# Lab Report <Template>

Abgabe <01>

<Jan> <Mühlbacher>

<Sebastian> <Toporsch>

<Tobias> <Weigl>

Repository: <URL>

## Vorgehen

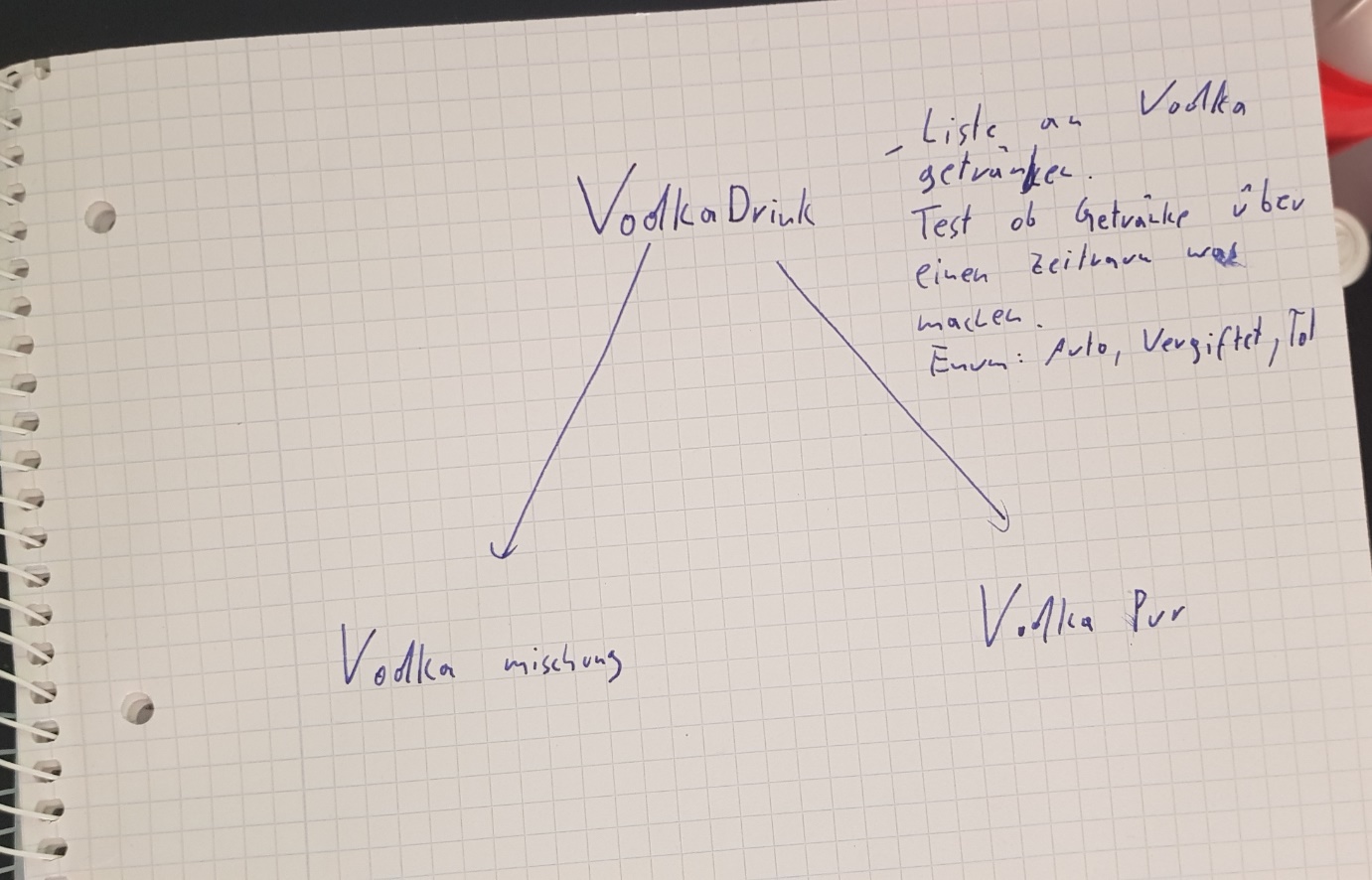
Notieren Sie Ihr Vorgehen in den Übungen, dokumentieren Sie fortlaufend wichtige Schritte, erstellen Sie Screenshots (wenn hilfreich), dokumentieren Sie für Sie wichtige Erkenntnisse, usw.

