Dokumentation

Präsentationsprüfung Informatik— Eine datenbankgestützte Mitglieder-verwaltung

**Ausführliche Gliederung des Vortrags**

**1. Einleitung**

1.1 Begrüßung, Vorstellung des Themas  
1.2 Überblick über die Präsentationsstruktur (kurzer Fahrplan/Ablauf)

**2. Demonstration des lauffähigen Programms**

2.1 Start der Anwendung (lokale MySQL-Datenbank läuft schon, JDBC-Verbindung)  
2.2 Vorstellung grundlegender Funktionen:  
 2.2.1 Hinzufügen eines neuen Mitglieds

 2.2.2 Suche nach Mitgliedern (mittels ID und Namen)  
 2.2.3 Bearbeiten und Löschen eines bestehenden Mitglieds   
 2.2.4 Anzeige aller Mitglieder mit zugehöriger Mitgliedsart und allen weiteren Informationen

**3. Aufbau und Architektur der Anwendung**

3.1 Übersicht über die dreischichtige Architektur (Packages: model, db, app)

3.2 UML-Klassendiagramm  
 3.2.1 Model-Layer: Klassen Member, MembershipType  
 3.2.2 DB-Layer: Klasse DatabaseManager (CRUD, JOIN-Abfragen, JDBC-Logik)  
 3.2.3 App-Layer: Klassen Main (Einstiegspunkt) und MemberManager (Benutzerinteraktion, Validierung)  
3.3 Datenbankmodell (kurze Vorstellung ER-Diagramm, 2 Tabellen mit 1:n-Beziehung)  
**4. Erläuterung wichtiger Code-Abschnitte**

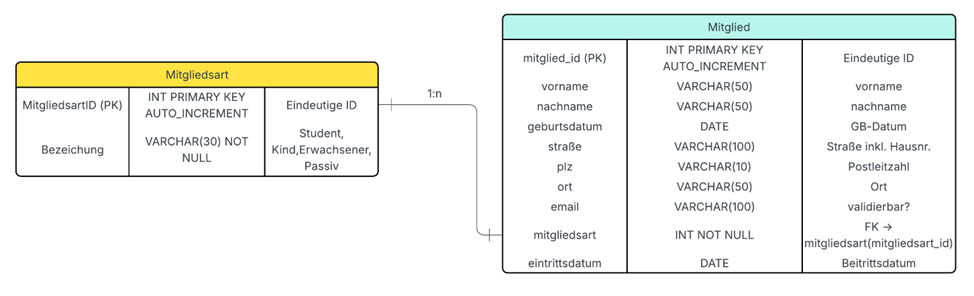
4.1 JDBC-Verbindung und Nutzung von PreparedStatements  
 4.1.1 Implementierung in DatabaseManager.connect() und DatabaseManager.addMember()  
 4.1.2 Sicherheit vor SQL-Injection

4.2 JOIN-Abfrage (Beispiel anhand getMemberById())  
 4.2.1 Durchführung und Erklärung des INNER JOIN, Objektmapping aus dem ResultSet  
4.3 Robuste Eingabevalidierung (MemberManager)  
 4.3.1 Validierung von Nutzereingaben (Datum, leere Felder, ID-Auswahl), Erklärung einiger implementierten Schleifen und Fehlerabfragen

