

Studiengang: Informatik (Bachelor of Science)



Transfer-Projekt

Erstellung eines PoS-Systems zur Berechnung und Erfassung von Verkäufen mit Java

Vorgelegt von: Marcello Dorschu
 Matrikel Nr. 653222017
 Mirco Franzkowiak
 Matrikel Nr. 653222018

Dozent: Dr. Susanne Rosenthal

Wintersemester 2022/23

1 Projektbeschreibung

1.1 Ziele des Projekts

Das Projekt hat zum Ziel eine Point-of-Sale- Anwendung zu erstellen. Zum Termin der Abgabe soll ein funktionsfähiger Prototyp bereitstehen, der folgenden Funktionsumfang haben soll:

- Eingabe von Produkten (per GUI) die in einer Datenbank abgerufen werden
- Zusammenstellung der angeforderten Produkte in Anzahl und Kosten
- Warenmanagement, das im Hintergrund ablaufen soll
- Ausgabe und Speicherung des Verkaufs zur Wiedervorlage

1.2 Voraussetzungen

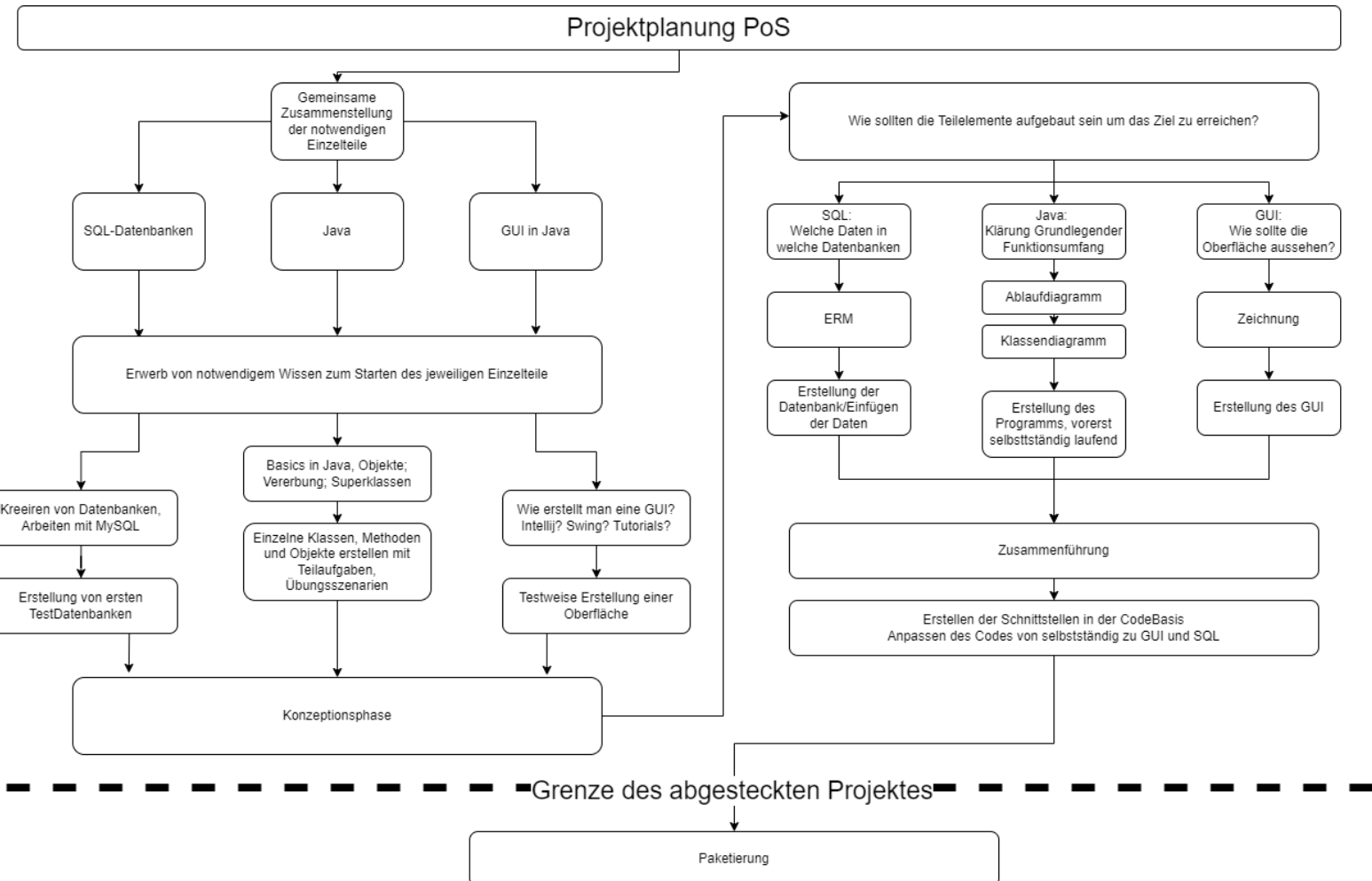
Um das Ziel zu erreichen Bedarf es vorrangig einen Wissenserwerb zu den Themen Java, SQL und GUI Erstellung in Java.

