Dokumentation Projekt des ÜK318-1



Lino Bucher

05.12.2018

ÜK 318-1

Abschlussprojekt Dokument

Inhaltsverzeichnis

[Einleitung 3](#_Toc531718000)

[Zweck dieses Dokuments 3](#_Toc531718001)

[Welche Funktionen wurden umgesetzt 3](#_Toc531718002)

[Was ist noch fehlerhaft/unvollständig 3](#_Toc531718003)

[Diagramme der Priorität 1 Aufgaben 4](#_Toc531718004)

[Klassendiagramm 4](#_Toc531718005)

[Use-Case Diagramm 4](#_Toc531718006)

[Aktivitätsdiagramm 5](#_Toc531718007)

[GUI 5](#_Toc531718008)

[Programmierrichtlinien 6](#_Toc531718009)

[Testfälle 10](#_Toc531718010)

[Installationsanleitung 10](#_Toc531718011)

[Reflexion 10](#_Toc531718012)

[Positives 10](#_Toc531718013)

[Negatives 10](#_Toc531718014)

[Anderes 10](#_Toc531718015)

# Einleitung

Im ÜK 318-1 behandelt man das analysieren und das objektorientierte Programmieren. In diesem ÜK haben wir diverse Übungen zum objektorientiertem Programmieren gelöst, welche uns nachher im Abschlussprojekt helfen können. Ausserdem haben wir viel Theorie behandelt, welche auch abgelegt wurde und uns helfen kann.

Das oben genannte Abschlussprojekt besteht darin eine Applikation zu entwickeln, welche einen Fahrplan visualisiert und durch Such-/Filteroptionen vereinfacht. Dazu verwendet man die API-Schnittstelle, welche von SBB zur Verfügung gestellt wird.

# Zweck dieses Dokuments

Das Dokument dient zum besseren Verständnis des Vorgehens und des Codes. Ausserdem soll es meine Gedankensprünge sowie mein Vorgehen aufzeigen.

# Welche Funktionen wurden umgesetzt

In meiner Umsetzung des Projekts wurden folgende Funktionen umgesetzt.

* Auf Knopfdruck werden alle Stationen angegeben, welche auf die Eingabe in eine Combobox des Benutzers zutreffen.
* Nachdem man ausser der Startstation einen Ankunftsort auf dieselbe Weise wie der Startpunkt definiert hat, kann man auf einen Knopf drücken und es werden einem die nächsten vier bis fünf zutreffenden Verbindungen zwischen den beiden Orten angezeigt.

Dies geschieht durch eine DataGridView.

* Wenn man nur eine Startstation eingegeben hat, hat man die Möglichkeit ein Fahrplan aller Verbindungen ausgehend von diesem Ort aus anzeigen zu lassen.

Dies wird auch in einer neuen DataGridView angezeigt, welche erst sichtbar wird, wenn auf den dafür vorgesehenen Knopf gedrückt wird.

* Für das noch einfachere Handling habe ich noch die Autocompletion in den beiden Comboboxen des Abfahrtsortes und des Ankunftsortes eingebaut. Damit fällt eigentlich der erste genannte Punkt weg, jedoch liess ich ihn trotzdem in dem Programm, da doch die eine oder andere Zeile Code dahinterstecken und ich es recht sauber gelöst finde.

Weitere Anforderungsziele habe ich noch nicht weiter programmiert, da ich mir selber die Prioritäten nicht nur auf die Quantität setzte, sondern mehr auf die Qualität.

