

## Esercizio 20: Esercitazione periodo di chiusura per emergenza

### Obbiettivi:

- 1) a partire dai siti di ARPA Piemonte o Torino Respira, cercare degli "open data" riguardo a qualità dell'aria o su argomenti che riguardano i cambiamenti climatici o l'ambiente in generale. Cercare i dati in un formato che sia csv oppure json o xml. Comprendere bene il significato dei dati e il loro formato.
- 2) Progettare e scrivere un programma java che, tramite interfaccia GUI legga il file con i dati, li elabori e visualizzi in una forma grafica i dati stessi, in modo da poterli interpretare e poi discutere. La lettura da file può essere utilizzata per inserire i vari dati in un vettore (in modalità pila o coda a vostra scelta). Potete usare sia la classe Vector che ArrayList a scelta.

Indice:

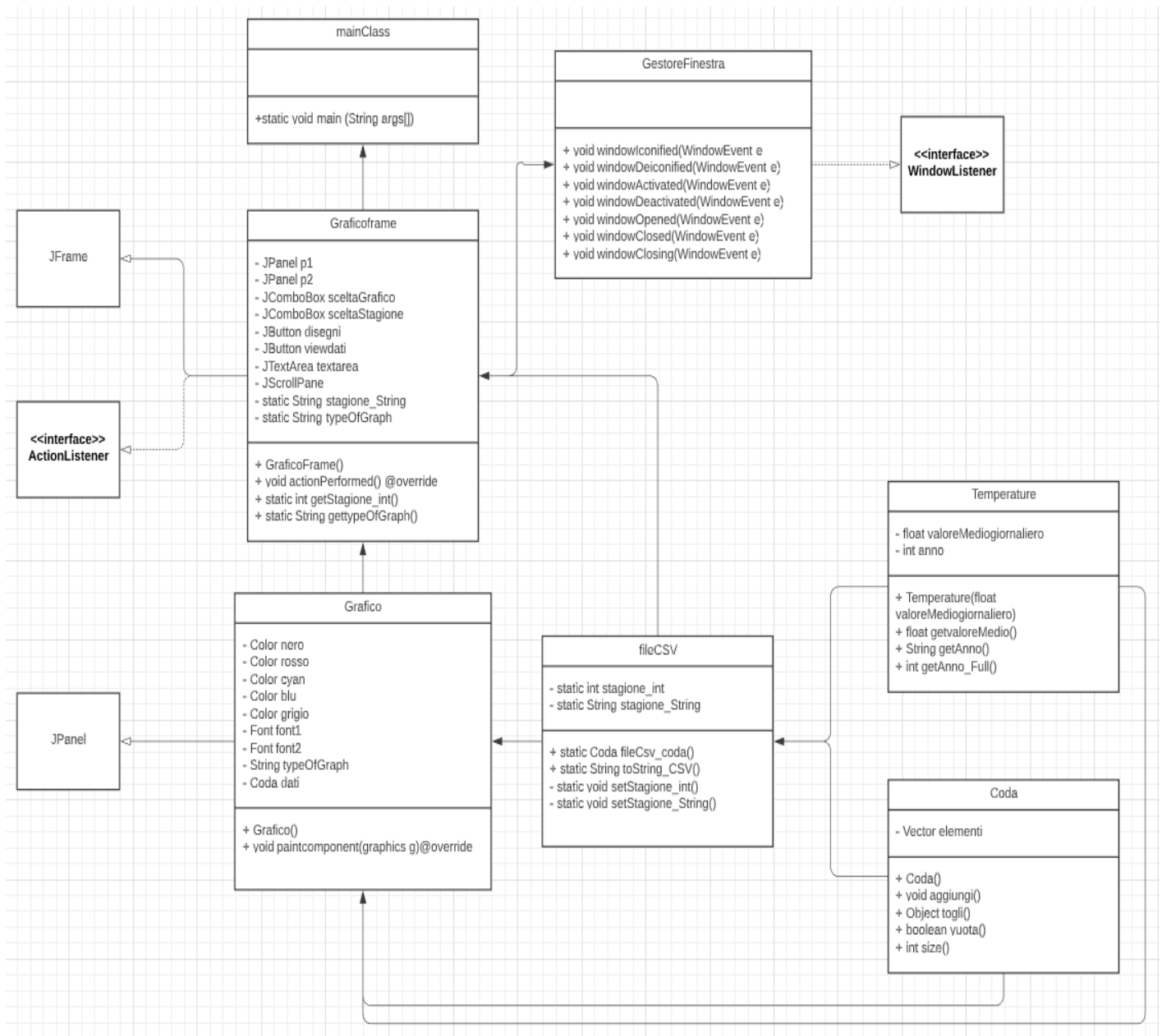
uml diagram pag.3

open data pag.4

il programma pag.5

lettura file csv pag.6

UML diagram:



Open data:

abbiamo ricavato l' open data dal sito Arpapiemonte nel link :

[https://www.arpa.piemonte.it/rischinaturali/accesso-ai-dati/opendata/elenco\\_clima.html?delta=1](https://www.arpa.piemonte.it/rischinaturali/accesso-ai-dati/opendata/elenco_clima.html?delta=1)

il file csv contiene le temperature minime e massime di ogni stagione dal 1991 al 2017. purtroppo i valori delle temperature massime risultavano uguali a quelle minime quindi abbiamo optato per utilizzare solo le temperature minime nel programma.