Übung 1 (19.03.2020):

Als erstes wurde besprochen welche Aufgaben jeder übernehmen soll. Wir einigten uns darauf, dass Tobias Weigl und Jan Mühlbacher sich um die Klassen und Unterklassen kümmern und Sebastian Toporsch die README und die Tests schreibt. Danach wurde besprochen wie

#### Detailliertes Vorgehen Tobias Weigl:

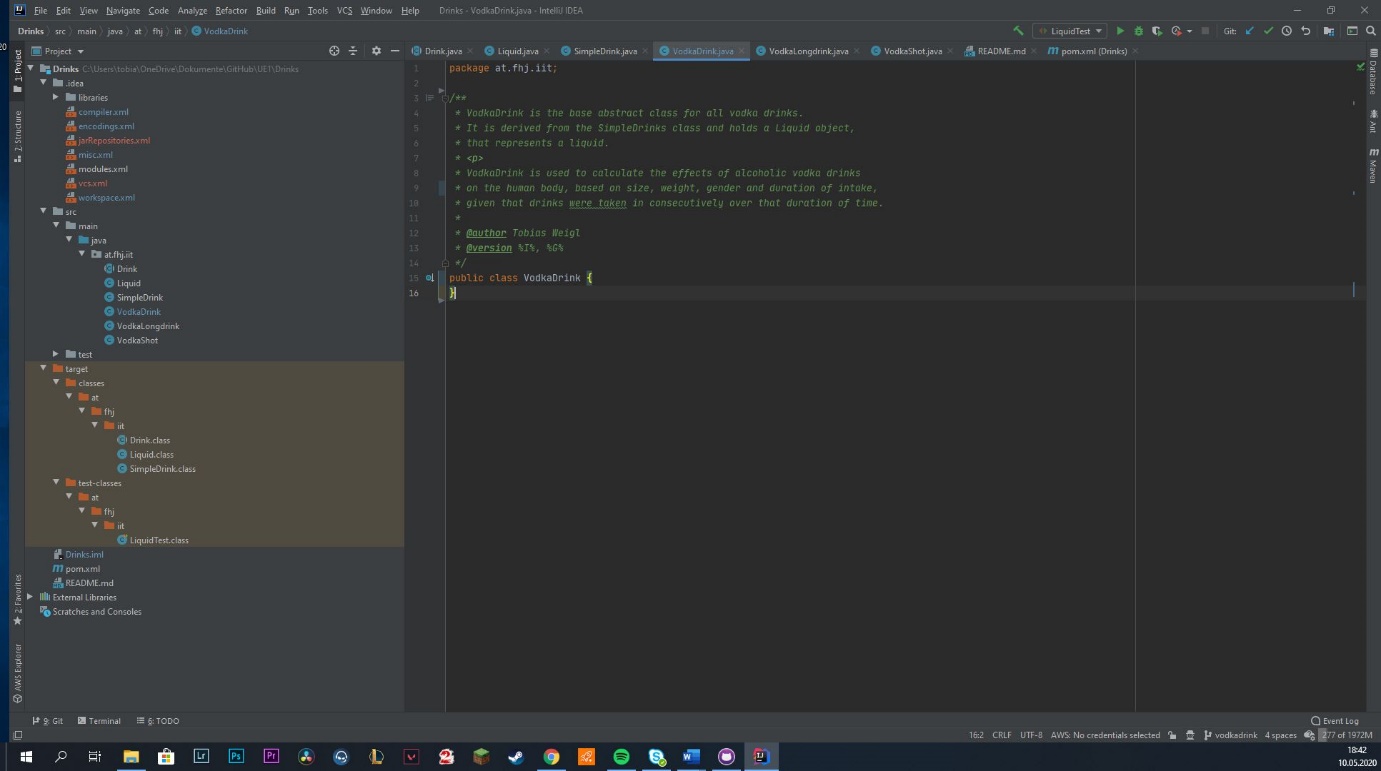
Nachdem das GitHub Repository von Herrn Toporsch angelegt wurde, habe ich zuerst die Repository lokal auf meinen Rechner gecloned und dann angefangen meine Subklasse mitsamt der zwei Subklassen dieser Klasse zu konzeptionieren. Ich habe mich hier für Vodka basierte Getränke entschieden. Zuerst habe ich auf Papier ein ganz simples Diagramm erstellt um die Klassenvererbung zu visualisieren und um festzustellen, welche Klassen überhaupt erzeugt werden müssen. Als Überklasse habe ich „VodkaDrink“ gewählt und als Subklassen „Vodka mischungen“ und „Vodka Pur“. Um das Ganze ins Englische zu bringen habe ich die Subklassen im Nachhinein in „VodkaLongdrink“ und „VodkaShot“ umbenannt.