Weitere Voraussetzungen, die bereits erfüllt sind, sind ein funktionsfähiger Computer, die Möglichkeit Java Code zu schreiben, kompilieren und auszuführen. Ebenso bedarf es der Möglichkeit SQL-Datenbanken zu erstellen.

Innerhalb des Projektes wurde sich darauf geeinigt die IDEs IntelliJ & DataGrip von JetBrains zu nutzen.

2 Projektablauf

Das Projekt wurde zu Beginn wie folgend geplant:



Anhand des Diagramms lässt sich der Soll- und der Ist- Zustand sehr gut vergleichen.

Folgende Meilensteine lassen sich identifizieren:

- Planungsbeginn
- Grundlegender Wissenserwerb
- Konzeptionsphase
- Erstellung der einzelnen Bestandteile
- Anpassung und Zusammenführung der Bestandteile
- Paketierung

Der letzte Punkt ist nicht mehr Bestandteil des Projektes gewesen und liegt außerhalb des Fokus‘.

Der Soll-Zustand bei Abschluss des Projektes ist dementsprechend ein erstellter Prototyp, bei diesem die Teile GUI und Datenbank an eine Code-Basis angeschlossen sind und nutzbar ist.

Das entspricht nicht dem Ist-Zustand. Dieser sieht derzeit wie folgt aus:

- Das Programm selbst auf Basis von Java ist in sich selbst ausführbar und erwartet Anpassungen zur Zusammenführung mit der Datenbank und anschließender Zusammenführung mit der GUI
- GUI als Prototyp erstellt, Muss noch an interne Vorgabe angepasst werden
- Datenbank als Prototyp erstellt, Fehler bezüglich der Anbindung an Java Code. Mehr Wissen notwendig

Der folgende Meilenstein ist dem entsprechend der Zusammenführung der Projektbestandteile. Bei einem von 5 Meilensteinen macht das 20% des Gesamtprojektes aus. Da die Teile GUI und Datenbank in Ihrer Basis schon erstellt aber noch nicht die Anforderungen für die Anbindung erfüllen, sind das weitere 5% die noch nicht beendet sind.