# Was ist noch fehlerhaft/unvollständig

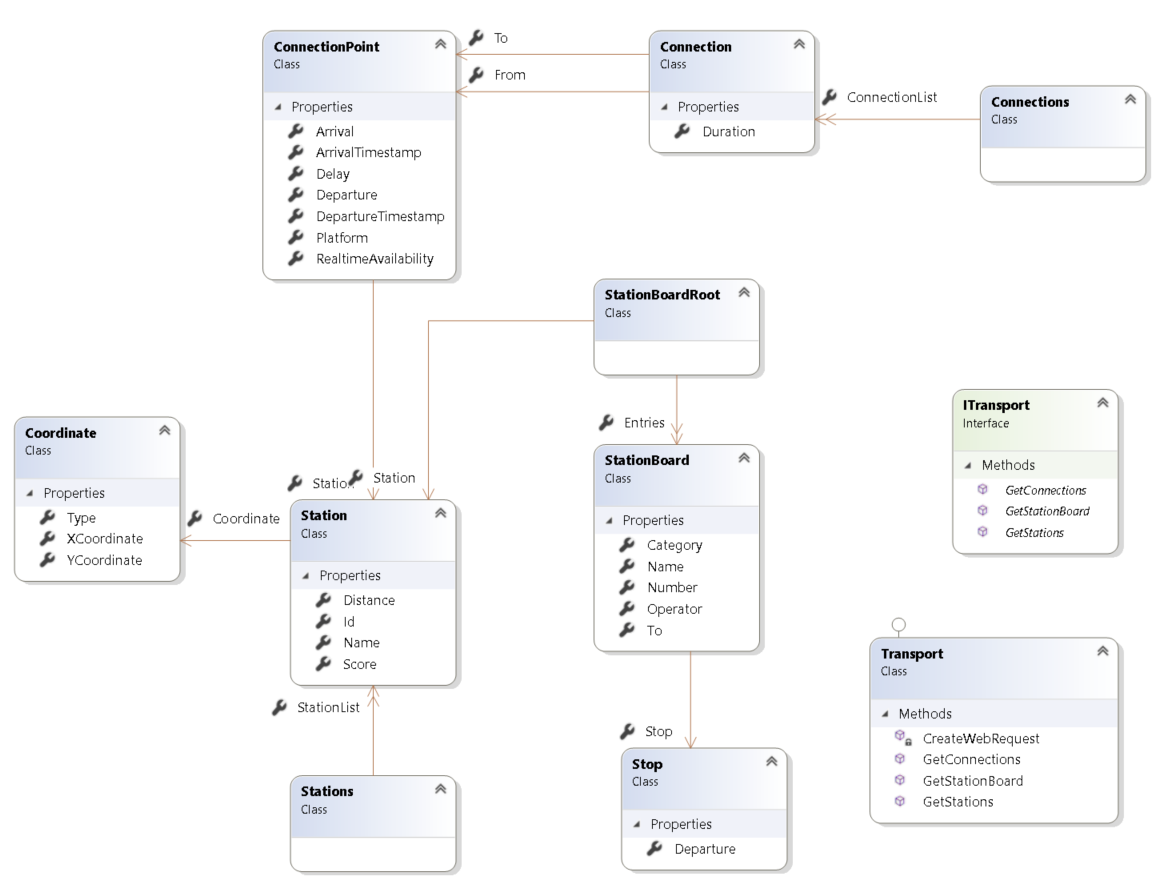
Was noch fehlt sind Anforderungspunkte wie zum Beispiel nach einer Verbindung suchen, welche in einem anderen Zeitraum steht. Ausserdem habe ich die Locations gar nicht eingebunden, da ich wie oben bereits erwähnt die Qualität vor die Quantität setzte.

Meine Autocompletion läuft, jedoch tritt manchmal eine mir unerklärliche Exception auf. Diese konnte ich nicht umgehen, da der Fehler nicht immer auftauchte, sondern nur zu bestimmten Ereignissen. Diese fand ich noch nicht heraus und somit kann ich den Fehler nicht selber aufrufen.

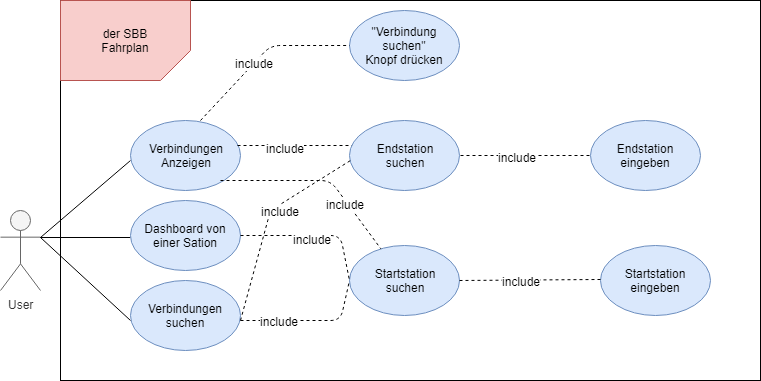
# Diagramme der Priorität 1 Aufgaben

In diesem Absatz meiner Dokumentation finden Sie meine erstellten Diagramme zu den drei Priorität 1 Aufgaben.

