Programmentwurf

einer Todo-Liste in der

Programmiersprache Microsoft C#

Fach: Advanced Software Engineering

vorgelegt von

Michaela Fleig und Mohammad Mehjazi

Matrikel-Nr. HIER und HIER

Inhaltsverzeichnis

[Bearbeitung der Aufgabenstellung (Praxisprojekt) 3](#_Toc69157778)

[Auswahl geeigneter Methoden 3](#_Toc69157779)

[Formulierung der verwendeten Algorithmen in einer Programmiersprache 3](#_Toc69157780)

[Testen und Überprüfen der Ergebnisse 3](#_Toc69157781)

[Programmdokumentation 3](#_Toc69157782)

[Unit Test (mind. 10) 3](#_Toc69157783)

[Programming Principles 3](#_Toc69157784)

[Refactoring (mind. 2) 3](#_Toc69157785)

[Clean Architecture (mind. 2 Schichten) -> 3](#_Toc69157786)

[Entwurfsmuster (mind. 1) -> Presentation Model (setzt auf MVC auf) nach Gernot Starke 3](#_Toc69157787)

# 

# Bearbeitung der Aufgabenstellung (Programmentwurf)

## Auswahl geeigneter Methoden (Wahl Programmiersprache, OS)

## Formulierung der verwendeten Algorithmen in einer Programmiersprache

## Testen und Überprüfen der Ergebnisse

# Programmdokumentation

## Unit Test (mind. 10) -> ein paar (Hashes, beschreiben der Funktionalitäten, AAA, Code Coverage, Mock)

## Programming Principles -> SOLID-Principles (SRP, OCP nicht möglich, LSP möglich?, ISP fehlt noch, DIP fehlt noch)

## Refactoring (mind. 2) -> Bad Smells (Duplizierter Code unten, Lange Funktionen fehlt noch, mysteriöser Name fehlt noch teilweise, Code-“Klumpen” über Kommentare fehlt noch -> Understand, Shotgun Operation Gefahr mit aktueller Realisierung -> resultiert in Feature Neid)

Refactoring verändert das Programm in kleinen Schritten, dass wenn ein Fehler passiert, ist es einfach, den Bug zufinden. (Refactoring – Improving the Design of Existing Code, Martin C. Fowler, S. 13)

## Clean Architecture (mind. 2 Schichten) -> Verwaltung Tabs, IoHandler & User & Date, DataStorage & GUIs; (auch Presentation Model, nach Gernot Starke)

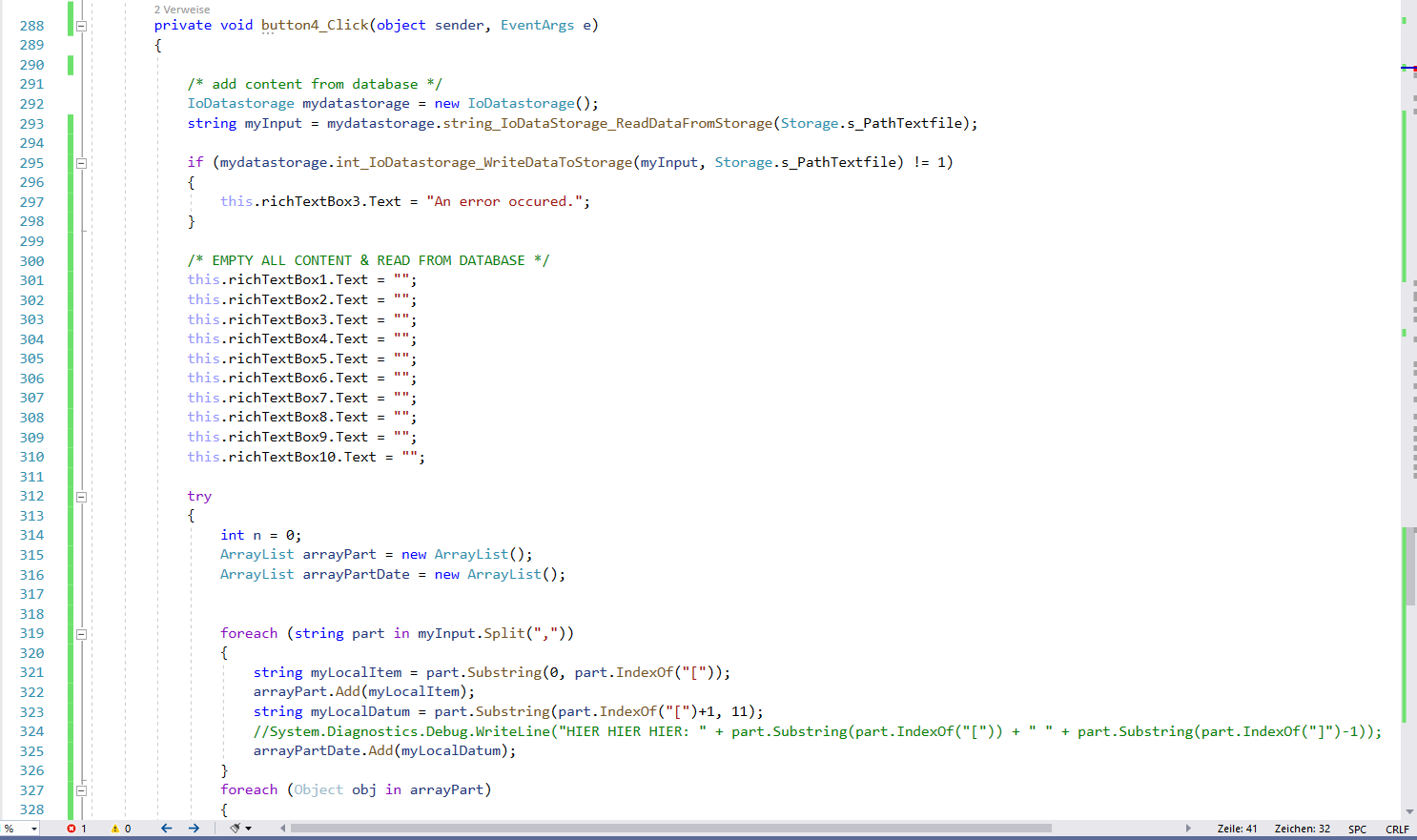
## Entwurfsmuster (mind. 1) -> ?

