

VAWI OBJC WS 23/24 Block B

Hausarbeit

Aufgabe 1

von: Schubert, Patrick

1 Dokumentation C# - Projekt

1.1 Allgemeine Beschreibung:

Das Werkzeugverleihprojekt wurde mithilfe der IDE Visual Studio 2022 und dem .NET Core Framework 8.0 entwickelt. Es handelt sich um eine Konsolenanwendung, bei der der Benutzer über die Konsole mit den verschiedenen Menüpunkten interagiert. Die nachfolgende Abbildung veranschaulicht beispielsweise den Einstieg in das Werkzeugverwaltungsprogramm über das Hauptmenü.

```
*****
Main Menue
*****
1 Customer Management
2 Tool Management
3 Tool Category Management
4 Lending Management
5 Quit Program
*****
Please select menue item and hit enter: |
```

Abb. 1.: Hauptmenü

Die Werkzeugverwaltung besteht aus insgesamt vier Untermenüs: Kundenverwaltung, Werkzeugverwaltung, Werkzeug-Kategorienverwaltung und Ausleihverwaltung. In diesen Menüs kann der Benutzer vorwiegend numerische Eingaben tätigen. Die Entscheidung für die englische Sprache wurde bewusst getroffen, um Benutzern und Entwicklern unterschiedlicher Herkunft die Bedienung und Weiterentwicklung des Projekts zu erleichtern.

In den einzelnen Untermenüs wurden die CRUD-Operationen (Erstellen, Lesen, Aktualisieren und Löschen) implementiert. Die nachfolgende Abbildung veranschaulicht dies beispielhaft am Kundenmenü (Customer):

```
*****
***** Menue Customer *****
*****
1 Create Customers
2 Edit Customers
3 Delete Customers
4 Display Customers
5 Back to main menue
*****
Please select menue item and hit enter:
```

Abb. 2.: Untermenü Customers

Durch numerische Eingabe wählt der Benutzer eine entsprechende Option aus und gelangt dadurch in ein weiteres Untermenü seiner Wahl.

1.2 Aufbau des Programms im Backend

Das Programm ist wie folgt strukturiert: Für die einzelnen Klassen existiert jeweils ein entsprechender Ordner im Verzeichnis "Classes" (siehe Abbildung 3). Dort befindet sich beispielsweise die abstrakte Klasse `CRUDManager`, die die einzelnen Methoden für die Manager-Klassen vorschreibt. Diese Methoden werden dann von den Klassen `CustomerManagement`, `LendingManagement`, `ToolManagement` und `ToolCategoryManagement` entsprechend implementiert. Zusätzlich gibt es Datenklassen für `Customer`, `Tool`, `Lending` und `ToolCategory`.

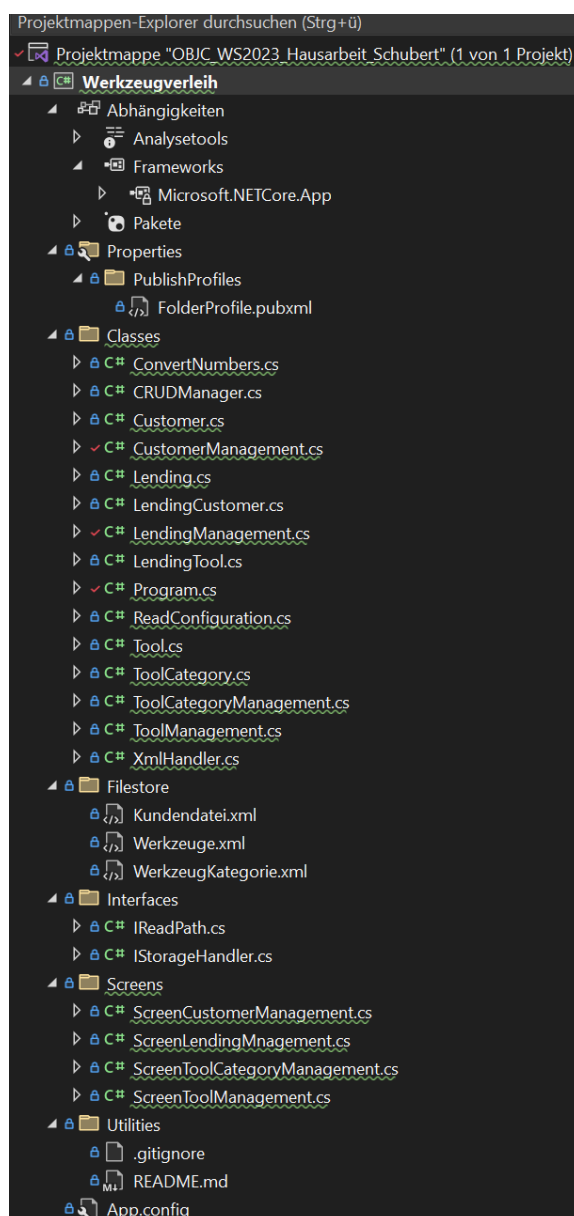


Abb. 3.: Übersicht Programmaufbau

Die Klasse XMLHandler wird von den Manager-Klassen benötigt und ermöglicht das Lesen und Schreiben von Daten, hier in Form von XML-Dateien. "Reinschreiben" bedeutet hier, dass Serialisieren und "rausschreiben" das Deserialisieren der Daten. Entsprechend stellt das Interface IStorageHandler diese Funktionalitäten für die Manager-Klassen zur Verfügung. Zusätzlich sorgt die Klasse ConvertNumbers dafür, dass Benutzereingaben über die Konsole in das richtige Zahlenformat konvertiert werden.