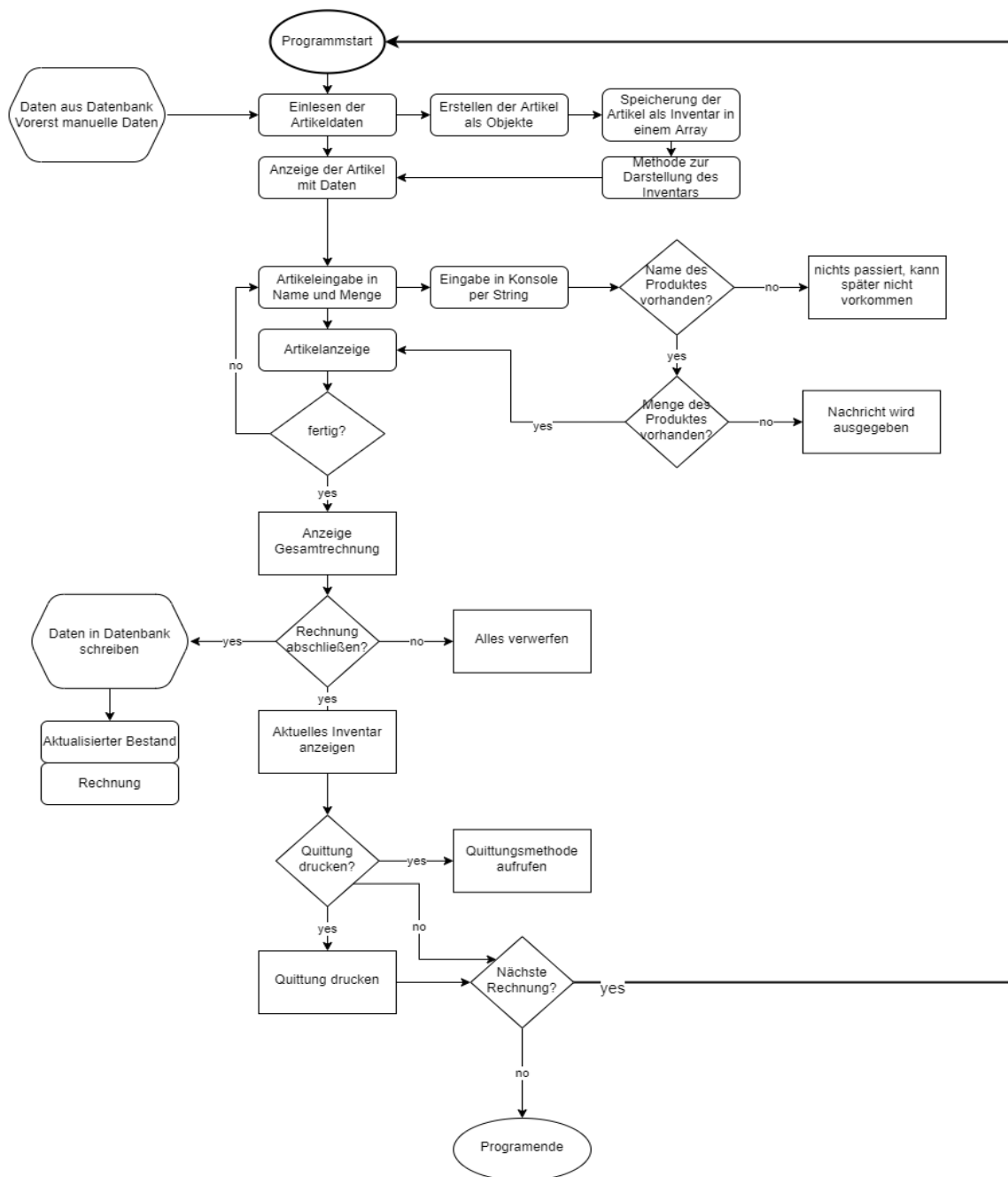
Geschätzt ist das Projekt zu 75% beendet.

3 Realisierung

Wie bereits unter 1.1 erwähnt, besteht der PoS im Grundlegenden aus 3 Komponenten:

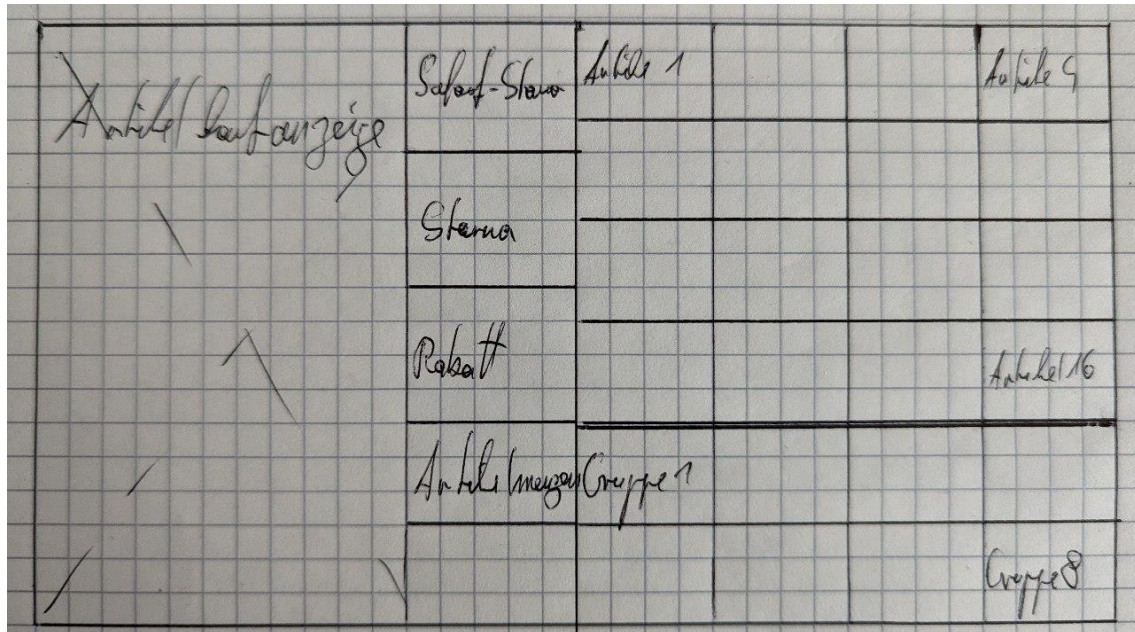
- Java Code Basis
- Java Swing GUI
- MySQL Datenbank

In der Konzeptionsphase sind folgende Grafiken entstanden, die den Aufbau des jeweiligen Systems widerspiegeln:



In diesem Ablaufdiagramm fehlt die Anbindung an die GUI.

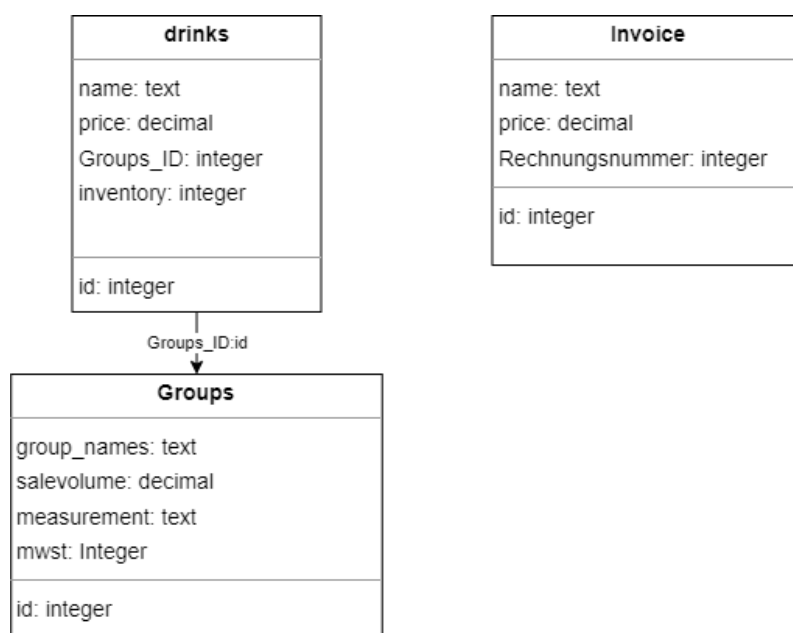
Im Weiteren wurde eine konzeptionelle Grafik für das GUI erstellt:



Diese soll im späteren Verlauf beim Start des Programms noch ein eigenes Ablaufdiagramm bekommen, da sie die Daten anders bereitgestellt bekommen muss. Letztendlich ersetzen die Buttons aber die Eingabe in die Konsole. Eine zusätzliche, aktuelle Ausgabe in das GUI muss noch realisiert werden ebenso wie die Funktion der Gruppenbuttons.

Die Funktionen der Buttons „Sofort-Storno“, „Storno“, „Rabatt“ und „Artikelmenge“ stehen derzeit nicht im Fokus des Projektes.

Die Datenbank wurde wie folgt geplant:



Um die Datenbanken klein zu halten, wurden mehrere Datenbanken konzeptioniert die hierarchisch voneinander abhängen.

