

Modul 120: Benutzerschnittstelle implementieren

LB2 Projektarbeit „Bibliothek“

Dies ist eine bewertete Arbeit, welche alle Lernenden in **Einzelarbeit** durchführen. Sie behandelt folgende **Lernziele**:

- Für grafische Benutzerschnittstellen vorgegebene Standards und Ergonomieanforderungen anwenden.
- Geeignete Entwurfsmuster zur Trennung von Präsentations- und Anwendungslogik einsetzen. (In unserem Fall Code behind)
- Benutzerschnittstellen implementieren.
- Ergonomie (Benutzerfreundlichkeit) und Funktionalität der Benutzerschnittstellen prüfen.

Die **technischen Grundlagen** wurden alle in den Arbeitsblättern 1-7 erarbeitet:

- Trennung von Darstellung (XAML) und Programmlogik (C#)
- Hierarchie der Präsentationsschicht
- Container für statische und dynamische Layouts
- Nutzung von Steuerelementen mittels Designer und Programmcode
- Behandlung von Ereignissen
- Anbindung von Daten aus der Applikationsschicht
- Benutzerdefinierte Steuerelemente
- Anzeige von Listen
- Automatisches Testen einer Applikation

Ablauf des Projekts:

Das Projekt wird innerhalb von **3 Wochen** realisiert. Es wird davon ausgegangen, dass nebst der Präsenzzeit weitere Arbeiten zu Hause oder in der Lehrfirma aufgewendet werden, dies hängt von der individuellen Arbeitsgeschwindigkeit ab.

Vorphase: Einarbeitung (KW 20)

Die Lehrperson zeigt die Aufgabenstellung, das Bewertungsmodell sowie die Code - Vorgaben. Die Lernenden arbeiten sich ins Projekt ein und stellen allfällige Fragen. Ein bestehendes UML Klassendiagramm dient als Hilfsmittel.

Umsetzungsphase und Abgabe (KW 21-23)

Die Lernenden erstellen die geplanten Funktionalitäten, testen die Korrektheit und dokumentieren ihre Arbeit in einer Tabelle: Datum, Stunden, durchgeführte Arbeiten
→ **Abgabe Projektdatei .zip (des Projektordners + Tabelle Arbeitszeiten)**

Datum der Abgabe auf dem Share:

_____ 8. Juni 2017 _____

Korrektur / Besprechung (KW 24-25)

Alle Lernenden halten eine kurze Besprechung bei der Lehrperson ab. Die Besprechung sollte maximal 5 Minuten dauern. **Alle Lernenden machen eine eigene Bewertung anhand des Bewertungsrasters.** Die Noten werden durch die Lehrperson nach Abgabe *aller* Arbeiten festgelegt und kommuniziert.

Verboten sind:

Codeaustausch unter Lernenden innerhalb der Klasse und des Jahrgangs. Kopieren von Code aus Büchern, Internet usw. Hilfestellungen durch Drittpersonen. Es gelten die Weisungen zu Leistungsbeurteilungen (Sept 2016)¹

Der Einsatz von ReSharper ist erlaubt, ansonsten sind keine Drittkomponenten zugelassen!

¹ https://www.gibb.ch/Berufslehre/Berufe/Documents/IET_ERZ_Weisungen_zu_LB_Richtlinien_Plagiate.pdf

Der Auftrag

Es wird ein Programm erstellt, welches Ausleihen von Büchern verwalten kann. Eine Ausleihe ist definiert durch ein Start- und ein Enddatum. Es werden keine Uhrzeiten angegeben. Das Hauptziel des Programms ist es, den Kunden Auskunft über die Verfügbarkeit der gewünschten Bücher anzuzeigen (Verfügbar Ja/Nein, falls Nein ab wann) und überfällige Bücher einzufordern. Jedes Buch ist genau einmal vorhanden und kann nicht gleichzeitig mehrfach ausgeliehen werden.

Vom Programm existiert eine Vorabversion, welche die Datenobjekte und Testdaten beinhaltet.

