Laboratorio interdisciplinare A

Climate Monitoring

Un sistema di monitoraggio di parametri climatici

CATTANEO LUCA

FICARA PAOLO  
MAURI ANDREA

# Introduzione

Climate monitoring è un progetto sviluppato nell’ambito del progetto di Laboratorio A per il corso di laurea in Informatica dell’Università degli Studi dell’Insubria.

Il progetto è stato sviluppato in Java 17, usa un’interfaccia grafica costruita con Java Swing, libreria inclusa in NetBeans apposita per lo sviluppo del design delle applicazioni desktop, ed è stato sviluppato e testato su Windows 11.

# Librerie esterne utilizzate

Durante lo sviluppo di questo progetto è stata utilizzata una libreria di terze parti in aggiunta, chiamata “codice-fiscale-java-master”, sviluppata dagli autori del progetto e inclusa nel progetto, ai fini di denominare il codice fiscale avendo tutte le generalità necessarie fornite dall’utente in fase di registrazione.

# Struttura generale del sistema di classi

Nel progetto si possono suddividere le classi in due macro-rami: classi adibite per la gestione dei dati e la rappresentazione degli oggetti in oggetto e classi impiegate alla gestione dell’interfaccia grafica.

* Gestione dei dati e rappresentazioni oggetti
  + - DatiCondivisi(classe statica)
    - FileManager
    - Forecast
    - InterestingAreas
    - MonitoringStation
    - User
* Gestione dell’interfaccia grafica
  + - AddNotes
    - CreateMonitoringStation
    - Login
    - Menu(main)
    - MenuOperatore
    - Register
    - SearchResult

Verranno ora presentate le classi adibite alla gestione dei dati dettagliamente, non verranno invece affrontate nel particolare le classi per la gestione dell’interfaccia grafica essendo che non sono espressamente richieste nei requisiti del progetto.

# Classi per la gestione dei dati

InterestingAreas

InterestingAreas è la classe che rappresenta le aree di interesse ai quali si riferisce un utente.

Un’area di interesse viene identificata dai seguenti attributi:

* Id: identificativo numerico univoco rappresentato da una String
* name: nome dell’area di interesse rappresentato da una String
* countryCode: il codice del paese dell’area di interesse rappresentato da una String
* lat: la latitudine dell’area di interesse rappresentata da una String
* lon: la longitudine dell’area di interesse rappresentata da una String

Per ogni attributo sono presenti get e set propri, inoltre è presente un metodo “contains” che verifica se il nome dell’area di interesse contiene una determinata stringa.