VAWI OBJC WS 23/24 Block B

Hausarbeit

Aufgabe 1

von: Schubert, Patrick

# Dokumentation C# - Projekt

## Allgemeine Beschreibung:

Das folgende Projekt eines Werkzeugverleihs wurde mit der IDE Visual Studio 2022 und dem .NET Core Framework 8.0 entwickelt. Das Projekt wurde als Konsolenanwendung geschrieben. Demnach interagiert der Benutzer mit der Konsole, um einzelne Menüpunkte in seiner Werkzeugverwaltung zu wählen. Der Einstieg in die Verwaltung bzw. in das Programm findet dabei über ein Hauptmenü statt, wie nachfolgend gezeigt:

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abb. 1.: Hauptmenü

Insgesamt gibt es die 4 Untermenüs Customer Management, Tool Management, Tool Category Management und Lending Management in denen der Benutzer seine Eingaben vollziehen kann. Es wurde bewusst die englische Sprache gewählt, damit Benutzer und Entwickler anderer Herkunft das Projekt bedienen und weiter entwickeln können.

In den jeweiligen Untermenüs sind die sog. CRUD-Operationen (create, read, update und delete) umgesetzt worden. Die nachfolgende Abbildung zeigt dies exemplarisch am Beispiel des Untermenüs für Kunden (Customer):

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abb. 2.: Untermenü Customers

Die Eingaben werden überwiegend numerisch vorgenommen. D. h. der Benutzer gibt eine entsprechende Nummer ein und gelangt dadurch in ein weiteres Untermenü seiner Wahl, indem weitere Eingaben gemacht werden können.

## Aufbau des Programms im Backend

Das Programm ist wie folgt aufgebaut. Für die einzelnen Klassen gibt es entsprechend einen Ordner Classes (s. Abb. 3). Hier ist z. B. die Abstrakte Klasse CRUD-Manager, die für die Manager Klassen die einzelnen Methoden vorschreibt. Diese Methoden werden von den Klassen Customer Management, Lending Management, Tool Management und Tool Category Management entsprechend implementiert. Außerdem gibt es jeweils Datenklassen für Customer, Tool, Lending und Tool Category.

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abb. 3.: Übersicht Programmaufbau

Die Klasse XML-Handler wird von den Manager Klassen benötigt und sorgt dafür, dass Daten – hier in Form von XML-Dateien – rein- und rausgeschrieben werden können. Reinschreiben meint hier serialisieren und rausschreiben deserialisieren. Dem folgend stellt ein Interface IStorageHandler diese Funktionalitäten zur Verfügung. Die Klasse Convert Numbers sorgt dafür, dass Benutzereingaben über die Konsole in das richtige Zahlenformat geparst werden.

Eine weitere Klasse ReadConfiguration liest entsprechend den Dateipfad und die Dateinamen der XML-Dateien ein. Diese Konfiguration wird über die Datei App.config zur Verfügung gestellt. Folglich kann ein Benutzer seine Speicherpfade anpassen. Hier ergeht der Hinweis, dass die Pfade auf dem eigenen Rechner bitte entsprechend selbst anzupassen sind. Das Interface IReadPath stellt die Funktionalität der Klasse ReadConfiguration entsprechend zur Verfügung.

Im Ordner Filestore befinden sich alle XML-Dateien. Dies sind insgesamt 4 Dateien. Eine Datei für die Customer, eine für Lendings, eine für Tools und eine letzte für ToolCategories.

Im Ordner Screens befinden sich die einzelnen Menüs, die als einzelne Bildschirmseite repräsentiert werden. Dafür gibt es jeweils eine Methode Show…, mit der ein Bildschirm (Bildschirm = Menü) umgesetzt wurde. Der User Input wird jeweils an das darüber liegende Menü aus den Methoden zurückgegeben. Dadurch kann der Zustand der Eingabe durch den Benutzer an die darüber liegenden Menüs übergeben werden.

## Schwierigkeiten und Probleme

Bei der Umsetzung war die größte Herausforderung den Verleih umzusetzen, denn dabei mussten jeweils unterschiedliche Objekte gemeinsam serialisiert werden. Deshalb gibt es zwei weitere Klassen LendingCustomer und Lending Tool, die die Objekte jeweils „entpacken“. D. h. deren Properties werden einzeln geholt, um sie später in das Objekt Lending übergeben zu können. Dadurch kann eine Ausleihe (Objekt Lending) mit den zuvor entpackten Eigenschaften aus dem jeweiligen Werkzeug, dass verliehen werden soll und den Eigenschaften eines Kunden, der ein Werkzeug ausleihen möchte, gemeinsam abgespeichert werden. Durch eine Datenstruktur Liste, in denen Werkzeuge abgelegt werden und durch die zuvor beschriebene 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.

Schwierig war es, die in einer Datenbank standardmäßige sog. Autonumber Funktion umzusetzen. D. h. eindeutige Ids für die einzelnen Items müssen separat berechnet werden. Momentan ist die Anzahl nicht begrenzt, sodass es irgendwann in der Zukunft zu einem Overflow kommen könnte. Einfacher – aber nicht gewollt – wäre es, eine Id mit der Random-Funktion zu vergeben. Derzeit wird eine neue Id durch die höchste vorkommende Id in der XML-Datei berechnet. Ist diese z. B. 10, wird der nächste Eintrag mit der Id 11 vorgenommen. Bisher gelöschte Id´s werden nicht mehr aufgefüllt. Wird also der Eintrag mit der Id = 8 gelöscht und ist die höchste Id=10 in der XML-Datei, wird die Id mit der Nummer 8 nicht wieder vergeben. Außer die Datei wird gelöscht und neu befüllt.

## Annahmen

* Kunden, Werkzeuge, Werkzeugkategorien und eine Ausleihe erhalten jeweils eine eindeutige Id
* Die Id´s werden automatisch, auf Basis der bisher größten vorkommenden Id innerhalb der XML-Datei berechnet. Folglich richtet sich diese nach der Anzahl der angelegten Kunden, Werkzeuge, Kategorien oder Ausleihen. Bisher wird deren Höhe nicht begrenzt.
* Über eine Id der Werkzeugkategorie können dementsprechend Werkzeuge einer Kategorie zugeordnet werden
* Dadurch ein Kunde mehrere Werkzeuge ausleihen kann, taucht dieser mehrfach in der XML für die Ausleihe auf. Außerdem kann dieser mehrfach das gleiche Werkzeug ausleihen, da ggf. auch mehrere Werkzeuge vom gleichen Typ benötigt werden.
* Die Ausleihe kann nur für ganze bzw. volle Tage vorgenommen werden
* Die Gesamtkosten werden pauschal durch die Anzahl der Tage der Ausleihe berechnet mit: Anzahl Tage \* Preis pro Tag
* Der Wochenpreis taucht in den XML-Daten nicht mit auf, ist allerdings in der Datenklasse mit umgesetzt. Dieser wird pauschal mit 6 Tagen \* preis pro Tag berechnet, da für eine Woche ein kleiner Rabatt gewährt werden soll und an Sonntagen i. d. Regel nicht gearbeitet wird.

## Datensicherung

Das Projekt kann unter dem folgenden GitHub Repo abgerufen werden:

https://github.com/PatrickSchubert-Munich/OBJC\_WS2023\_Hausarbeit\_Schubert