Labor_07 4XHIF
JDBC Kurssystem

HTL ST. PÖLTEN

1. **JDBC - Kussystem**

Gegeben ist ein SQL-Skript (`kurssystem.sql`), das eine einfache Datenbank für die Verwaltung eines EDV-Kurssystems generiert. Es besteht aus folgenden Tabellen:

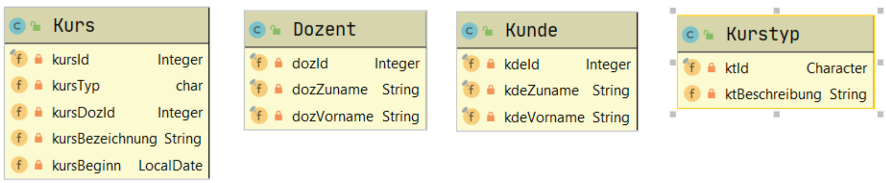
- Tabelle `dozent`
- Tabelle `kurs`
- Tabelle `kunde`
- Tabelle `kurs_kunde`

Lösen Sie folgende Aufgaben:

(a) Vorbereitende Arbeiten

- Ergänzen Sie das Skript für eine neue Tabelle `kurstyp`. Diese Tabelle enthält als Primarykey das Kurzzeichen des Kurstyps (ein Zeichen) und als weiteres Datenfeld die Langbezeichnung des Typs. Zur Zeit sind die Kurstypen `'P'` (Programmierung), `'S'` (Skriptspachen) und `'W'` (Webtechnologien) verwendet.
- Erzeugen Sie aus dem Skript eine PostgreSQL-Datenbank und adaptieren Sie das Skript so, dass es auch als INIT-Skript für eine H2-InMemory Datenbank für JUnit-Tests verwendet werden kann. Dabei ist der PostgreSQL Typ `SERIAL` für autogenerierte Pk's auf den Typ `IDENTITY` zu ändern.

(b) Entwickeln Sie entsprechende Geschäftsklassen `Kunde`, `Dozent`, `Kurs` und `Kurstyp` gemäß dem gegebenen UML-Diagramm.



Stellen Sie dabei sicher, dass eine ID ungleich `null` pro Instanz nur einmal vergeben werden kann und überschreiben Sie die Methoden `equals()` und `hashcode()` entsprechend.

(c) Gegeben ist weiter das Paket `persistence` mit entsprechende Interfaces für die diversen Repositories. Diese definieren grundlegende Datenbankoperationen. Entwickeln Sie entsprechende DB-Zugriffsklassen.

(d) Testen:
Die Implementierung ist mit JUnit-Tests gegen eine H2-InMemory-Datenbank zu testen. Gegeben ist eine Testkalsse `JdbcDozentRepositoryTest`. Entwickeln Sie Testklassen für die anderen Repositories, konkret `JdbcKundeRepository` und `JdbcKursRepository` und führen Sie die Tests durch.

2. **Näherungsweise PI-Berechnung**

Die näherungsweise Berechnung von π kann gemäß Herleitung von PI als Umfang eines einem Kreis mit Radius $r = 0.5$ eingeschriebenen regelmäßigen n-Ecks mit $\lim\limits_{n\to\infty} (n \cdot s_n)$ erfolgen. Dabei wird auf der Webseite die folgende Formel hergeleitet:

$$s_{2n} = \sqrt{0.5 \cdot \left(1 - \sqrt{1 - s_n^2}\right)}$$

Dabei kommt es bei immer kleiner werdendem s_n zu einer Subtraktionskathastrophe (vgl. Skriptum Seite 80ff). Eine zu obiger Formel äquivalente, bei der die Subtraktionskathastrophe verhindert wird, ist

$$s_{2n} = \frac{\sqrt{0.5} \cdot s_n}{\sqrt{1 + \sqrt{1 - s_n^2}}}$$

a) Zeige die mathematische Äquivalenz der beiden Formeln

b) Schreibe ein Javaprogramm, das beginnend mit $n = 6$, $s_6 = 0.5$ 25 Iterationen durchführt und den Umfang des entsprechenden n-Ecks mit beiden Formeln berechnet. Der jeweils letzte Näherungswert liegt also für $n = 100663296$ vor. Das Programm könnte etwa folgende Ausgabe liefern:

```
n =          6 pi: 3,00000000 pi: 3,00000000
n =         12 pi: 3,10582854 pi: 3,10582854
n =         24 pi: 3,13262861 pi: 3,13262861
n =         48 pi: 3,13935020 pi: 3,13935020
...
```