Media delle temperature massime giornaliere ºC						
anno	inverno	primavera	estate	autunno		stagione invernale
1991	-1.7	7.4	18.2	9.3		1990-01
1992	0.6	8.6	17.3	9.1		1991-92
1993	0.4	7.8	17.3	8.9		1992-93
1994	1.2	9.2	18.5	9.5		1993-94
1995	0.8	7.9	17.1	8.6		1994-95
1996	1	7.9	17.1	8.5		1995-96
1997	1.5	8.7	17.1	9.7		1996-97
1998	0.4	7.8	18.1	7.9		1997-98
1999	-0.1	8.7	17.4	9.5		1998-99
2000	1.3	8.9	17	9.8		1999-00
2001	0.1	9.1	17.7	9		2000-01
2002	1	8.6	17.3	9.4		2001-02
2003	-0.2	8.4	20.5	8.5		2002-03
2004	0.5	7.4	17.7	10.1		2003-04
2005	-1.6	8.2	17.3	9.6		2004-05
2006	-0.3	8	17.6	10.7		2005-06
2007	1.5	9.8	17.1	8		2006-07
2008	0.9	8.3	17.2	9.2		2007-08
2009	-1.2	8.8	17.7	9.5		2008-09
2010	-1.2	7.2	17	8.4		2009-00
2011	0.1	8.6	16.5	8.9		2010-11
2012	-2.2	8	17.5	9.1		2011-12
2013	-0.9	6.8	16	8.9		2012-13
2014	1.7	7.3	15.7	10.4		2013-14
2015	0.1	7.7	18	8.2		2014-15
2016	0.3	7.9	17.1	9.6		2015-16
2017	0.4	8.8	18.2			2016-17

Il programma:

il programma presenta un frame “centrale”(immagine 1) in cui nel primo panel è possibile scegliere tramite dei JComboBox il tipo di grafico (istogramma,lineare) e la stagione di cui si vogliono sapere le temperature.

Nel secondo panel ci sono 2 Jbutton che permettono :

-la creazione di un secondo frame contenente il grafico.(immagine 2, 3)

-la visualizzazione dei dati del file nel textarea presente nel terzo panel (JscrollPane)(immagine 4) .

