

Esame Laboratorio di Programmazione I

22/07/2025

Svolgimento

Chiamate il file in cui scrivere il vostro codice `esame.matricola.py` (la matricola è SMnumeri o ECnumeri). All'inizio del file scrivete un commento con il vostro nome e numero di matricola. Consegnate un codice che compila, se il codice non compila non sarà valutato!!! Commentate bene il codice descrivendo quello che fate! Hint: fate un pezzo alla volta e testate il vostro codice per vedere che funziona nel modo corretto.

Dataset Il file `Temperatures.csv` contiene la serie temporale delle temperature medie mensili globali, registrate a partire da Gennaio 1750 a Dicembre 2015. Le colonne principali sono:

- `dt`: data in formato DD/MM/YYYY,
- `LandAverageTemperature`: temperatura media mensile,
- `LandAverageTemperatureUncertainty`: incertezza associata alla temperatura.

1 (10 punti) Lettura e filtraggio dei dati

Definire la classe **CSVTimeSeriesFile**:

- (2 punti) La classe deve essere istanziata con il nome del file tramite la variabile `name`.
- (8 punti) Deve avere un metodo `get_data()` che torni una lista di liste, dove il primo elemento delle liste annidate è la data (sotto forma di stringa), ed il secondo è la temperatura media mensile (sotto forma di float). Il valore della temperatura deve essere salvato se è solo se la variazione associata a quel valore, `LandAverageTemperatureUncertainty`, è minore di 5 gradi.

Esempio d'uso:

```
time_series_file = CSVTimeSeriesFile(name="GlobalTemperatures.csv")
data = time_series_file.get_data()
```

2 (10 punti) Analisi della variabilità annuale

Definire la funzione `compute_month_variation(time_series, first_year, second_year)` che calcoli la variazione nel valore della temperatura tra lo stesso mese, tra il `second_year` ed il `first_year`.

- (3 punti) Raggruppare le temperature per anno, per i due anni considerati,
- (3 punti) Considerare solo i mesi presenti in entrambi gli anni,
- (2 punti) Fare la differenze mensili tra i mesi,
- (2 punti) Ritornare un dizionario con chiave i mesi e valore la variazione per quel mese.

Prendiamo ad esempio: `compute_variation(time_series, 1900, 2000)`

Esempio output:

```
{
    1: val_mese_1_2000 - val_mese_1_1900,
    2: val_mese_2_2000 - val_mese_2_1900,
    ...
}
```

N.B. La chiave è di tipo `int` ed il valore è di tipo `float`.

3 (10 punti) Eccezioni e gestione input

Tutte le eccezioni devono essere istanze della classe seguente:

```
class ExamException(Exception):
    pass
```

Gestire i seguenti casi:

- (2 punti) La classe `CSVTimeSeriesFile` deve controllare l'esistenza del file nell'`__init__()` (non in `get_data`) e, nel caso il file non esista o non sia leggibile, alza un'eccezione: `"Errore: impossibile aprire il file"`
- (2 punti) Stampare un messaggio quando il mese viene saltato perché l'incertezza è ≥ 5 : `"Data saltata perché valore troppo incerto"`
- (2 punti) I due anni inseriti devono essere due numeri interi con il primo strettamente minore del secondo. Se non sono inseriti degli interi sollevare un'eccezione: `"Errore: gli anni inseriti devono essere di tipo intero."`. Se il secondo anno è minore o uguale al primo sollevare un'eccezione: `"Errore: il secondo anno deve essere maggiore del primo."`
- (2 punti) Se un mese non è presente in uno o entrambi gli anni, stampare a schermo: `"La variazione per il mese X non può essere calcolata"`. X sarà il valore del mese non presente.
- (2 punti) Se nessun mese è disponibile, alzare un'eccezione: `"Gli anni considerati non hanno mesi validi"`