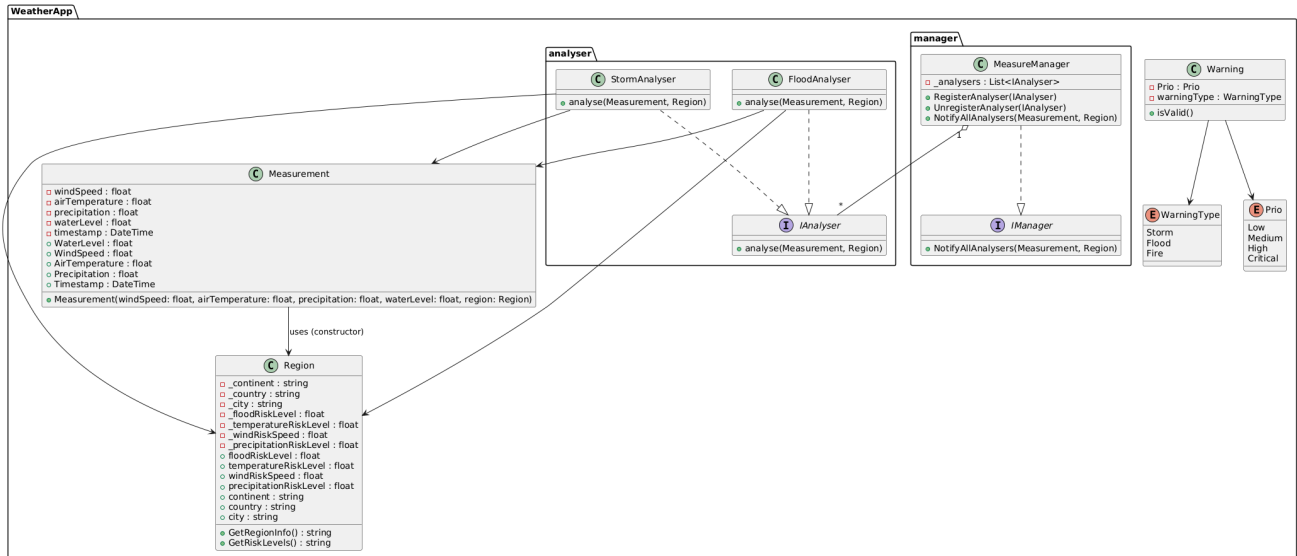


WeatherApp - Documentation

Klassendiagramm

Erstellen Sie ein Klassendiagramm der oben skizzierten Architektur. Beachten Sie die Verwendung der Interfaces, (abstrakten-)Klassen und die Benennung der Methoden



Methode "StormAnalyser"

Implementieren Sie die Methode Analyse in SturmAnalyser (Sturm bei Windgeschwindigkeit > 75)

```
namespace WeatherApp.analyser;

public class StormAnalyser : IAnalyser
{
    public void analyse(Measurement measurement)
    {
        float windRiskSpeed = measurement.Region.windRiskSpeed;
        float windSpeed = measurement.WindSpeed;
        if (windSpeed > windRiskSpeed)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine($"[ { Prio.Medium } ] Wind speed in
{measurement.Region.city} is ABOVE wind risk speed by {windSpeed -
windRiskSpeed:0.00}!");
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine($"[Storm Safe] Wind speed in
```

```

{measurement.Region.city} is BELOW wind risk speed by {windRiskSpeed -
windSpeed:0.00}.";
    }
    Console.ResetColor();
}
}

```

Messungsmanager

Realisieren Sie eine Methode im MessungsManager, welche alle Analyser über eine neue Messung informiert. Beschreiben Sie hierfür das Observer Pattern

```

using WeatherApp.analyser;
namespace WeatherApp.manager;

public class MeasureManager : IManager
{
    private List<IAnalyser> _analysers;
    public MeasureManager()
    {
        _analysers = new List<IAnalyser>();
    }

    public void RegisterAnalyser(IAnalyser analyser)
    {
        _analysers.Add(analyser);
    }

    public void UnregisterAnalyser(IAnalyser analyser)
    {
        _analysers.Remove(analyser);
    }

    public void NotifyAllAnalysers(Measurement measurement)
    {
        foreach (IAnalyser analyser in _analysers)
        {
            analyser.analyse(measurement);
        }
    }
}

```

Testverfahren

Beschreiben Sie ein mögliches Testverfahren der Anforderungen

In diesem Fall sind **Unit-Tests** gut geeignet, da Methoden und Klassen einzeln getestet werden können.

Reflektion

Beschreiben Sie eine allgemeine Vorgehensweise, die bei der Erstellung eines

Softwareprojektes verfolgt werden kann und erläutern Aspekte, die bei einer

Reflexion berücksichtigt werden sollten

Es sollen folgende Aspekte berücksichtigt werden:

- Die Software soll getestet sein
- Code soll gut dokumentiert sein
- Software soll intuitiv und benutzerfreundlich sein
- Methoden und Variablen sollen intuitive Namen haben