Die Klasse ReadConfiguration liest die Dateipfade und Dateinamen der XML-Dateien entsprechend ein. Diese Konfiguration wird über die Datei App.config bereitgestellt, sodass Benutzer ihre Speicherpfade anpassen können. Es sei jedoch darauf hingewiesen, dass die Pfade auf dem eigenen Rechner entsprechend anzupassen sind. Das Interface IReadPath stellt die Funktionalität der Klasse ReadConfiguration entsprechend zur Verfügung.

Der Ordner Filestore dient als Speicherort für alle XML-Dateien und fungiert somit als Container für die Datenbankobjekte des Projekts. Insgesamt existieren vier Dateien: eine für Kunden, eine für Ausleihen, eine für Werkzeuge und eine letzte für Werkzeugkategorien.

Im Ordner Screens sind die einzelnen Menüs gespeichert, die jeweils als separate Bildschirmseiten repräsentiert werden. Dafür existiert jeweils eine Methode namens Show..., mit der ein Menü (also ein Bildschirm) umgesetzt wurde. Die Benutzereingabe wird jeweils an das übergeordnete Menü zurückgegeben, was es ermöglicht, den Zustand der Eingabe durch den Benutzer an die darüber liegenden Menüs weiterzugeben.

1.3 Schwierigkeiten und Probleme

Die größte Herausforderung bei der Umsetzung bestand darin, den Verleih zu realisieren, da dabei unterschiedliche Objekte gemeinsam serialisiert werden mussten. Aus diesem Grund wurden zwei weitere Klassen, LendingCustomer und LendingTool, eingeführt, um die Objekte jeweils zu „entpacken“. Das bedeutet, dass ihre Eigenschaften einzeln abgerufen werden, um sie später dem Lending-Objekt übergeben zu können. Auf diese Weise können Ausleihvorgänge (Lending-Objekte) mit den zuvor entpackten Eigenschaften des jeweiligen Werkzeugs, das verliehen werden soll, und den Eigenschaften eines Kunden, der ein Werkzeug ausleihen möchte, gemeinsam gespeichert werden. Mithilfe einer Listen-Datenstruktur, in der Werkzeuge gespeichert

werden, und der zuvor beschriebenen Logik können auch mehrere Werkzeuge von einem Kunden ausgeliehen werden.

Bisher gibt es keine Möglichkeit eine Eingabe während der CRUD-Operationen zu unterbrechen, außer das Programm abzubrechen bzw. es komplett zu beenden.

Die Implementierung der standardmäßigen Autonummer-Funktion (engl. Autonumber), wie bei einer Datenbank stellte sich als herausfordernd dar. Das bedeutet, dass eindeutige IDs für einzelne Elemente separat berechnet werden müssen. Derzeit gibt es keine Begrenzung für die Anzahl der IDs, was zu einem potenziellen Überlauf in der Zukunft führen könnte.

Eine einfachere, aber nicht gewünschte Lösung wäre die Verwendung der Random-Funktion zur Vergabe von IDs. Stattdessen wird derzeit eine neue ID durch Inkrementieren der höchsten vorhandenen ID in der XML-Datei berechnet. Wenn beispielsweise die höchste vorhandene ID 10 ist, wird der nächste Eintrag mit der ID 11 versehen. Gelöschte IDs werden nicht wieder verwendet, es sei denn, die Datei wird gelöscht und neu erstellt.

Es ist möglich, dass nicht jede Benutzereingabe optimal abgefangen wird und nicht alle Eventualitäten berücksichtigt wurden. Daher wird nicht fortlaufend überwacht, ob die Eingaben Sinn ergeben. Es wird jedoch erwartet, dass der Benutzer seine Eingaben mit Sachverstand macht.

1.4 Annahmen

- Kunden, Werkzeuge, Werkzeugkategorien und Ausleihen erhalten jeweils eine eindeutige ID.
- Die IDs werden automatisch, basierend auf der bisher höchsten in der XML-Datei vorhandenen ID berechnet. Daher richtet sich ihre Höhe nach der Anzahl der angelegten Kunden, Werkzeuge, Kategorien oder Ausleihen. Es gibt bisher keine Begrenzung für die ID-Höhe.
- Über eine ID der Werkzeugkategorie können Werkzeuge dieser Kategorie zugeordnet werden.
- Ein Kunde kann mehrere Werkzeuge ausleihen, daher kann er mehrfach in der XML-Datei für Ausleihen auftauchen. Außerdem kann derselbe Kunde mehrfach dasselbe Werkzeug ausleihen, da

möglicherweise mehrere Werkzeuge des gleichen Typs benötigt werden.

- Ausleihen können nur für ganze bzw. volle Tage durchgeführt werden.
- Die Gesamtkosten werden pauschal durch Multiplikation der Anzahl der Ausleihtage mit dem Preis pro Tag berechnet: $\text{Anzahl Tage} * \text{Preis pro Tag}$.
- Der Wochenpreis ist in den XML-Daten nicht enthalten, aber in der Datenklasse implementiert. Er wird pauschal mit $6 \text{ Tagen} * \text{Preis pro Tag}$ berechnet, da für eine Woche ein kleiner Rabatt gewährt wird und an den meisten Sonntagen in der Regel nicht gearbeitet wird.
- Die Gesamtkosten der Ausleihe, unabhängig davon, ob es sich um den Wochen- oder Tagespreis handelt, werden über die Eigenschaft "Gesamtkosten" abgebildet.

1.5 Datensicherung

Das Projekt ist unter dem folgenden GitHub-Repository verfügbar:

https://github.com/PatrickSchubert-Munich/OBJC_WS2023_Hausarbeit_Schubert