## Klassendiagramm



## Use-Case Diagramm



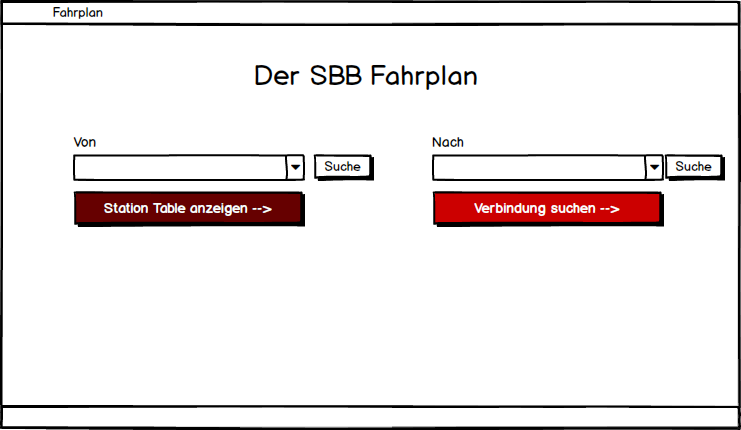
## Aktivitätsdiagramm



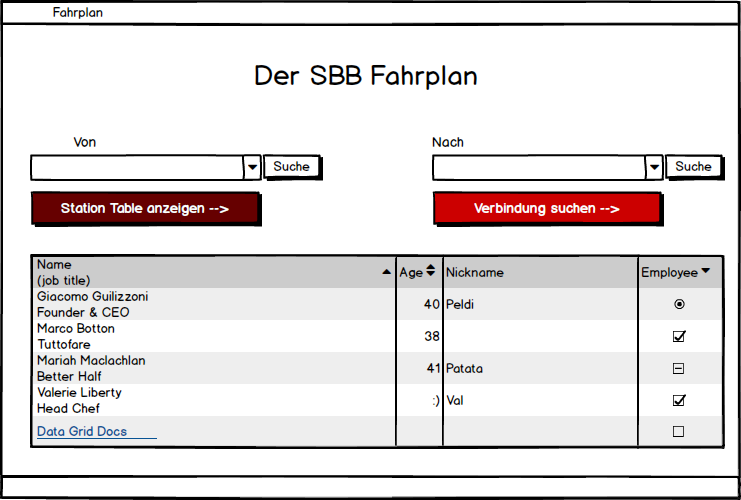
# GUI

In diesem Absatz können Sie mehr über meine Gedanken hinter meinem GUI-Design erfahren. Dazu habe ich in Balsamiq Mockups 3 mein GUI designt und möchte gerne die einzelnen Ausschnitte erläutern.

Zu meinem User Interface ist noch zu sagen, dass es sehr einfach gehalten wurde und es deshalb auch nur zwei Ausschnitte davon gibt.



Wenn man das Tool öffnet, erscheint das obige Fenster. Wenn man in der ersten Combobox beginnt zu schreiben, kommt sofort die Autocompletion und hilft einem bei der Suche nach dem gewünschten Ort. Man kann auch sein Ziel eingeben und dann auf den Button drücken, welcher jeweils rechts von der Combobox liegt. Wenn man nun nur eine Startstation eigegeben hat, erscheint einem der dunkelrote Butten als Verfügbar und man kann Ihn drücken, vorher nicht. Den anderen grossen Button kann man erst drücken, wenn in beiden Comboboxen etwas drinsteht.



