

# Terminkalender in c#

## Inhaltsverzeichnis

Terminkalender in c#	1
Inhaltsverzeichnis	1
Aufgabenstellung	2
Problemstellung	2
Lastenheft / Spezifikation	2
Benutzerhandbuch / Bedienungsanleitung	2
Programm Dokumentation	2
Projektbeschreibung	2
Programmierungsumgebung	2
Projektaufbau	2
Datenmodell	2
Diagramme mittels UML	2
Struktogramm / Programmablaufplan	2
Datenbankmodelle	2
Programmbeschreibung	3
Klassen- / Modulnamen	3
Benutzerschnittstellen	3
GUI	3
Konsolendarstellung	3
Testdokumentation	3

# Aufgabenstellung

Schreiben Sie eine Anwendung, die die Verwaltung von Personen und Terminen ermöglicht. Ein Termin sollte dabei fest mit einer Person, einem Datum, einem Ort und einer Beschreibung verbunden sein. Eine Person weist die typischen personenbezogenen Daten auf.

## Problemstellung

Das Programm soll sowohl eine leichte Benutzerführung bieten, als auch eine robuste konsistente Speicherung der Benutzerdaten gewährleisten.

## Lastenheft / Spezifikation

Das Programm wird als Konsolenanwendung in c# (ggf. unter Zuhilfenahme von Mono) erstellt. Geforderte Kernkompetenzen der Software sind bereits in den Abschnitten Aufgabenstellung und Problemstellung dieser Dokumentation aufgelistet.

## Benutzerhandbuch / Bedienungsanleitung

## Programm Dokumentation

### Projektbeschreibung

#### Programmierungumgebung

Die Software wurde mit „Visual c# express 2010“ unter Windows 7, sowie mit „xamarin Studio“ unter Mono auf OSX 10 entwickelt. Zur kompilierung wurden die Systemweiten, bzw. die zu Mono gehörenden Standardcompiler eingesetzt.

#### Projektaufbau

Die Software wurde modular unter Berücksichtigung Objektorientierter und funktionaler Paradigmen entwickelt.  
Die Software wurde durch den Einsatz von Git unter eine Versionsverwaltung gestellt.

### Datenmodell

#### Diagramme mittels UML

#### Struktogramm / Programmablaufplan

#### Datenbankmodelle

# **Programmbeschreibung**

## Klassen- / Modulnamen

- Program
  - Hauptklasse, welche die statische Main-Methode, als auch statische Helfer-Methoden beinhaltet
- Person
  - Entity-Klasse zur objektorientierten Darstellung einer Person-Entität
- Termin
  - Entity-Klasse zur objektorientierten Darstellung einer Termin-Entität
- Data
  - Entity-Klasse, welche eine Liste von Personen und eine Liste von Terminen kapselt, um so eine gemeinsame de-/serialisierung zu gewährleisten
- DataSerializer
  - Serializer-Klasse für Data
- AbstracLoop
  - Abstrakte Implementierung eines CLI-Loops, der ein assoziatives Array aus Optionsnamen und zugehörigen Funktionen beherbergt
- MainLoop
  - Ein CLI-Loop für die Hauptauswahl des Programs
- PersonLoop
  - Ein CLI-Loop für das Personen-Unterprogramm
- TerminLoop
  - Ein CLI-Loop für das Termin-Unterprogramm
- PersonSortLoop
  - Ein CLI-Loop für das Personen-Sortier—Unterprogramm
- TerminSortLoop
  - Ein CLI-Loop für das Termin-Sortier—Unterprogramm

## **Benutzerschnittstellen**

### GUI

### Konsolendarstellung

## **Testdokumentation**