Die Tabelle Drinks beinhaltet den Namen (z.B.: Wasser, Caipirinha), Preis und die Menge im Inventar als eigene Variablen. Groups_ID verweist auf den PK der Tabelle Groups.

Die Tabelle Groups beinhaltet den Gruppennamen (z.B.: Alkoholfrei), die Verkaufsmenge (Salesvolume; z.B.: 0,2), measurement (Einheit) und die Mehrwertsteuer als mwst.

Eine Paketierung der Software steht außerhalb des Fokus des Projekts und daher ist noch keine Installation der fertigen Software geplant. Daher ist zur Ausführung der Software erstmal eine IDE notwendig, die Java Swing unterstützt und idealerweise eine direkte Datenbank Anbindung ermöglicht. IntelliJ von JetBrains ermöglicht das.

4 Qualitätssicherung

Die Qualität des Programms lässt sich vorrangig an der Korrekten Ausführbarkeit des Programmes messen. Im Weiteren an der Geschwindigkeit der Vorgänge, die später insbesondere an der korrekten Arbeit mit der Datenbank hängt.

Da das Programm noch nicht vollständig erstellt ist, lehnt sich die Qualitätssicherung an der Konzeption an. Insofern sind die Bedingungen weitestgehend erfüllt und müssen im Nachhinein noch für die Zusammenführung angepasst werden.

Ist das Programm fertig erstellt, wird die Qualität durch ausgiebiges Testen sichergestellt. Das sollte Testfälle, die der normalen Anwendung entsprechen, als auch Sonderfälle beinhalten und werden im Rahmen der Qualitätssicherung dokumentiert.

5 Projektabschluss

Das Projekt wurde noch nicht zufriedenstellend beendet und es Bedarf noch einigen Anpassungen, wie bereits erläutert.

Die Zielerreichung gemessen an der bisher erarbeiteten Qualität ist zufriedenstellend, auch wenn das Projekt erst zu 75% fertig ist (vgl. 2.1.).

Das Ziel selbst ist jedoch nicht erreicht.

Schwierigkeiten bestanden unter anderem in dem Prozess des Lernens, da dieser schwer mit einer festen Zeit zu belegen ist wie lange der Prozess andauert, um das Grundwissen für die Anwendungsfälle sich anzueignen.

Weiter müssen größere Recherchen während der Programmerstellung angestellt werden, um spezifische Funktionen zu implementieren. Diese nehmen ebenfalls häufig mehr Zeit ein. Diese bestehen zu hohem Grad aus einer Schleife, an der zuerst nach einer Lösung gesucht wird, dann der Versuch der Implementation stattfindet und sich so lange wiederholt, bis die gewünschte Funktionalität erreicht ist. Ebenfalls ist die Qualitätssicherung während der Arbeit am Produkt ein weiterer Faktor mit einer weitgehend unbekannten Zeit.

Diese Szenarien sollen hier bei erwähnt werden:

- Nach der Aufsplittung eines Input-Strings bricht das Programm ab, da das Wort ‚fertig‘ in einem String Array nur den Index 0 hat, eine Funktion aber auf Index 1 zugreift - gelöst
- Das Programm geht nach der Eingabe des Wortes „fertig“ nicht in den nächsten Zustand über - gelöst
- Das Erstellen eines Sicherheitsarrays („tempDrinks“) das selbst Objekte beinhaltet lässt sich über Java eigene Mittel nicht realisieren, da eine Kopie selbst nur Referenzen zu den Objekten beinhaltet, welche selbst angefasst werden in der Funktion decreaseStock(). – ungelöst
- Das Java GUI wurde so weit mit Funktionalitäten beschmückt. Einzelne Buttons, welche auf die Datenbank zugreifen sollen, geben wiederholt „NullPointerException“ -> Die Datenbank Connection funktioniert nicht. – ungelöst
- Optimierung des Java GUI Interface entspricht nicht 1:1 unserer Vorstellung - ungelöst

Weiterhin war eine große Schwierigkeit mit Optimierungspotenzial das Zeitmanagement, welches durch die duale Natur des Studiums zusätzlich erschwert worden ist.