Auf diesem Ausschnitt sieht man was passiert, wenn einer der Buttons ausgelöst wird. Es erscheint die jeweilige Datagridview zu dem gedrückten Button. Auf den Button «Station Table anzeigen 🡪» erscheint eine Datagridview mit den dazugehörigen Inhalten, genau wie bei dem Button «Verbindung suchen 🡪»

# Programmierrichtlinien

Was ist der Zweck von gemeinsamen Programmierrichtlinien:

**Wartbarkeit, Lesbarkeit, Ästhetik**

Bei der Erstellung von Software sollten immer folgende Faktoren in Erinnerung gerufen werden:

* Kann ich diesen Source-Code auch in einem Jahr noch verstehen?
* Ist meine Software ordentlich strukturiert/modularisiert oder gibt's haufenweise dubiose

Querverbindungen zwischen den Modulen?

* Was passiert, wenn jemand anderer meinen Code betrachtet ("Code–Review") oder Änderungen vornehmen muss – wird er sich zurechtfinden?
* Erscheint meine Software aus "einem Guss" (konsequentes Anwenden von Kommentierungen, Einrückungen, Variablennamen...)?

**Software-Aktualität**

Wird eine Software erstellt oder geändert, so muss diese wie auch alle Dokumente und andere Unterlagen auf dem Server gespeichert werden. Nach einer eventuellen Software-Änderung ausser Hause und muss diese Software sofort bei der Rückkehr (und nicht erst Stunden später oder nie) auf den gemeinsamen Server.

**Firmen-Name, Firmen-Logo**

Wird in einer Software der Firmen-Name irgendwo verwendet (innerhalb eines Hilfe-Fensters o. ä.) so lautet dieser

TEAM ELECTRONICS

und nicht etwa "TEAM Electronics" oder "Team Electronics" oder gar "TEAM Elektronic".

Das Firmenlogo findet sich im Dokument D0393 – bitte keine Eigenkreationen ersinnen.

**Sprache**

Grundsätzlich orientieren sich viele Variablennamen an der englischen Sprache und die Kommentare sind in Deutsch verfasst, sofern nicht Kundenanforderungen dies verhindern.

Wie in der Programmierung üblich, ergibt sich oft ein Mix aus deutsch/englisch, was aber im Rahmen der natürlichen Empfindung zulässig ist. Es gelten allgemein die Regeln der neuen deutschen Rechtschreibung.

