

CLIMATE MONITORING

Manuale Tecnico

Iuri Antico *matricola:* 753144

Michael Bernasconi *matricola:* 752259

Gabriele Borgia *matricola:* 753262

Beatrice Balzarini *matricola:* 752257

19 maggio 2023

Indice

1	Introduzione	5
1.1	Librerie esterne utilizzate	5
1.1.1	Apache Commons CSV	5
2	Struttura generale del sistema di classi	6
2.1	cli	6
2.1.1	Help	6
2.1.2	Login	6
2.1.3	MainMenu	6
2.1.4	MostraMisurazioni	6
2.1.5	Registrazione	6
2.2	gestori	6
2.2.1	DataBase	6
2.2.2	Gestore	6
2.2.3	GestoreArea	7
2.2.4	GestoreCentro	7
2.2.5	GestoreDato	7
2.2.6	GestoreMisurazioni	8
2.2.7	GestoreOperatore	8
2.3	Magazzino	8
2.3.1	AreaGeografica	9
2.3.2	CentroMonitoraggio	9
2.3.3	DatoGeografico	9
2.3.4	Filtratore	9
2.3.5	Indirizzo	9
2.3.6	ListaAree	9
2.3.7	Misurazioni	9
2.3.8	Operatore	9
2.4	Utils	9
2.4.1	listacustom	9
2.4.1.1	CollezioniIterator	10
2.4.1.2	Nodo	10
2.4.2	result	10
2.4.2.1	Panic	10
2.4.2.2	Result	10

	2.4.2.3	Either	10
2.4.3	terminal	10
	2.4.3.1	Screen	10
	2.4.3.2	Terminal	10
	2.4.3.3	View	10
2.4.4	CercaAree	10
2.4.5	Convertible	10
2.4.6	DataTable	10
2.4.7	MediaAree	11
2.4.8	DatoGeografico	11
2.5	Main	11

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 Librerie esterne utilizzate

1.1.1 Apache Commons CSV

E' una libreria di Apache che fornisce i metodi per la gestione dei file (*.CSV). In particolare è stata usata nel progetto per la lettura e scrittura dei dati da memorizzare.

Capitolo 2

Struttura generale del sistema di classi

2.1 cli

2.1.1 Help

2.1.2 Login

2.1.3 MainMenu

2.1.4 MostraMisurazioni

2.1.5 Registrazione

2.2 gestori

All'interno del package Gestori si trovano una serie di classi finalizzate alle operazioni di lettura e scrittura su File contenenti dati utili al monitoraggio di parametri climatici sul territorio italiano. Ad ogni record memorizzato su file viene associato un indice (ID) univoco.

2.2.1 DataBase

La classe DataBase si occupa di creare (per ogni classe che estende Gestore) un oggetto in grado di richiamare le funzioni associate.

2.2.2 Gestore

La classe astratta Gestore contiene metodi relativi la gestione dei file contenenti i dati d'interesse. In particolare:

- **void close()** Metodo che si occupa di aprire un file.
- **void reload()** Metodo che si occupa di ricaricare (chiudere e riaprire un file)

- **DataTable buildObject (CSVRecord r)** Metodo astratto che si occupa di prendere in input come parametro formale un record e crea l'oggetto associato all'implementatore.
- **Result<String>getProperty(String key)** Metodo che si occupa di prendere una proprietà nel file (*.CSV.DAT) associato ad una tabella (*.CSV). Il parametro formale di questo metodo è una stringa che è la chiave per prendere la proprietà del file. Nel caso in cui l'operazione di ricerca è corretta il metodo restituisce un Result valido con la stringa della proprietà che hanno chiesto. Altrimenti nel caso in cui l'operazione non è corretta il metodo restituisce una stringa associata ad un codice di errore.
- **Result<Object>setProperty(String s, String k)** Metodo che si occupa di impostare una proprietà nel file (*.CSV.DAT) associato ad una tabella (*.CSV). Il metodo ha due parametri formali: una stringa che è la proprietà da impostare e l'altra è la chiave del file. Nel caso in cui l'operazione non è corretta il metodo restituisce un Result di Object come errore.

2.2.3 GestoreArea

la classe GestoreArea estende la classe Gestore e ne eredita tutti i metodi implementa l'interfaccia CercaAree e contiene i seguenti metodi:

- **Result<AreaGeografica>getArea(long geoID)** Metodo che ricerca una determinata area geografica in base al suo ID. In questo metodo c'è un solo parametro formale che il numero di geoID. Nel caso in cui l'area esiste il metodo restituisce un Result di AreaGeografica sennò restituisce una stringa di errore.

2.2.4 GestoreCentro

la classe GestoreCentro estende la classe Gestore, eredita tutti i metodi e contiene i seguenti:

- **Result<CentroMonitoraggio>getCentro(String nome)** Metodo che ricerca un determinato centro di monitoraggio in base al nome e restituisce il centro di monitoraggio corrispondente al nome di tipo String fornito come parametro formale.
- **boolean addCentro(CentroMonitoraggio cm)** Metodo che crea un nuovo record relativo a un determinato centro di monitoraggio cioè il parametro formale del metodo e lo memorizza nel file CentriMonitoraggio (*.CSV). Il metodo restituisce una variabile booleana per capire se l'operazione è andata a buon fine.

2.2.5 GestoreDato

la classe GestoreDato estende la classe Gestore, eredita tutti i metodi e contiene i seguenti:

- **Result <DatoGeografico >getDato(long rid)** Metodo che ricerca un determinato dato geografico in base al suo ID che è il parametro formale del metodo. Il metodo restituisce un Result di DatoGeografico se l'operazione è andata a buon fine. Invece Se l'operazione non è stata eseguita correttamente restituisce un Result di stringa per indicare l'errore.
- **Result <Object>addDato(DatoGeografico dato)** Metodo che crea un nuovo record relativo a un determinato dato geografico che è il parametro formale e lo memorizza nel file ParametriClimatici.CSV. Il metodo restituisce un Result di Object se l'operazione non è andata a buon fine.