Mögliche Lösungsansätze wären in den meisten Bereichen im Zeitmanagement anzu-
setzen. Dazu gehören:

- Feste Zeitspannen für Probleme festlegen nach dieser Alternativen in Betracht
gezogen werden
- Bessere Teamarbeit durch mehr Austausch in Regelterminen
- Bessere Planung der Programmierarbeit in Klassen vor Beginn des Schreibens

6 Dokumentation (Anhänge)

Dieses Dokument kommt mit allen erarbeiteten Dateien der Projektarbeit. Die von uns und nicht von der IDE eigenständig erzeugten Dateien sind wie folgt:

Unter PoS_Java:

- Alkoholfrei.java (konkrete Klasse für Alkoholfreie Getränke)
- Items.java (abstrakte Klasse für alle Artikel)
- PoS_Main.java (Hauptklasse des Programms mit der Hauptfunktion des Durchlaufs des Programms in der Konsole und den ausgelagerten Methoden:
 - o stringToChar (nimmt einen String und wandelt den ersten Buchstaben in einen lowercase char um)
 - o addDrinks (fügt Drinks dem Inventar hinzu, solange Platz darin ist),
 - o printInventory (mit Hilfe der toString Methode der jeweiligen Objekte druckt es das Inventar in der Konsole)
 - o printReceipt (druckt die Quittung in die Konsole auf Basis der Variablen des letzten Durchlaufs aus)
- Test.java (Klasse um einzelne Aspekte vor Implementierung zu testen)
- Wein.java (konkrete Klasse für die Gruppe „Wein“ Getränke)

Unter POS:

- POS_GUI (der Code für die GUI)
 - o Erstellt das Interface der GUI (Layout, Buttons..)
- Class Database (versucht die Connection zur Datenbank aufzubauen)
- getPrice (den Preis des ausgewählten Drinks aus der Datenbank abrufen mit executeQuery)
- getAllDrinks (Alle Drinks aus der Datenbank abrufen und die Liste drinks mit den Namen füllen – wird am Ende ausgegeben)
- getDrinksInGroup (Alle Drinks aus der Datenbank, welche einer bestimmten Gruppe angehören abrufen und ebenfalls in einer Liste abspeichern, dann in ein Array umwandeln und zurückgeben)

- actionPerformed (Gibt den einzelnen Buttons ihre Funktion):
 - Storno: Setzt den Gesamtbetrag auf 0,00€ und ,clear' die Liste
 - Discount: Gibt einen Discount von 10% auf den Gesamtbetrag
 - Sofort-Storno: Löscht das Letzte Element aus der Liste und vom Gesamtbetrag
 - Artikelmenge: Gibt alle Drinks aus der Datenbank aus
 - Gruppe *: Gibt die Drinks der speziellen Gruppe aus

- Sqlite.jar (JDBC Driver für die Datenbank Connection)

Unter PoS_SQL

- Identifier.sqlite (Die PoS Datenbank):
 - Drinks (ID, Name, Preis)
 - Gruppen (GruppenID (alcohol, not-alcohol), salevolume (0,2l / 0,3l))

7 Quellenangaben

[SQLite | IntelliJ IDEA Documentation \(jetbrains.com\)](#) - SQLite Datenbank mit IntelliJ verbinden

[Java Connect to SQLite Database Example - YouTube](#) - SQLite und IntelliJ Verbinden mit JDBC

[Lerne SQLite in 60 Minuten // SQLite Tutorial Deutsch - YouTube](#) - SQLite Tutorial

[Java GUI: Full Course 🍷 \(FREE\) - YouTube](#) - Java Swing Tutorial

<https://www.techiedelight.com/copy-object-array-java/> - Erörterung von Methoden zur Kopie von Arrays mit Objekten

<https://www.baeldung.com/string/split> - Split Methode zur Verwendung der .split Methode für Strings

<https://greenteapress.com/thinkjava6/html/index.html> - Think Java: How to Think Like a Computer Scientist

Grundlagen in Java vermittelt durch Mauer, Johannes. Modul Programmieren I