Die grafische Oberfläche ist umzusetzen für:

- Listenansicht aller Bücher mit Anzeige der Verfügbarkeit (ja/nein)
- Listenansicht aller Ausleihen mit Anzeige Überfällig (ja/nein)
- Einzelansicht Ausleihe (Create, Read, Update)
- Die Bedienung soll möglichst benutzerfreundlich sein, d.h. das Zusammenspiel der drei Ansichten soll den Anwender in seinen Prozessen ideal unterstützen
 - o Klick auf Buch aus Liste -> Neue Ausleihe erstellen
 - o Klick auf Ausleihe aus Liste -> vorhandene Ausleihe bearbeiten

Logische Vorgaben:

- Die Bedienung der Ausleihen sollte möglichst einfach sein für den Anwender
- Keine doppelten Ausleihen von Büchern (2 Ausleihen am gleichen Datum)
- Ein Kunde kann ein Buch frühzeitig zurückgeben
- Die vorgegebenen Klassen dürfen angepasst (oder neu geschrieben) werden, müssen aber die Funktionalität beibehalten

Vorgaben der Benutzerschnittstelle

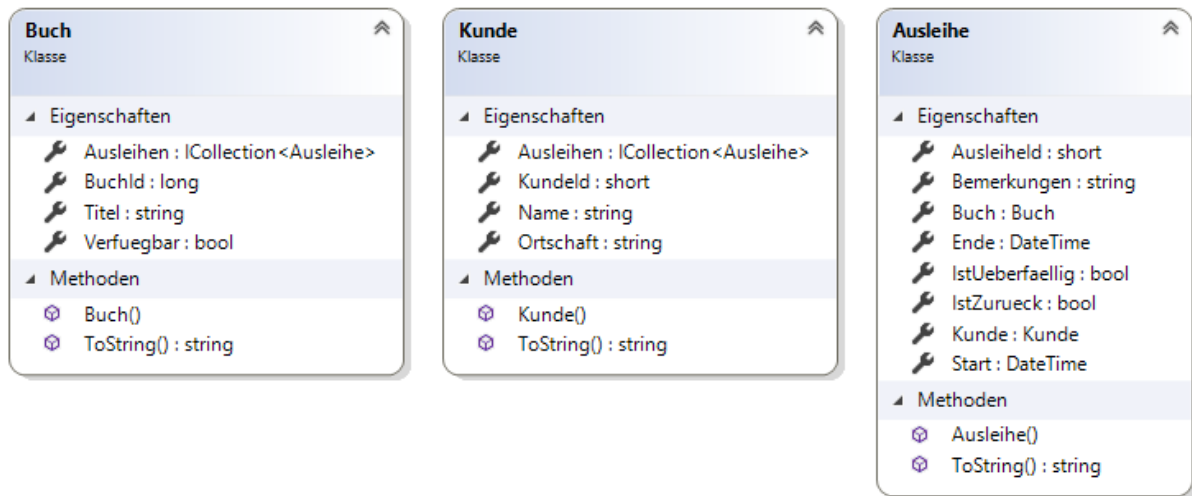
- Sprachen XAML und C#, Desktop Applikation, Visual Studio als Editor
- Freie Wahl des Layouts und der Steuerelemente
- Anzeige in verschiedenen Bildschirmauflösungen (realistische Fenstergrößen) sollen möglich sein wenn nötig

Vorgabe für Tests:

- Es werden 3 Testfälle (coded UI Test) erstellt
 - o Positivtest1:
Bestehendes Buch X
Bestehender Kunde X
Neue Ausleihe erstellen im September
Anzeige Liste alle Bücher (verfügbar)
Neue Ausleihe Januar bis Juli
Anzeige Liste alle Bücher (nicht verfügbar)
 - o Positivtest2:
Bestehendes Buch X (das gleiche wie für Fall 1)
Bestehender Kunde X
Bestehende Ausleihe (Januar bis Juli) frühzeitige Rückgabe gestern
Anzeige Liste alle Bücher (verfügbar)
 - o Negativtest:
Bestehendes Buch X (das gleiche wie für Fall 1)
Bestehender Kunde X
Neue Reservation (Morgen bis Dezember)
- Die Testfälle sollen das Programm starten, den Test durchlaufen und das Programm schliessen!

