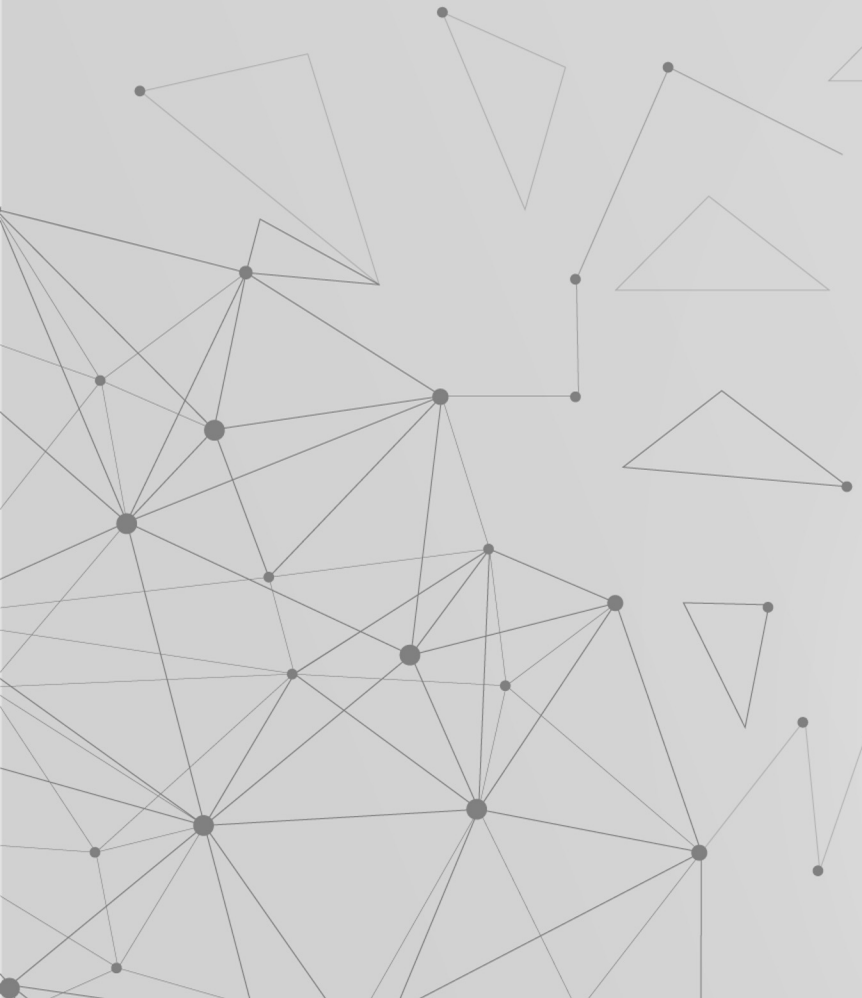


Pattern of the Week

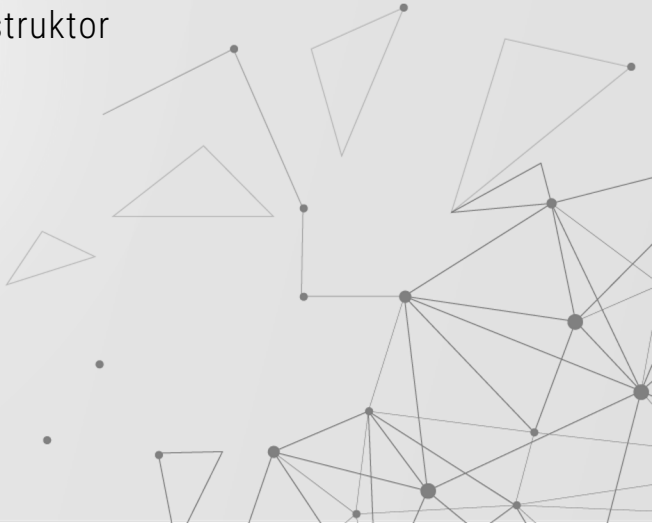
Week 3: Factories & Co.

TABLE OF CONTENTS

Die "falsche" Factory Method

- Ein Objekt wird durch den Aufruf einer Methode erzeugt,

  anstatt durch den direkten Aufruf des Konstrukturs
- Create() Methode in der Klasse statt des Konstruktors
- Validierung, Business Logik und lange Prozesse nicht im Konstruktor



Aufgabe

"Falsche" Factory Method implementieren

Projekt: "Person"

Konstruktor Privat
Create() mit Daten
ArgumentException wenn "null" oder Name weniger als 2 Buchstaben

01

Factory Method

Factory Method

- Ein Objekt wird durch den Aufruf einer Methode erzeugt,

  anstatt durch den direkten Aufruf des Konstrukturs
- Aufrufer kennt die Abhängigkeiten und den Instanziierungsprozess nicht
- Trennung zwischen Erstellung und Verwendung

→ Ich kenne jemanden, der das kann

Aufgabe

Factory Method implementieren

Projekt: “PdfTools”

IPdfArchiverFactory
Create() gibt PdfArchiver zurück

02

Abstract Factory

Abstract Factory

- Erzeugt ein Gruppe von zusammengehörigen Instanzen
- Rückgabe als Interface
- Instanzen können in konkreten Typ gecastet werden

- Beispiel: IWindow, IButton, IContainerControl
- WpfWindow, WpfButton, WpfContainerControl

```
New WpfButton( (WpfContainerControl) controlAsInterface)
```

Aufgabe

Abstract Factory implementieren

Projekt: “PdfTools”

Umbau der IPdfArchiverFactory

Interface IPdfArchiver

Verwendung

- Neue Instanz im aktuellen Code
- “DI” erzeugte Instanz reicht nicht aus
- Smell: State in der Instanz (nicht “static”)

→ Lambda-Factory

Lambda-Factory Method

- Factory Method Pattern ohne Interface und Ableitung
- Konstrukturparameter Func<T> factory
- Funktion die beim Aufruf eine (neue) Instanz zurück gibt

- Aber: Ohne DI Container können sich die Implementierungen unterscheiden
- Wichtig: Bei OOTB Interfaces kann es zu Problemen kommen (Func<String>)

Lambda-Abstract Factory

- Abstract Factory mit generischer Implementierung
- Eine Klasse enthält mehrere Lambda-Properties

- Public Func<T> TeeFactory { get; }
- T bar = FooAbstractFactory.TeeFactory()