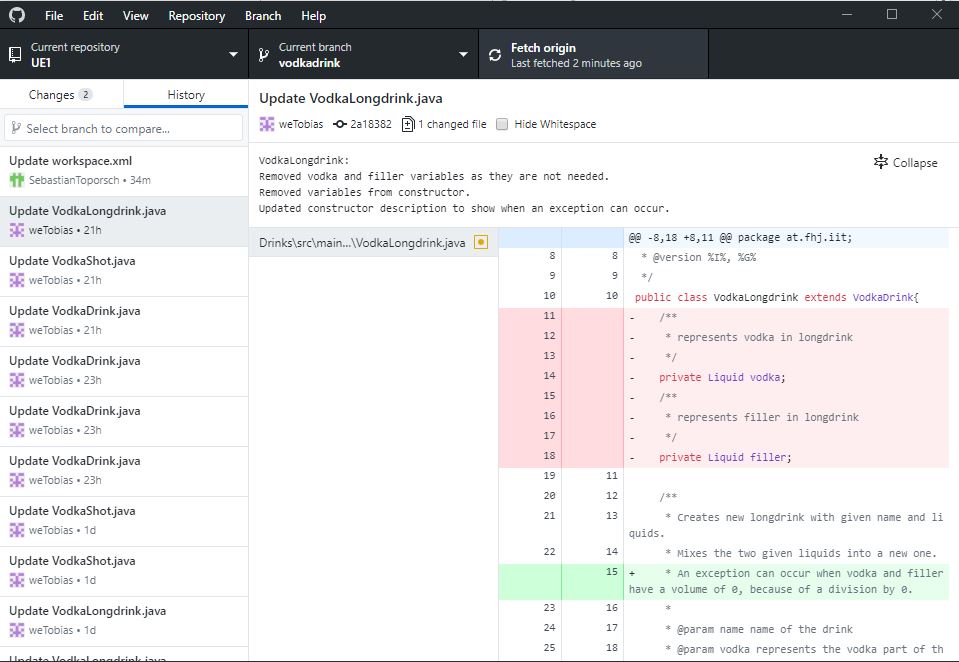
„VodkaDrink“ sollte in diesem Konstrukt einen Sammelbegriff für alle Drinks darstellen, die Vodka beinhalten. „VodkaLongdrink“ sollte dagegen ein Mischgetränk aus Vodka und einem alkoholfreien Getränk und „VodkaShot“ puren Vodka darstellen.  
Als nächsten Schritt habe ich mir überlegt, was dieses Konstrukt als Funktionalität mitbringen sollte und bin schlussendlich auf die Idee gekommen, einen Blutalkoholgehalt Rechner zu programmieren, oder besser gesagt einen Rechner, der dir basierend auf deinem Blutalkoholgehalt deinen aktuellen Körperstatus zurückgeben soll. Dies soll verwirklicht werden, indem man durch eine Methode in der „VodkaDrink“ Klasse eine Liste aus „VodkaDrink“ Objekten einliest, und daraus basierend auf Geschlecht und Gewicht der zu berechnenden Person, sowie der Zeit der Einnahme einen Blutalkoholgehalt errechnet und dann einen Körper Status zurück gibt.  
Die Rückgabe war zuerst als Enum konzipiert, ich habe mich letzten Endes aber für einen String entschieden.

Nach der Konzeption habe ich eine kurze Zusammenfassung meines Konzepts geschrieben und als „Description\_VodkaDrinks.txt“ auf die GitHub Repository geladen, damit Herr Toporsch sie in die „readme.md“ implementieren kann.  
Im Anschluss habe ich in der Repository eine neue Branch namens „vodkadrink“ geöffnet, in der ich von nun an meine updates gepushed habe um sie zum Schluss mit der master Branch zu vereinigen.

Jetzt habe ich das Projekt in IntelliJ importiert indem ich ein neues Projekt aus bestehenden Quellen erstellt habe und hierfür die „pom.xml“ angewählt habe, um es auch wirklich als Maven Projekt zu importieren. Angefangen habe ich die Implementierung meines Konzeptes, indem ich zuerst für alle Klassen festgestellt habe, welche Methoden und Konstruktoren sie benötigen. Diese habe ich dann als Signatur implementiert und mithilfe von JavaDoc beschrieben.



Als Solches habe ich die Klassen auch das erste Mal auf die GitHub Repository gepushed. Danach habe ich die Methoden und Konstruktoren implementiert, damit sie ihre Funktionalität erhalten. Im Verlauf des Programmierens haben sich manche Ideen leicht abgeändert und es wurden Bearbeitungen durchgeführt, diese sind aber immer einher gegangen mit einer Bearbeitung der Beschreibung, sodass immer klar ist, was der Code gerade macht. Die Änderungen habe ich regelmäßig auf die Repository gepushed und mit Nachrichten versehen, die beschreiben was geändert wurde.



## Links / Literatur

Dokumentieren Sie hilfreiche Links zu Onlineliteratur bzw. welche Literatur Sie zur Erarbeitung der Abgabe/Übung verwendet haben.

Apache Maven Standard Directory Layout:

<https://www.baeldung.com/maven-directory-structure>

Markdown Syntax:

<https://www.markdownguide.org/basic-syntax/>

* Adaptieren Sie das Template für Ihre Bedürfnisse.  
  Beachten Sie dabei folgende Angaben.
* Entfernen Sie diese Checkliste
* Verwenden Sie folgendes Schema zum Speichern Ihres Lab Reports  
  <JAHR>-Abgabe<Nr>-<Nachname1>-<Nachname2>.pdf  
  **zB: 19-Abgabe01-Ulm-Hofer.pdf**
* Verwenden Sie dieselbe Namenskonvention für Ihre Abgaben bzw. das Repository  
  zB: https://<url-git-server>/19-Abgabe01-Ulm-Hofer/
* Min. 1 A4 für das jeweilige Kapitel in der Lehrveranstaltung, Dokumentieren Sie in einem sinnvollen Maß sodass Sie und StudienkollegInnen Ihre Dokumentation nachvollziehen können!