

# Inhaltsverzeichnis

1	Einleitung .....	2
2	Systemvoraussetzungen .....	2
3	Projektstruktur .....	3
4	UML Diagramm .....	0
5	Datenmodell.....	0
5.1	Child .....	0
5.2	Training .....	0
6	Datenbankdiagramm .....	1
7	Klassen und Member .....	1

# 1 Einleitung

---

Dieses Dokument beschreibt die Entwicklung und Funktionalität der Anwendung ‚Pattern Pix Konzentrationstraining – Trainer Panel‘.

Die Anwendung wurde mit C# und WPF unter Verwendung des MVVM-Architekturmusters realisiert und unterstützt Trainer:innen bei der Verwaltung und Auswertung von Kinderprofilen und deren Trainingsleistungen. Sie dient als begleitendes Werkzeug zur mobilen ‚Pattern Pix‘-App und ermöglicht eine benutzerfreundliche sowie robuste Erfolgskontrolle und Verlaufsanalyse.

## 2 Systemvoraussetzungen

---

Hardware	
Prozessor	Dual-Core, 64-Bit, mindestens 1 GHz
Arbeitsspeicher	1 GB (2 GB empfohlen für flüssige Bedienung)
Festplattenspeicher	200 MB verfügbarer Speicherplatz (inkl. Daten)
Grafikkarte	integrierte Grafik mit Unterstützung für DirectX 11
Displayauflösung	mindestens 1024×768 (für responsive UI optimiert)
Software	
Betriebssystem	Windows 10 oder neuer (64-Bit)
.NET-Laufzeit	.NET Desktop Runtime 8.0 (oder aktueller)
Framework	WPF mit MVVM-Pattern
IDE (Entwicklung)	Visual Studio 2022
Datenbankoptionen	SQLite (lokal) oder JSON-Dateien (serialisiert)

