

Documentație: Sistem de Monitorizare a Vremii

I. Prezentarea Proiectului

Proiectul reprezintă un sistem de monitorizare a vremii, proiectat pentru a colecta, actualiza și afișa date meteorologice în timp real. Acesta este dezvoltat în Java, demonstrând utilizarea eficientă a design pattern-urilor Singleton și Observer.

- **Funcționalități**

1. **Colectarea Datelor Meteorologice:** Sistemul colectează date despre temperatura, umiditatea și presiunea atmosferică.
2. **Actualizare în Timp Real:** Datele sunt actualizate periodic, simulând schimbările de vreme în timp real.
3. **Afișarea Datelor:** Datele meteorologice sunt afișate pe diverse panouri de afișare (DisplayBoards).
4. **Sistem de Alertă:** Sistemul generează alerte când datele meteorologice depășesc anumite praguri stabilite.
5. **Istoricul Datelor:** Păstrează un istoric al datelor pentru afișarea grafică.
6. **Afișarea grafică**

- **Design Patterns Utilizate**

Singleton Pattern

Aplicat în: **WeatherStation**

- **Scop:** Acest pattern asigură că doar o singură instanță a clasei **WeatherStation** este creată, furnizând un punct global de acces la aceasta.
- **Implementare:** Constructorul **WeatherStation** este privat, iar accesul la instanța unică este controlat prin metoda statică **getInstance()**.
- **Utilizare:** Garantează că toate componentele sistemului folosesc aceeași instanță a stației meteorologice pentru colectarea și gestionarea datelor.

Observer Pattern

Aplicat în: **WeatherStation** și **DisplayBoard**

- **Scop:** Permite definirea unei dependențe de tipul unu-la-mai-mulți între obiecte, astfel încât atunci când un obiect își schimbă starea, toate obiectele dependente sunt notificate și actualizate automat.
- **Implementare:**
 - **Subject (Observable):** **WeatherStation** acționează ca subiect, menținând o listă de observatori și notificându-i despre orice schimbări ale datelor meteorologice.
 - **Observers:** Instanțele de **DisplayBoard** sunt observatori care reacționează la actualizările emise de **WeatherStation**, afișând datele actualizate.
- **Utilizare:** Acest pattern facilitează flexibilitatea și scalabilitatea, permițând adăugarea facilă de noi tipuri de panouri de afișare.

II. Implementare

Clasa WeatherStation

WeatherStation este o încarnare a ingineriei software avansate, îmbrățișând principiile Singleton și Observer pentru a furniza o sursă centralizată și consistentă de date meteorologice. Aceasta simulează și distribuie informații privind temperatură, umiditate și presiune, fiind echipată cu mecanisme de alertă pentru condiții extreme. Integrarea sa cu **WeatherChart** și capacitatea de a gestiona istoricul datelor demonstrează o aplicare ingenioasă a principiilor de colectare și gestionare a datelor în timp real, reprezentând un exemplu remarcabil de arhitectură software bine gândită.

Clasa DisplayBoard

DisplayBoard, implementând design pattern-ul Observer, este o clasă ce exemplifică aplicarea principiilor de design de înaltă calitate pentru a răspunde la schimbări de stări în timp real. Se integrează strâns cu **WeatherStation**, reflectând actualizări ale datelor meteorologice. Este un testament al designului modular și reactiv, evidențiind capacitatea de a răspunde rapid și eficient la schimbările de mediu, făcându-l un component esențial în arhitectura orientată pe evenimente.

Clasa Observer

Clasa abstractă **Observer** este o piatră de temelie în aplicarea pattern-ului Observer, oferind o structură de bază pentru obiectele care necesită monitorizare și răspuns la schimbările de stări. Această clasă demonstrează eleganța abstractizării în designul software, permițând extensibilitatea și flexibilitatea în implementarea specifică a

observatorilor. Rolul său în decuplarea obiectelor observate de cele observatoare este crucial pentru a menține coeziunea și acuratețea în răspândirea informațiilor.

Clasa Main

Clasa **Main** funcționează ca nucleu central al aplicației, orchestrând interacțiunea dintre diferitele componente. Ea inițializează și assemblează cu măiestrie obiectele cheie ale aplicației: **WeatherStation**, **DisplayBoard**, și **WeatherChart**. Implementând modelul Singleton pentru **WeatherStation** și stabilind conexiuni între aceste entități, clasa demonstrează un nivel înalt de ingeniozitate în gestionarea fluxului de date și a interdependențelor dintre componente.

Clasa WeatherChart

Clasa **WeatherChart** este o ilustrare sofisticată a ingineriei software în aplicarea de vizualizare a datelor meteorologice. Extinzând **JFrame**, aceasta creează o interfață grafică utilizator, integrând un set robust de serii de date XY via JFreeChart. Este o reprezentare grafică dinamică, cu serii de date pentru temperatură, umiditate și presiune, îmbogățită cu capacitatea de actualizare în timp real a datelor. Eleganța sa constă în abstractizarea complexității de prezentare a datelor meteorologice într-o formă grafică intuitivă și interactivă.

