

The background is a light gray with abstract geometric patterns. On the left, there is a dense network of dark gray lines connecting various points, some of which are solid black dots. Scattered across the rest of the page are several thin, light gray outlines of triangles of various sizes and orientations. In the top right corner, there are small, faint circles and dots, resembling a starry sky or a data visualization.

PATTERN OF THE WEEK

Week 8: Visitor Pattern

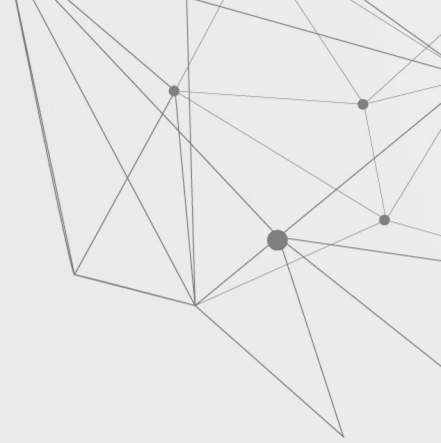
REFACTORING "EXTRACT LIBRARY"

01

VISITOR PATTERN

02

TABLE OF CONTENTS



01

REFACTORING

Extract Library



GEGEBEN: BROWNFIELD PROJECT

REFACTORING: EXTRACT LIBRARY

- Projekt mit einer Komponente oder Bibliothek
- Beispiel: WPF Control, Persistence Layer, Transformation algorithm
- Ziel: Projekt anlegen und als nuget veröffentlichen



VORTEILE

REFACTORING: EXTRACT LIBRARY

- Wiederverwendbarkeit
- 100%* Test Coverage
- Kleineres wartbareres Paket
- Brownfield Projekt wird wartbarer (weniger Code, weniger Angelegenheiten)



HERAUSFORDERUNGEN

REFACTORING: EXTRACT LIBRARY

- (allgemeine) Schnittstellen ohne “Applikationswissen”

Lösung: bspw. IDataProvider

- Änderungen sind aufwändig bei verfrühter Extrahierung

Lösung: Als Projekt in der Solution; Contracts-Projekt: Referenzen im Auge behalten





02

VISITOR PATTERN

FRAGE

ERWEITERN VON FUNKTIONEN

Projekt „Person“

Methode „SaveXml“
Und „SaveJson“



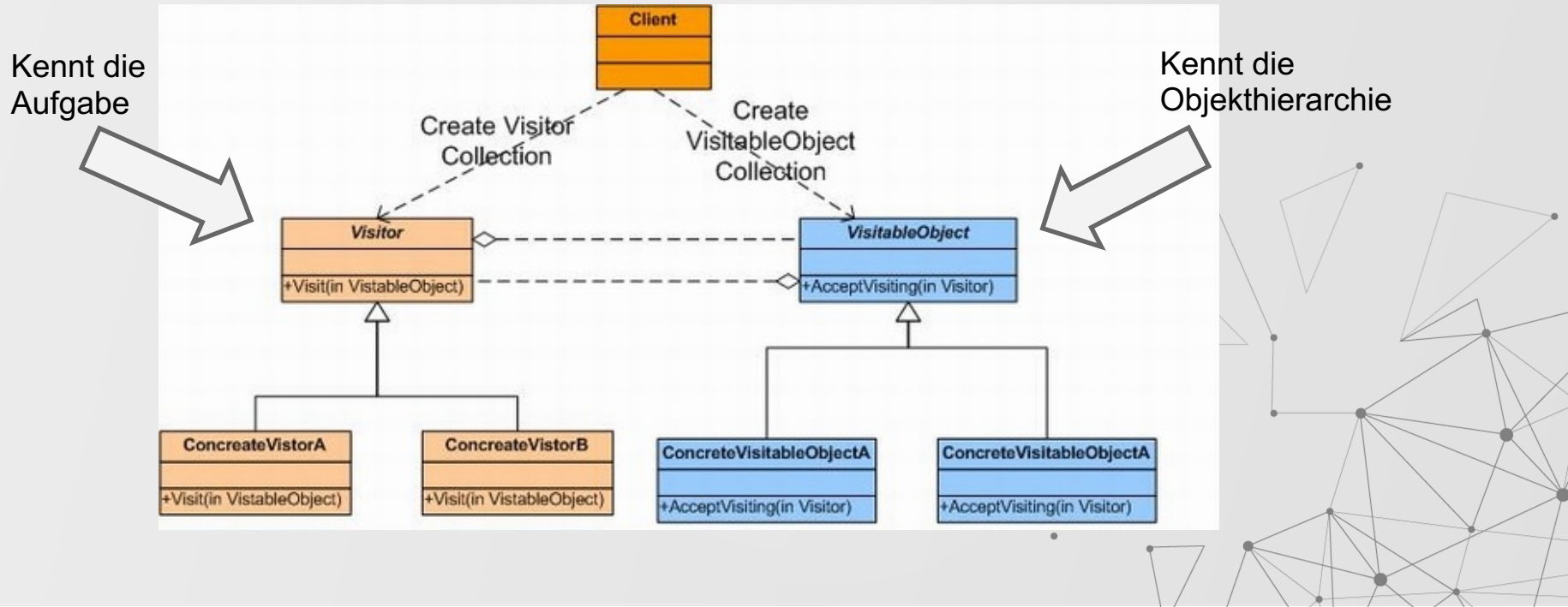
VISITOR PATTERN — DAS PROBLEM

- Ähnliche Objekte (z.B. Person und Boss)
- Operationen auf den (hierarchischen) Objekten
- Separation of Concerns – Operationen nicht in den Objekten implementieren



VISITOR PATTERN

- Wichtig: Die konkreten Klassen kennen nur die Schnittstellen



AUFGABE

IMPLEMENTIEREN DES VISITORS

Projekt „VisitingPerson“

Klasse „JustPrintTheName“
Und „SaveAsXml“