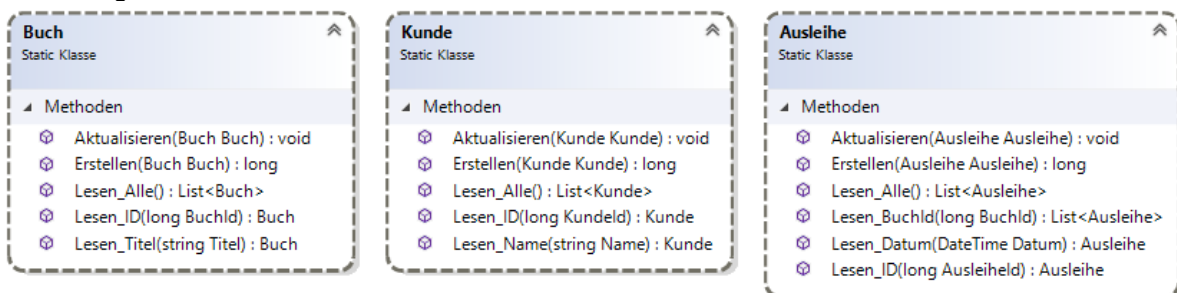
Vorabversion des Programms

Es existieren die folgenden Objekte zur Datenspeicherung:



Jedes Objekt hat eine ID, welche den Primärschlüssel darstellt.
Innerhalb der Bücher und der Kunden werden die Ausleihen referenziert.
Innerhalb der Ausleihen werden die Bücher und Kunden in Listenform referenziert.

Als Applikationsschicht stehen statische Klasse zur Verfügung, welche sich um die Verknüpfungen und Abfragen kümmert:



Mittels «seeding» werden bereits 3 Bücher und 2 Kunden angelegt, diese sind nach dem ersten Start des Programms in der Datenbank «m120_Bibliothek» sichtbar. (Migrations/Configuration.cs)
Die Datenbank kann bei Bedarf gelöscht werden, nach jedem Programmstart wird sie wieder im ursprünglichen Zustand hergestellt. (nur wenn im Programm auch auf sie zugegriffen wird)

In der Klasse MainWindow sind folgende Methoden vorhanden:

- Testdaten: Zeigt wie Ausleihen angelegt werden können
- Check: Prüft die Verfügbarkeit der Bücher sowie die Ausleihen auf Überfällig und gibt diese im Debug-Fenster «Ausgabe» aus.

NICHT LAUFFÄHIGE PROGRAMME WERDEN NICHT BEWERTET!

Bewertungsraster **Klasse** _____ **Name** _____

Beschreibung	Punkte	Erreichte Punkte
Programm läuft in einem einzigen Fenster (Ausnahme Messageboxes)	2	
Listenansicht von Büchern (1=Autogenerated, 2=Benutzerdefiniert)	2	
Listenansicht von Ausleihen (1=Autogenerated, 2=Benutzerdefiniert)	2	
Listenansicht von Büchern kann Einzelansicht aufrufen (Neue Ausleihe)	2	
Listenansicht von Ausleihen kann Einzelansicht aufrufen (Ausleihe bearbeiten / Rückgabe)	2	
Einzelansicht Layout, Ergonomie und Auswahl Steuerelemente (Nur gültige Funktionen möglich, Benutzerführung)	2	
Einzelansicht CRU funktioniert	2	
Logik: Keine doppelten Ausleihen möglich	2	
Logik: Keine Ausleihe möglich wenn Buch noch nicht zurück	2	
Manueller Benutzertest Ergonomie Gesamtapplikation (Durch Lehrperson)	2	
Manueller Benutzertest ohne Abstürze (Durch Lehrperson)	2	
Automatischer Testfall 1	2	
Automatischer Testfall 2	2	
Automatischer Testfall 3	2	
TOTAL	28	
NOTE		

Bemerkungen: