The background features a complex network of thin, light gray lines and dots, forming a web-like structure. Scattered throughout are various triangles of different sizes and orientations, some outlined in a slightly darker gray. The overall aesthetic is clean, modern, and technical.

Clean Code & Design Pattern

Tag 6

Zeitplan



12:00
Start

Pause

13:15 & 14:45



16:00
Ende

The background features a complex network of thin, light gray lines and dots, forming a web-like structure. Scattered throughout are various triangles of different sizes and orientations, some with solid outlines and others with dashed or dotted outlines. The overall aesthetic is minimalist and technical.

Review Tag 5

Coding, Coding, Coding

Trennung UI und Daten

- Form (View) ohne Logik
- FormModel (ViewModel) ohne Oberfläche
- “Bindings” für die Verbindung
- Events (INotifyPropertyChanged/CanExecuteChanged) für Kommunikation



ICommand

- ICommand als wiederverwendbare Einheit
- Kann direkt gebunden werden
- Für eine Verbindung zum ViewModel ein kleiner Umweg
 - a. RelayCommand(Action<>, Predicate<>, ViewModel)
 - b. Binding Context



Design Pattern und Principles

- Dependency Inversion Principle
- Strategy und Command Pattern
- Separation of Concerns
- Empty Pattern (`string.Empty`, `EventArgs.Empty`)
- Information Hiding Principle



The background features a complex network of thin grey lines connecting various points, creating a web-like structure. Scattered throughout are numerous triangles of different sizes and orientations, some with solid outlines and others with dashed or dotted lines. The overall aesthetic is technical and digital.

Unsere Kleine Applikation

Coding, Coding, Coding

Aufgabe

FormModelCommandDecorator

Automatisch das Model als Parameter





Observer Pattern

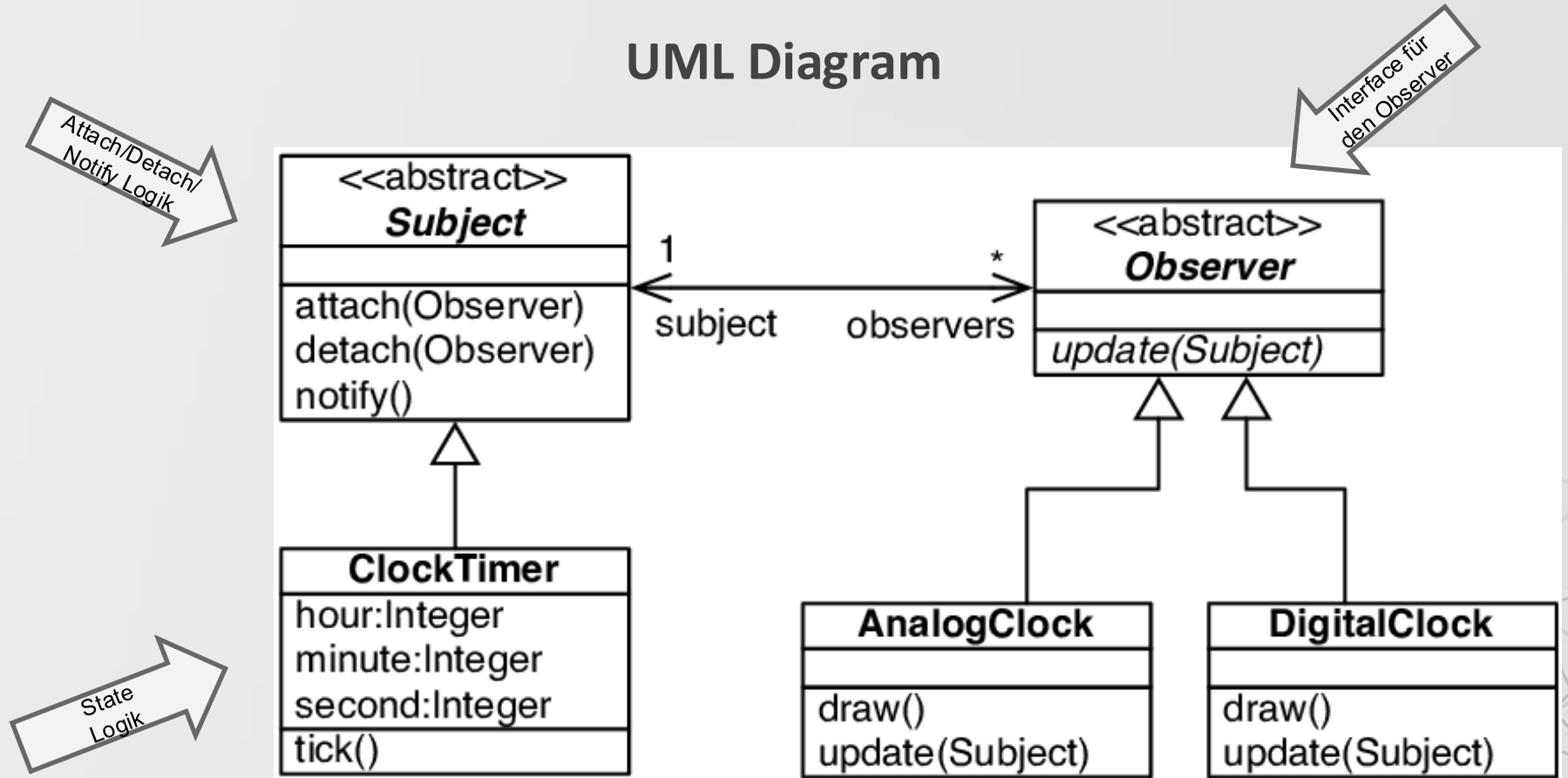
By the book

“The” Observer pattern

- Beobachterentwurfsmuster
- Senden und abonieren von Benachrichtigungen
- IObserver → Update(state) zum Empfangen
- ObserverX implementiert das Interface
- Subject → Notify(state) zum Benachrichtigen aller registrierten Observer
- Subject → Attach(), Detach() zum Hinzufügen/Entfernen von Observern



UML Diagram



INotifyPropertyChanged

- Event CanExecuteChanged
 - a. += → Subscribe
 - b. -= → Unsubscribe
- CanExecuteChanged.Invoke()
 - a. Alle Subscriber werden aufgerufen



The background features a complex network of thin, light gray lines and dots, primarily concentrated on the left side, forming a web-like structure. Scattered across the entire background are numerous triangles of various sizes and orientations, some outlined in a slightly darker gray than the background. The overall aesthetic is minimalist and technical.

Review

Design Pattern, Clean Code und Lebenserfahrung

Clean Code Prinzipien

- Duzende Bücher für besseren Code & <https://cleancodedeveloper.de>
- SOLID (SRP, OCP, LSP, ISP, DIP)
- Version Control System, Code Review, Pair Programming & Pull Requests
- KISS & YAGNI
- “Optimization Rule”
- Refactoring
- Broken Window Principle & Pfadfinderregel
- Information Hiding Principle



Design Pattern

- Command Pattern & Strategy Pattern
- Null-Object Pattern & Composite Pattern
- Factories & Lazy<T>
- Wrapper – Decorator, Adapter, Façade
- Service Locator Pattern
- ZIIP (Zero Impact Injection Pattern)



Lebenserfahrung

- Think Big, Start Small
- new() is Glue
- Drittanbieterschnittstellen schaffen Abhängigkeiten
- ADRs helfen die Entscheidungshistorie zu verstehen
- Abhängigkeiten als Interface. "Interfacen" auch nachträglich.
- "... im Moment reicht erst einmal nur ..." → Austauschbarkeit
- Einzeiler-Dokumentation "Diese Klasse ist zuständig für ... und kennt ..."



Alaaf

