

Versuchen Sie immer, Ihren Code zu kommentieren!

Aufgabe 1

Erstellen Sie die Klasse WarpKern. Die Klasse hat als Eigenschaft eine WarpKernTemperatur, die sich ständig ändern kann. Simuliert wird diese Änderung über eine Property (WarpKernTemperatur). Die Klasse WarpKern löst einen Event aus, wenn sich die Temperatur ändert und löst einen weiteren Event aus, wenn die Temperatur über 500 Grad steigt.

Erstellen Sie eine Klasse WarpKernKonsole, welche die Aufgabe hat, die Temperatur und die Warnmeldung auf der Brücke in einer Anzeige (Konsole) darzustellen.

Verwenden Sie ein EventArgs-Objekt, welcher Sie die alte und die neue Temperatur des WarpKerns, sowie die aktuelle Uhrzeit (Ausgabe mittels der Methode DateTime.Now.ToLongTimeString()) übergeben. Geben Sie diese Daten in der WarpKernKonsole aus.

Aufgabe 2

Flüsse führen im Verlauf der Zeit unterschiedlich viel Wasser. Ein zu hoher oder zu niedriger Wasserstand kann für Schiffe, Anwohner und andere Objekte eine Gefahr darstellen. Deshalb muss der Wasserstand eines Flusses überwacht werden, damit bei Gefahr reagiert werden kann.

Ein Fluss in ihrem Modell soll folgende Punkte erfüllen:

- Einen Namen haben
- Einen Wasserstand (Wert zwischen 100 und 10.000)

Es soll folgende Typen von Beobachtern geben:

- Schiffe, die anhalten, sobald der Wasserstand unter 250 sinkt oder über 8000 steigt
- Städte, die eine Wasserschutzwand errichten, sobald der Wasserstand über 8200 steigt
- Klärwerke, die ihre Einleitungen stoppen, sobald der Wasserstand über 8000 steigt und Ihre Einleitungen steigern, wenn der Wert unter 3000 sinkt

Wenn der Wasserstand die Grenzwerte überschreitet, sollte ein Text in der Konsole angezeigt werden.

Der „Rhein“ bekommt die „Beobachter“ Stadt „Köln“, Stadt „Düsseldorf“ und die Schiffe „Rheingold“ und „Lorelei“.

Die Donau bekommt die „Beobachter“ Stadt „Ulm“, die Schiffe „Xaver“ und „Franz“ und das Klärwerk „Strauß 1“.

Aufgabe 3

Implementieren Sie eine Zählerklasse Counter mit einer Methode ZählerstandErhöhen(x) zum Erhöhen des Zählerwerts und einer Methode Clear() zum Löschen des Zählerwerts.

Wenn der Zählerwert eine bestimmte Bedingung erfüllt (z.B. Zählerwert > 1000) soll ein Event ausgelöst werden.

```
Counter meinCounter = new Counter();  
...  
meinCounter.ZählerstandErhöhen(100);  
meinCounter.ZählerstandErhöhen(200);  
...
```

Die Methode ZählerstandErhöhen(x) könnte ohne Delegaten wie folgt aussehen:

```
public void ZählerstandErhöhen(int x)  
{  
    zählerstand += x;  
  
    if (zählerstand >= 1000)  
    {  
        Console.WriteLine("Zählerstand erreicht");  
    }  
}
```

Sowohl die Bedingung als auch die Aktion bei Erreichen des Zählerstandes sollen als Delegaten an die Zählerklasse Counter übergeben werden können. Das Erreichen des Zählerstandes soll ein Event auslösen.