Refactoring

Von Stand: xxx



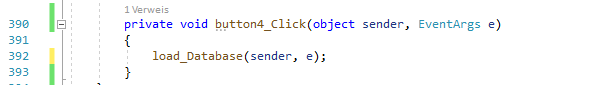
Ist nicht lesbar, nicht voll funktionsfähig, nach Martin C. Fowler ideal zum Refactoring. (Lange Funktion)



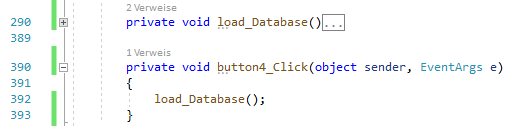
Funktion soll an anderer Stelle ebenfalls verwendet werden. Selbe Funktion, nicht verändert.



Funktion extrahiert. (Refactoring: Extract Method)



Aufruf der Funktion, als ein Simulieren des Buttons.



Funktion als eigenständige Funktion aufgerufen.

\*\*\*\*\*\*\*\*\*\*\*\*\*

Refactoring: Extract Method (Lange Funktion über viele ifs gefunden, doppelte Verwendung der selben Funktionalität) durch Decomposition

Ausgangsversion: f5d1184

* Viele if-Bedingungen, um jede Textbox auf ihren aktuellen Inhalt zu überprüfen und zu beschreiben. Danach Inhalte (Termin-Name und –Datum) formatieren und in Datenbank schreiben. Überprüfung, ob Termin-Datum mit heutigem Datum übereinstimmt. Falls ja, in die nächste freie Textbox von Gruppe „Upcoming Events!“ schreiben.
* Sehr viele ähnliche Bedingungen, wenige inhaltliche Unterschiede. Extract Method.

Schritt-Version: 9c11979

* Extrahierte if-Bedingungen, über einen Switch-Case mit dem Count gelöst.
* Count noch immer in Bedingung. Weitere if-Bedingungen noch offen.

Schritt-Version: e076a78

* Code für Count aus if gelöscht, Debug-Anweisung gelöscht.
* Viel doppelter Code. Wird aber so an keiner weiteren Stelle aktuell gelöscht (nicht umsonst Code für mögliche zukünftige Fälle schreiben), daher aktuell beste Lösung.

Schritt-Version: 59c9d9f

* Löschen von offensichtlich nicht benötigtem Code.
* Noch immer Bedingungen mit ähnlichen Abfragen. Extract Method mit Übergabeparameter.

Schritt-Version: a6c65ed

* Extrahierte if-Bedingungen.
* Methode wird direkt aufgerufen (Remove Variable).

Schritt-Version: eacb19

* Remove Variable, nur noch in extrahierter Methode bekannt.
* Funktion zum Kalender ist noch vorhanden. Remove Variable.

Schritt-Version: 883b515

* Remove Variable, nur noch direkt beim Aufruf verwendet.

Schritt-Version: a724813

* Remove Variable, nur noch direkt beim Aufruf verwendet.

Schritt-Version: c95225c

* Verändere Reihenfolge von if-Statement.

Schritt-Version: f41042f

* Extract Method.

Schritt-Version: be62f25

* Kommentare schreiben.

Finale Version: a881f0c

* Veränderter Name (Rename Method als Teil von Simplifying Method Calls)

Refactoring 2:

Start-Version: 1b20c72

Schritt-Version: ce14b3b

Schritt-Version: 2e6a092 (extrahiert Methode und Variable)

Schritt-Version: 54c15af (extrahiert Methoden und Variablen)

Schritt-Version: 42f9436 (if Bedingung in catch-Block umgewandelt)

Schritt-Version: 544bc03 (kommentierten Code-Block gelöscht)

Refactoring 3:

Start-Version: 25241b4

Schritt-Version: 11b1f1e

Schritt-Version: b91977e

Schritt-Version: b1d0cc7

Schritt-Version: b25d0c0

Clean Architecture:

Schritt-Version: 96c49e6 (umbenennen der Ordner, erste Abhängigkeiten geändert, von main zu gui)