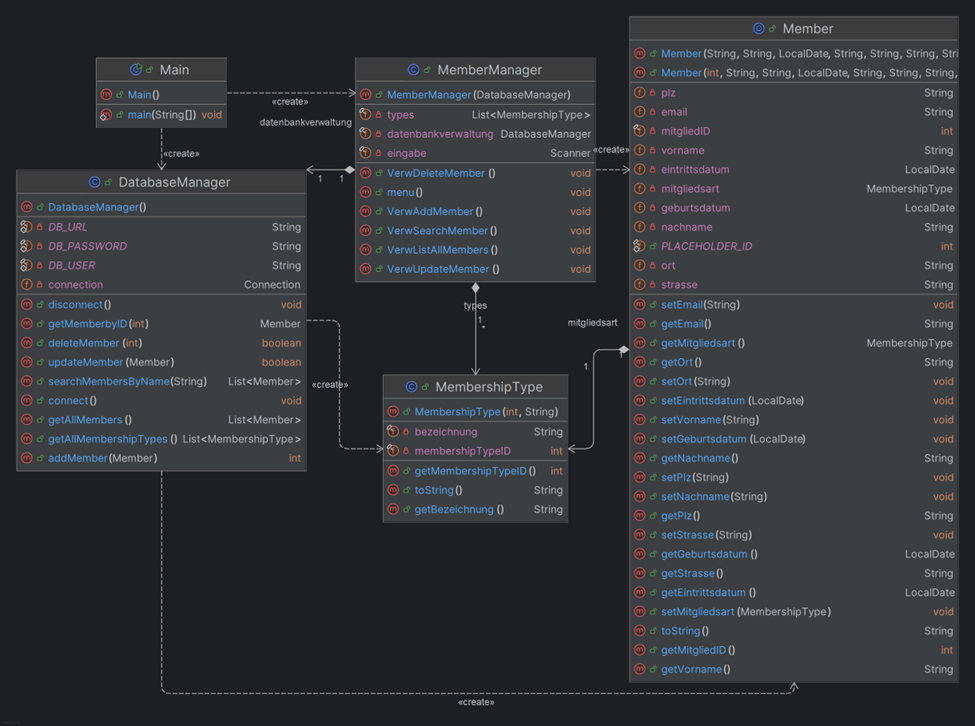
**5. Zusammenfassung und Fazit**

5.1 Zusammenfassung der wichtigsten technischen und methodischen Entscheidungen  
5.2 Bewertung des gewählten Ansatzes (lokale Datenbank, Konsolenanwendung, klare Strukturierung)

**ER-Diagramm**

****

**UML-Klassendiagramm**



**Quellenangaben**

6.1 Digitale Quellen

* The Java™ Tutorials:.
  + <https://docs.oracle.com/javase/tutorial/>
  + <https://docs.oracle.com/javase/tutorial/java/index.html>
  + <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>
* DEV.java Learn Java:
  + <https://dev.java/learn/intellij-idea/>
  + <https://dev.java/learn/oop/>
  + <https://dev.java/learn/language-basics/>
* <https://downloads.mysql.com/docs/connector-j-en.pdf>
* <https://www.marcobehler.com/guides/jdbc>
* <https://dzone.com/articles/jdbc-tutorial-part-2-running-sql-queries>
* <https://labex.io/tutorials/java-how-to-prompt-user-input-in-a-java-console-application-417592>
* <https://ioflood.com/blog/java-user-input/>
* <https://www.geeksforgeeks.org/java/>
* <https://zetcode.com/java/>

6.2 Analoge Quellen

* Java ist auch eine Insel – Christian Ullenboom (lässt sich teilweise auch im Internet abrufen)
  + Kapitel 21 (<https://openbook.rheinwerk-verlag.de/javainsel/21_001.html#u21>)

6.3 KI-Tools

ChatGPT (OpenAI) wurde genutzt zur

* Gliederungshilfe für Vortrag und Dokumentation
* Unterstützung bei Formulierungen und sprachlichem Feinschliff
* Finden von fachlichen Quellen zum Lernen/Recherchieren von Programmteilen (Java-Logik, SQL)

GitHub-Co-Pilot wurde in der IDE zur Vervollständigung von Code in der selben Zeile genutzt

**Eidesstattliche Versicherung über die Eigenständigkeit der Bearbeitung**

„Ich versichere, dass die Präsentation von mir selbstständig erarbeitet wurde und ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Diejenigen Teile der Präsentation, die anderen Werken im Wortlaut oder dem Sinn nach wissentlich entnommen wurden, sind als solche kenntlich gemacht.“

Habichtswald, 04.06.2025 Moritz Glaser

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ort, Datum) Unterschrift