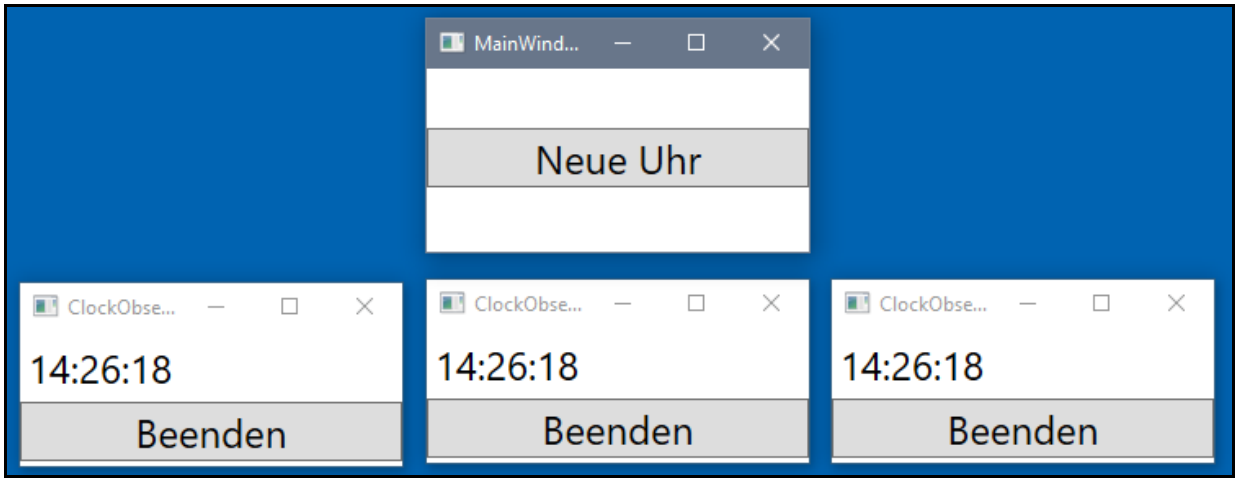
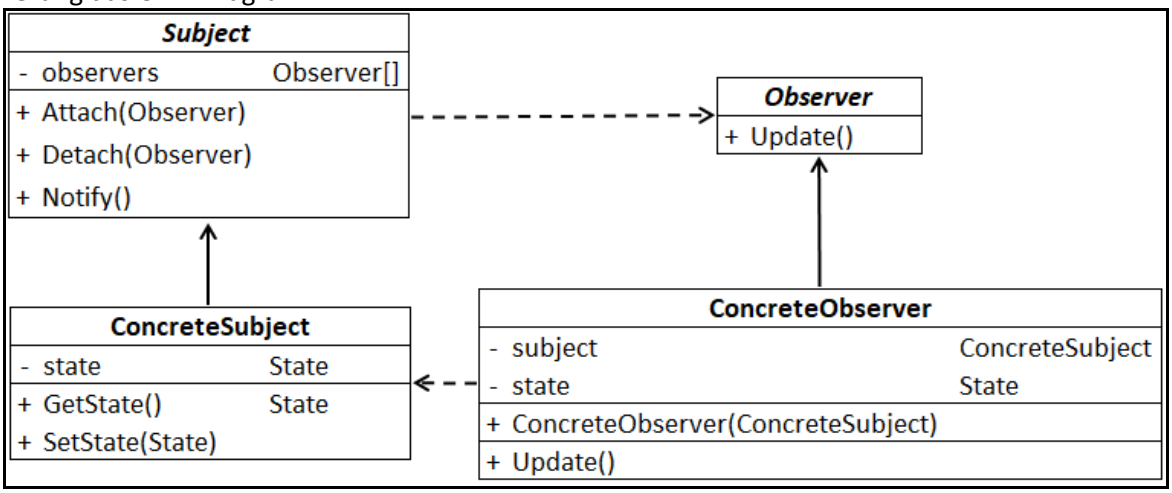
Übung Designpattern Observer
Uhr



Programmiert das Beobachter-Muster wie von mir gezeigt:

- Abstrakte Klasse **`Subject`**
- Klasse **`ClockSubject`**: verwaltet die aktuelle Uhrzeit
- Interface **`IObserver`**
- Konkreter **`ClockObserverWindow`**: Window, das in einem Label die Uhrzeit anzeigt
- MainWindow als Hauptprogramm:
  - Button zum Erzeugen und Anzeigen einer neuen Uhr (Window **`ClockObserverWindow`**)
  - Button zum Starten des Zeitgebers: Thread, der jede Sekunde dem **`ClockSubject`** die neue Uhrzeit setzt

Zur Erinnerung das UML-Diagramm:



Code-Snippets

Thread in C#:

```
new Thread(() =>
{
  while (true)
  {
    //...
  }
}).Start();
```

Umschalten in UI-Thread (braucht man im ClockObserverWindow):

```
public void Update()
{
  if (!CheckAccess())
  {
    Dispatcher.Invoke(Update);
    return;
  }
  //...
}
```

Erweiterung:
In beiden Windows das Event Closing-Event programmieren.

- ClockObserverWindow: vom Subject abmelden
- MainWindow: while-loop beenden (durch Setzen eines Flags)