### 3 Projektstruktur

---



---

▼ **N** Model

**C** Child

**C** MovingAveragePoint

**C** PeerComparisonData

**C** PeerSymmetryComparison

**C** SymmetryStats

**C** Training

▼ **N** Repositories

**C** IChildRepository

▼ **N** View

**C** AnalysisView

**C** ChildAddView

**C** ChildDetailView

**C** ChildrenListView

**C** MainView

▼ **N** ViewModel

**C** AnalysisViewModel

**C** BaseViewModel

**C** ChildAddViewModel

**C** ChildDetailViewModel

**C** ChildrenListViewModel

**C** MainWindowViewModel

**C** TrainingDayGroup

**C** TrainingTimeGroup

**C** App

**C** MainWindow

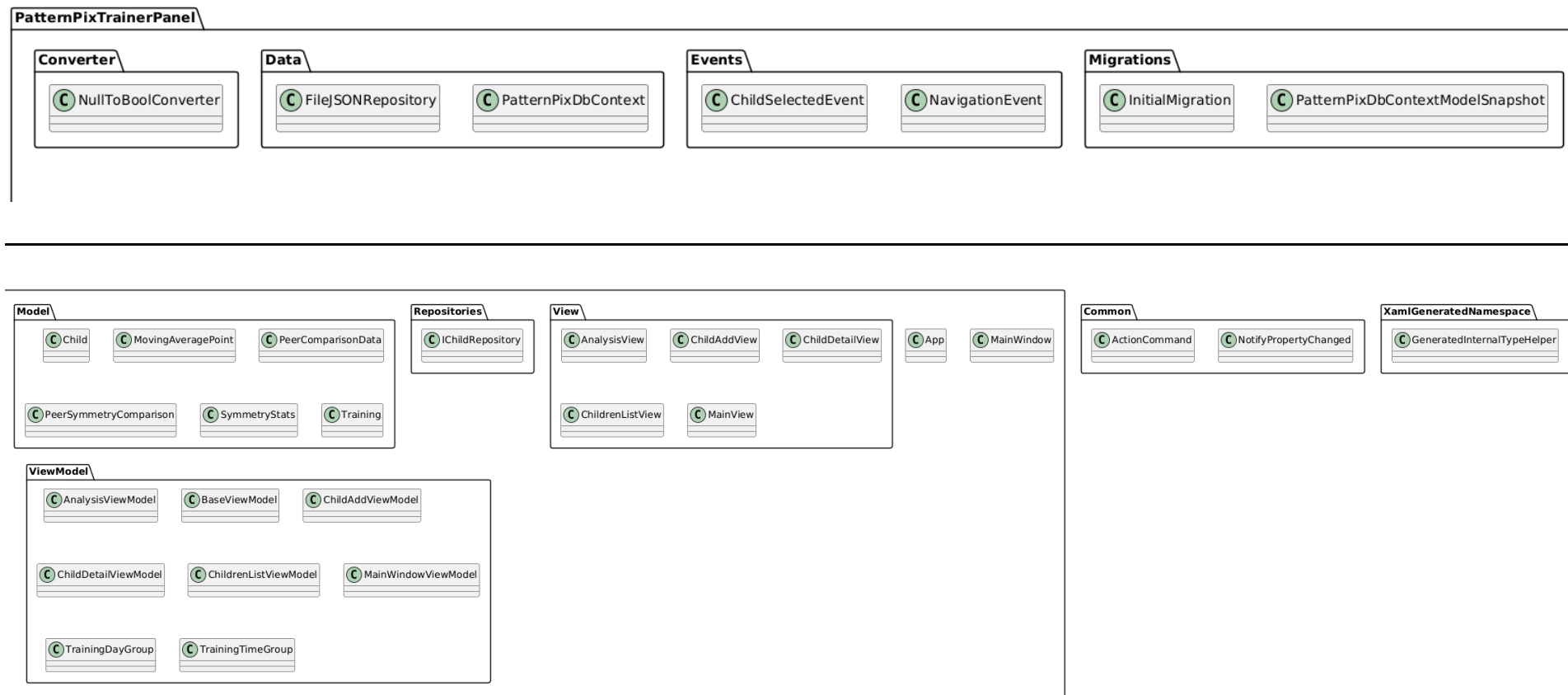
▼ **N** XamlGeneratedNamespace

**C** GeneratedInternalTypeHelper

## 4 UML Diagramm

Das folgende UML-Diagramm gibt einen Überblick über die grundlegende Projektstruktur.

Dargestellt sind die zentralen Namensräume sowie die enthaltenen Klassen. Darüber hinaus visualisiert das Diagramm die Abhängigkeiten zwischen den einzelnen Komponenten.



## 5 Datenmodell

---

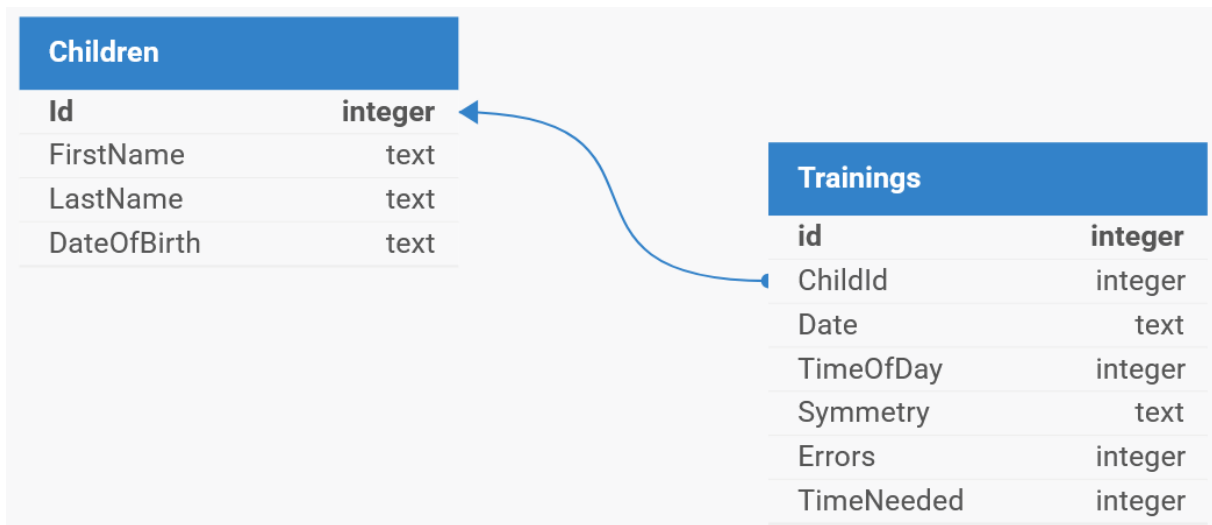
### 5.1 Child

Eigenschaft	Datentyp	Bedingungen
Id	int	NOT NULL, <b>Primary Key</b>
FirstName	string	NOT NULL
LastName	string	NOT NULL
DateOfBirth	DateTime	NOT NULL
Age	int	berechnete Eigenschaft (NotMapped)
FullName	string	berechnete Eigenschaft (NotMapped)
Trainings	List<Training>	Navigationseigenschaft
LastTrainingDate	DateTime?	berechnete Eigenschaft (NotMapped)

