");
        bw.write("</head>\n");
        bw.write("<body>\n");
        bw.write("<h1>Temperature nelle principali città
europee</h1>\n");
        bw.write("<table>\n");

```

```

        // Scrive la riga di intestazione
        if (!data.isEmpty()) {
            bw.write("<tr>\n");
            for (String header : data.get(0)) {
                bw.write("<th>" + header + "</th>\n");
            }
            bw.write("</tr>\n");
        }

        // Scrive i dati
        for (int i = 1; i < data.size(); i++) { // Ignora la riga di
intestazione
            bw.write("<tr>\n");
            for (String cell : data.get(i)) {
                bw.write("<td>" + cell + "</td>\n");
            }
            bw.write("</tr>\n");
        }

        // Chiude la tabella e il file HTML
        bw.write("</table>\n");
        bw.write("</body>\n");
        bw.write("</html>");
    } catch (IOException e) {
        System.err.println("Errore nella scrittura del file HTML: " +
e.getMessage());
    }
}
}

```

Esecuzione del programma

1. Crea il file CSV (**temperature.csv**) nello stesso percorso del programma, oppure specifica il percorso esatto nel codice.
2. Compila ed esegui il programma.
3. Troverai il file HTML generato (**temperature.html**) nello stesso percorso.

Output del file HTML

Il file HTML generato avrà una struttura simile a questa:

Tabella HTML

Data	Città	Stato	Temperatura Minima	Temperatura Massima
2024-11-10	Roma	Italia	10	18
2024-11-10	Parigi	Francia	7	15

Data	Città	Stato	Temperatura Minima	Temperatura Massima
2024-11-10	Berlino	Germania	5	12
2024-11-10	Madrid	Spagna	8	20
2024-11-10	Londra	Regno Unito	6	14

Estensioni possibili

- 1. **Validazione dei dati:** Controllare che le righe nel CSV siano corrette e gestire eventuali errori.
- 2. **Stile avanzato:** Migliorare lo stile della tabella con CSS.
- 3. **Aggiunta di funzionalità:** Ordinare i dati nel file HTML o aggiungere statistiche (es. temperatura media).