2.2.6 GestoreMisurazioni

la classe GestoMisurazioni estende la classe Gestore, eredita tutti i metodi e contiene i seguenti:

- **Result<Object>addMisurazione(Misurazione mis)** Metodo che serve per aggiungere una Misurazione. Prende in input un parametro formale di tipo Misurazioni e restituisce un Result di Object in caso di errore.
- **Result <Filtratore>getMisurazioni()** Metodo che prende le Misurazioni e memorizza i record relativi alle misurazioni presenti nel file ParametriClimatici (*.CSV).

2.2.7 GestoreOperatore

la classe GestoreOperatore estende la classe Gestore, eredita tutti i metodi e contiene i seguenti:

- **Result<Operatore>registrazione(Operatore op, String pwd)** Metodo che permette a un operatore di registrarsi. Il metodo riceve in input come parametri formali, un oggetto di tipo Operatore e una password di tipo String. Il Metodo restituisce un Result di Operatore se l'operazione che deve svolgere il metodo è corretta. Altrimenti restituisce una stringa con il codice di errore.
- **Result <Operatore>login(String uid, String pwd)** Metodo che permette ad un operatore di effettuare il login. I parametri formali presi in input sono userid e la password. In caso che userid e password corrispondano ad un operatore creato, il metodo restituisce l'accesso dell'Operatore. Altrimenti il metodo restituisce una stringa con un codice di errore.

2.3 Magazzino

All'interno del package Magazzino si trovano una serie di classi che servono per memorizzare informazioni che vengono lette o che dovranno essere scritte su file.(*CSV).

2.3.1 AreaGeografica

la classe AreaGeografica implementa l'interfaccia DataTable, che definisce tutti i metodi e contiene i seguenti: I metodi getter sono: getGeoID(), getLatitudine(), getLongitudine(), getStato(), getDenominazione().

- **toString()** Metodo che permette di stampare l'oggetto AreaGeografica, con i campi denominazione, stato, latitudine e longitudine.
- **equals()** Metodo che permette di confrontare un oggetto qualsiasi con un oggetto di tipo AreaGeografia.

2.3.2 CentroMonitoraggio

la classe CentroMonitoraggio implementa due interfacce: Convertable e Datatable che definisce tutti i metodi e contiene i seguenti: i metodi getter sono: getNome(), getIndirizzo(), getAree().

- **toString()** Metodo che permette di stampare l'oggetto CentroMonitoraggio con i campi nome, indirizzo e le aree associate ad esso.
- **equals()** Metodo che permette di confrontare un oggetto qualsiasi con un oggetto di tipo CentroMonitoraggio.
- **toCSV()** Metodo che permette di creare una stringa nel formato (*.CSV) adoperato all'interno del programma.

2.3.3 DatoGeografico

La classe DatoGeografico implementa l'interfaccia DataTable che definisce tutti i metodi e contiene i seguenti:

2.3.4 Filtratore

2.3.5 Indirizzo

2.3.6 ListaAree

2.3.7 Misurazioni

2.3.8 Operatore

2.4 Utils

2.4.1 listacustom

Package che contiene le classi che servono da supporto alla classe ListaCustom.

2.4.1.1 CollezioneIterator

classe che implementa l'interfaccia Iterator, permette a ListaAree di svolgere l'istruzione "for-each loop".

- **E next()** Metodo che restituisce l'elemento corrente e scorre a quello successivo.
- **boolean hasNext()** Metodo che restituisce true se il nodo che esegue il metodo ha un successore, false altrimenti.

2.4.1.2 Nodo

classe che rappresenta i nodi della lista, i metodi presenti servono per la gestione degli elementi della lista.

I metodi setter : setData, setNext;

I metodi getter : getData, getNext;

2.4.2 result

2.4.2.1 Panic

2.4.2.2 Result

2.4.2.3 Either

2.4.3 terminal

2.4.3.1 Screen

2.4.3.2 Terminal

2.4.3.3 View

2.4.4 CercaAree

L'implementazione di questa interfaccia consente la ricerca di aree geografiche.

- **ListaAree cercaAreaGeografica (String denominazione, String stato)**
Metodo che ricerca per denominazione e per stato di appartenenza, restituisce una lista di aree geografiche.
- **Result<AreaGeografica>cercaAreeGeografiche(double latitudine, double longitudine)** Metodo che ricerca per coordinate geografiche (prende in input una latitudine e longitudine) e restituisce un result di tipo AreaGeografica.

2.4.5 Convertable

L'implementazione di questa interfaccia consente a un oggetto di essere convertito nel formato (*.CSV).

- **String toCsv()** Metodo che converte l'oggetto nel formato (*.CSV). Restituisce una stringa.

2.4.6 DataTable

L'implementazione di questa interfaccia permette a due record di essere confrontati.

- **boolean equals(Object obj)** Metodo che confronta l'oggetto che richiama la funzione con l'oggetto fornito come parametro, restituisce true se sono uguali false altrimenti.

2.4.7 MediaAree

La classe che implementa questa interfaccia permette di visualizzare le informazioni relative ad un'area geografica.

- **DatoGeografico visualizzaAreaGeografica (AreaGeografica area)** Metodo che restituisce un nuovo dato geografico che rappresenta un prospetto riassuntivo dei parametri climatici associati all'area geografica fornita come input.

2.4.8 DatoGeografico

Enumerativo che rappresenta il tipo di un dato geografico.

2.5 Main