### 5.2 Training

Eigenschaft	Datentyp	Bedingungen
Id	int	NOT NULL, <b>Primary Key</b>
ChildId	int	NOT NULL, <b>Foreign Key</b> → Child.Id
Date	DateTime	NOT NULL
TimeOfDay	TimeSpan	NOT NULL
Symmetry	string	optional
Errors	int	optional (Standardwert: 0)
TimeNeeded	int	optional (Standardwert: 0)
Child	Child	Navigationseigenschaft
FormattedDateTime	string	berechnete Eigenschaft (NotMapped)
ErrorsScaled	double	berechnete Eigenschaft (NotMapped)
DateString	string	berechnete Eigenschaft (NotMapped)
Timestamp	DateTime	berechnete Eigenschaft (NotMapped)

## 6 Datenbankdiagramm



Die Datenbank besteht aus zwei Hauptbereichen: **Kinder** und **Trainings**.

- Jedes **Kind** hat einen eigenen Eintrag mit Vorname, Nachname und Geburtsdatum.
- Zu jedem Kind können mehrere **Trainings** gespeichert werden – zum Beispiel: wann das Training stattgefunden hat, wie lange es gedauert hat, wie viele Fehler gemacht wurden und welche Übung (Symmetrieart) trainiert wurde.

Die Trainings sind direkt mit einem Kind verknüpft. Das bedeutet:

**Ohne ein Kind kann kein Training gespeichert werden.**

Die Trainings „gehören“ also immer zu einem bestimmten Kind.

Die Datenbank speichert diese Informationen auf dem Computer, sodass die Anwendung später alles auslesen, anzeigen und auswerten kann – zum Beispiel in Form von Diagrammen oder Tabellen.

## 7 Klassen und Member

- Common

ActionCommand      Implementiert ICommand für UI-Aktionen

NotifyPropertyChanged      Stellt PropertyChanged-Event für Data Binding bereit

- PatternPixTrainerPanel.Converter

NullToBoolConverter      Konvertiert null-Werte in bool für UI-Bindings

- PatternPixTrainerPanel.Data



FileJSONRepository      Lädt und speichert Daten im JSON-Format

PatternPixDbContext    Datenbankkontext (Entity Framework) für Kinder und Trainings

- PatternPixTrainerPanel.Events

ChildSelectedEvent      Event zur Auswahl eines Kindes

NavigationEvent      Event zur Navigation zwischen Ansichten

- PatternPixTrainerPanel.Migrations

InitialMigration      Erstellt DB-Tabellen für Child und Training

PatternPixDbContextModelSnapshot    Snapshot der DB-Struktur für Migrationsverwaltung

- PatternPixTrainerPanel.Model

Child      Repräsentiert ein Kind mit Stammdaten und Trainings

Training      Repräsentiert eine Trainingseinheit eines Kindes

MovingAveragePoint      Punkt für gleitenden Durchschnitt (z. B. Auswertung)

PeerComparisonData      Vergleichsdaten eines Kindes mit Peer-Gruppe

PeerSymmetryComparison    Vergleich von Symmetrie-Leistungen mit Durchschnitt

SymmetryStats      Speichert Symmetrie-bezogene Leistungsdaten

- PatternPixTrainerPanel.Repositories

IChildRepository      Schnittstelle für Datenzugriff auf Kinder und Trainings

- PatternPixTrainerPanel.View

AnalysisView      Anzeige für Auswertungen und Vergleiche

ChildAddView      Ansicht zum Hinzufügen eines Kindes

ChildDetailView      Ansicht mit Detaildaten eines Kindes

ChildrenListView      Übersicht über alle gespeicherten Kinder

MainView      Hauptansicht der Anwendung

- PatternPixTrainerPanel.ViewModel

BaseViewModel	Basisklasse mit PropertyChanged-Logik
MainWindowViewModel	Logik für das Hauptfenster
AnalysisViewModel	Logik zur Datenanalyse und Visualisierung
ChildAddViewModel	Logik zur Eingabe neuer Kinderdaten
ChildDetailViewModel	Logik zur Anzeige und Bearbeitung eines Kindes
ChildrenListViewModel	Logik zur Verwaltung der Kind-Liste
TrainingDayGroup	Gruppiert Trainings nach Datum
TrainingTimeGroup	Gruppiert Trainings nach Zeitfenster

- App

App	Einstiegspunkt der WPF-Anwendung
-----	----------------------------------

- MainWindow

MainWindow	Hauptfenster der Anwendung
------------	----------------------------

- XamlGeneratedNamespace

GeneratedInternalTypeHelper	Automatisch generierter XAML-Helfer
-----------------------------	-------------------------------------