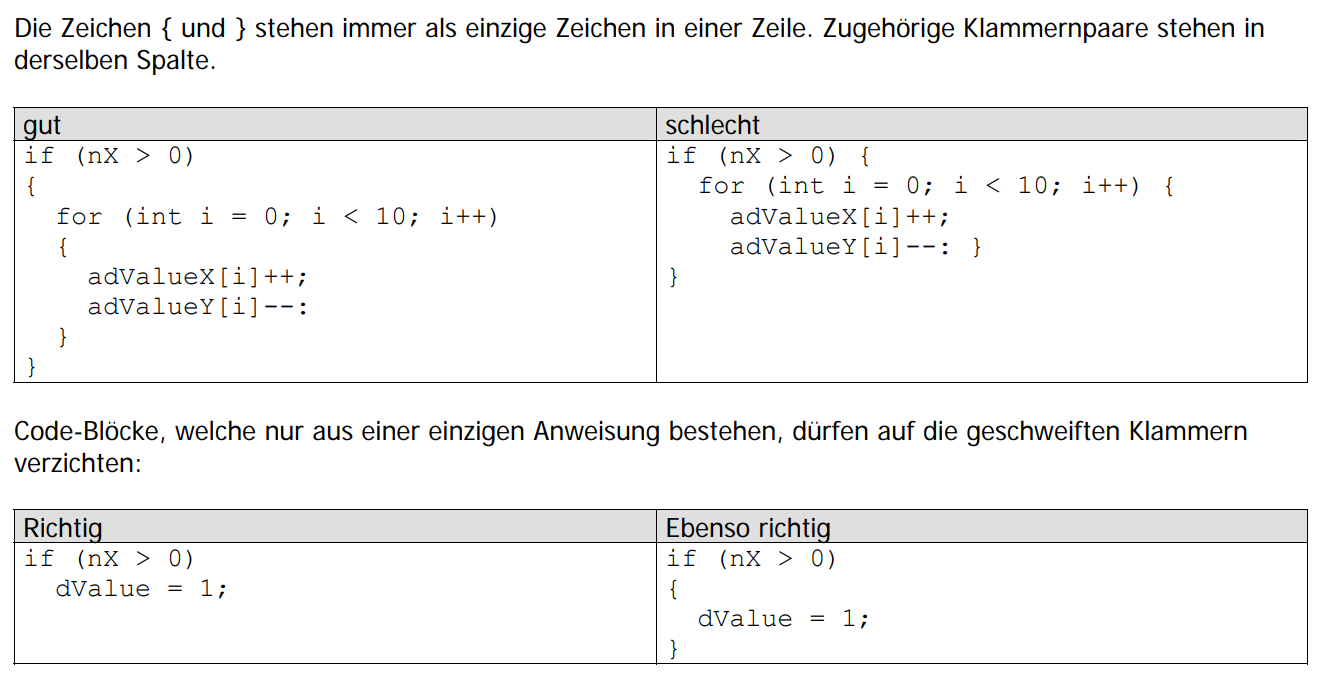
**Dokumentation**

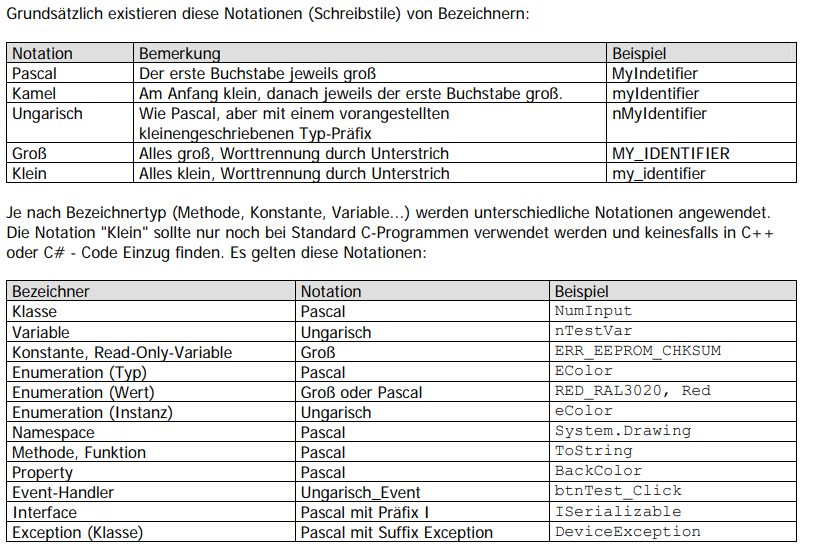
Lästig, aber notwendig: Inbetriebnahmeanleitungen, Beschreibung von Funktionsbibilotheken,

Bedienungsanleitungen, Berechnungs-Dokumentationen (Übersetzungsverhältnisse...) usw.

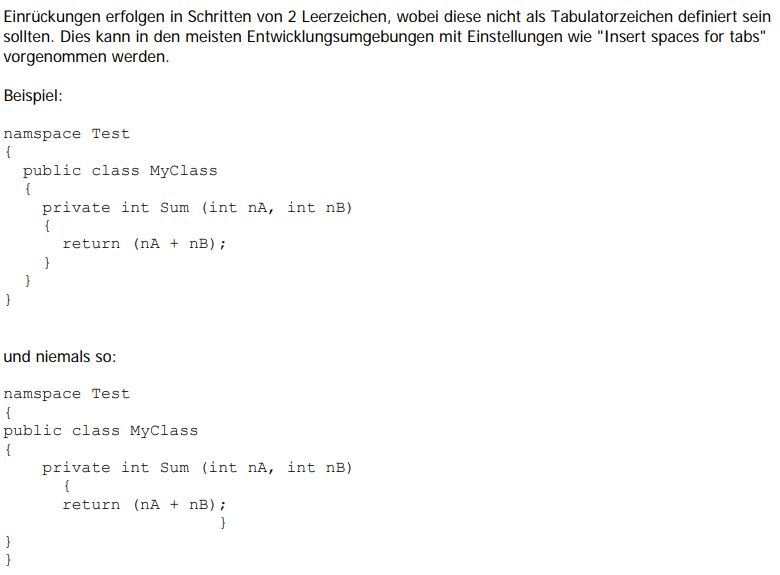
Verweise auf diese Dokumente finden sich im jeweiligen Info-File.