Immagine 1:



immagine 2,3:

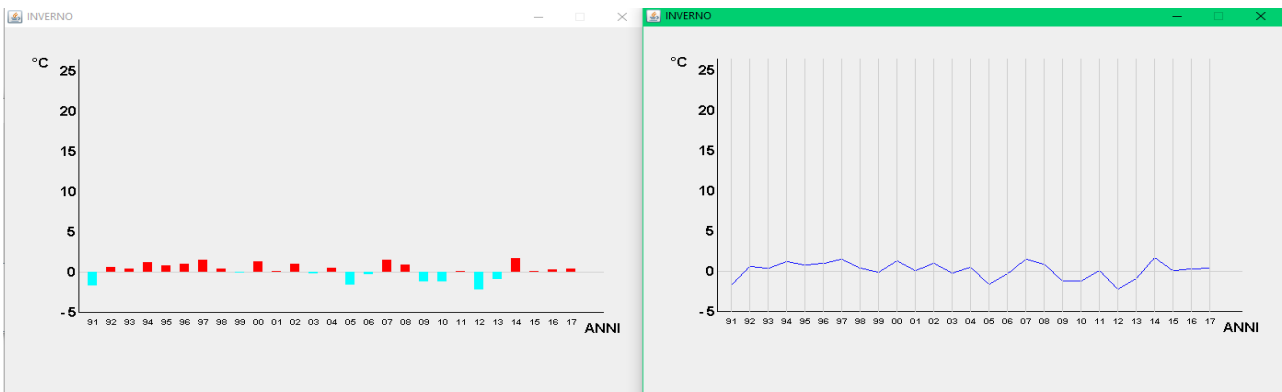
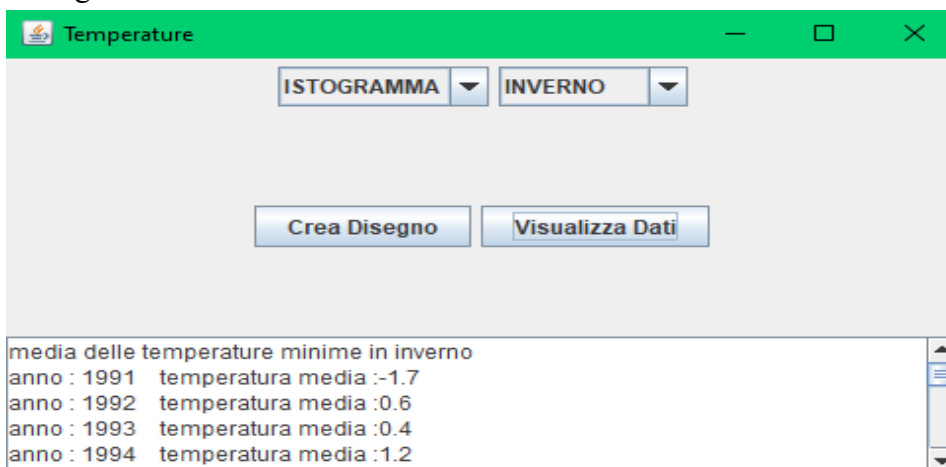


immagine 4:



lettura del file csv:

```
try (BufferedReader br = new BufferedReader(new FileReader(csvFile))) {  
    while ((line = br.readLine()) != null) {  
        lineCheck = line.split(csvSplitBy)[0];  
        if(lineCheck.substring(0,2).equals("19") || lineCheck.substring(0,2).equals("20"))  
        {  
            // use comma as separator  
            valoreMedioGiornaliero = Float.parseFloat(line.split(csvSplitBy)[stagione_int]);  
            anno = Integer.parseInt(line.split(csvSplitBy)[0]);  
            Temperature Dati = new Temperature(valoreMedioGiornaliero,anno);  
            insiemeDati.aggiungi(Dati);  
        }  
    }  
}
```

per leggere da file csv abbiamo usato il try-with resources.

La risorsa è l'elemento all'interno delle parentesi tonde dopo il try.

Il try-with resources permette di chiudere la risorsa automaticamente alla fine del try-catch.

I dati dal file vengono presi tramite il comando .split() sulla linea del file, il comando split viene usata su una stringa e la divide in un array di stringhe con i caratteri della stringa originale divisi dal parametro di split:

esempio :

```
String stringa = "ciao;hello;hola";  
String[] array = stringa.split(";");  
System.out.println(array[0]);  
System.out.println(array[2]);  
Output : ciao  
        hola
```

in questo modo è possibile prendere i dati di stagioni diverse con l'attributo statico stagione\_int che viene preso dal metodo statico getStagione\_int della classe GraficoFrame.