Programmierrichtlinien:

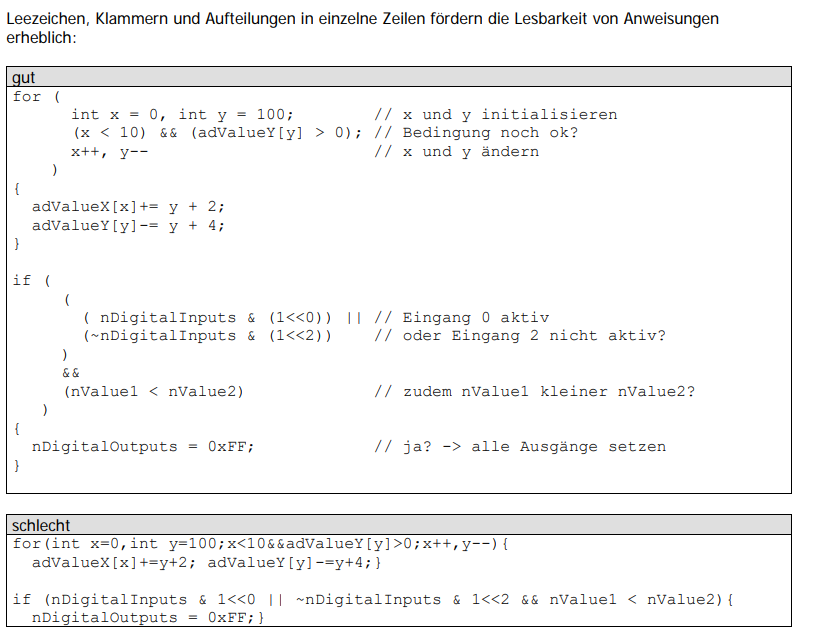
**Verwendung der geschweiften Klammern**

**Notationen**:

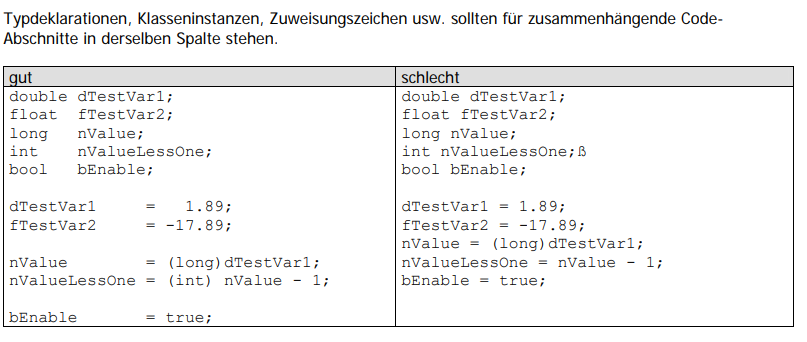
**Einrückungen:**



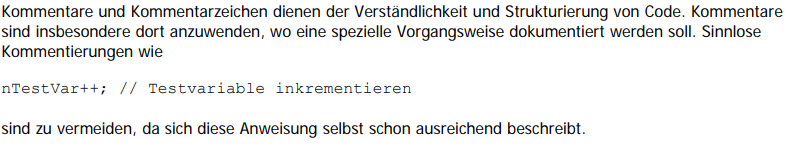
**Verwendung von Leerzeichen und Klammern:**



**Deklarationen und Zuweisungszeichen in derselben Spalte:**



**Kommentare:**



# Testfälle

# Installationsanleitung

# Reflexion

In diesem Abschnitt meiner Dokumentation finden Sie meine persönliche Meinung zu dem Projekt. Sie erfahren, ob ich selber damit zufrieden bin und was ich von der Idee halte.

## Positives

Ich finde die Punkte, welche ich eingebaut habe sind mir gelungen und mit dem was ich gemacht habe bin ich zufrieden. Auch die Idee eine öffentliche API zu verwenden finde ich eine sehr spannende und interessante Idee.

## Negatives

Mit was ich nicht so zufrieden bin ist, dass ich nicht so viele der Punkte erfüllen konnte. Jedoch ist das bereits alles.

# Anderes