**Aufgaben „Datenbankanbindung“**

Zielsetzung der Aufgaben

Diese Aufgaben ermöglichen es, sich mit Datenbankanbindungen und den damit verbundenen Vorgängen in der Sprache Java und der IDE IntelliJ IDEA zu befassen, und darüber hinaus Handlungssicherheit im Umgang mit bereits Bekanntem zu gewinnen.

Alle Aufgaben werden in getrennten Projekten bearbeitet. Wiederholungen sollten nicht mittels Kopien gelöst werden, sondern jeweils einzeln geschrieben werden.

Vorbereitung

Legen Sie auf Ihrem lokalen PostgreSQL-Datenbankserver eine leere Datenbank namens "wetter" an. Diese Datenbank wird im Rahmen der Aufgabe 0 als "Spielwiese" dienen, um verschiedene Operationen durchzuführen.

Aufgabe 1: FakePersonenGenerator an der Datenbank

Legen Sie ein neues Projekt an. Binden Sie die aktuellen PostGreSQL-Treiber ein, und legen Sie mittels *pgAdminIII* eine Datenbank personen an. Legen Sie in IntelliJ eine Datenquelle für diese Datenbank an.

Kopieren Sie die Klassen Person, Names und eventuelle Generator-Klassen aus dem FakePersonGenerator-Projekt kopieren, so dass hier auch Personen-Objekte erzeugt werden können.

Versehen Sie die Person-Klasse mit Gettern für alle Attribute und die Kombination strasse + " "+hausnummer.

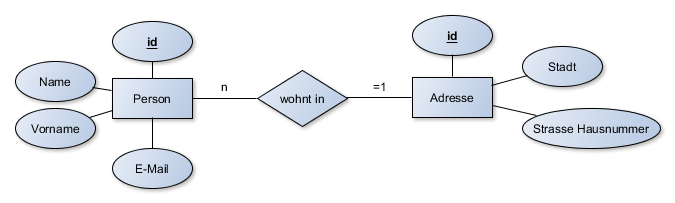
Schreiben Sie eine Klasse PersonDBConnector mit folgenden Methoden

* public PersonDBConnector(String url, String user, String pass) : Konstruktor, stellt die DB-Verbindung her (Fehlerbehandlung ausserhalb)
* public void dbAnlegen() : Anlegen der o.a. Tabellen in der Datenbank persondb mit vorherigem DROP IF EXISTS (Fehlerbehandlung ausserhalb)
* public void speichern( Person p ): das übergebene Person-Objekt p mittels PreparedStatement-Objektes in die Datenbank schreiben. Achten Sie auf die nötige Einfügereihenfolge! (Fehlerbehandlung selbst)
* private int adresseIdHolen(Person p) : holt die als Fremdschlüssel benötigte id der Adresse mittels SELECT (Fehlerbehandlung selbst)
* private int adresseSpeichern(Person p) : Speichern der Adresse in einer separaten, privaten Methode passieren, die id der Adresse zurückgibt – diese Methode muss entscheiden, ob sie eine Adresse neu einfügen muss, oder ob die Adresse bereits vorhanden ist. (Fehlerbehandlung selbst)
* public void disconnect() schlißt die DB-Verbindung (Fehlerbehandlung ausserhalb)

Der Konstruktor von PersonDBConnector soll die Datenbankverbindung herstellen (mit URL und Zugangsdaten aus Konstruktor-Parametern). Das entstehende Connection-Objekt muss in einem Attribut gespeichert werden.

Schreiben Sie ein Programm, das sich mittels eines PersonDBConnector-Objekts und mit dem Datenbankserver verbindet und die Tabellen gem. u.a. ER-Modell anlegt (dieses Modell ist nicht vollständig normalisiert, das soll aber so sein, damit die DB-Struktur nicht zu komplex wird).

Die ids sind dabei vom Typ SERIAL (vgl. PostGreSQL-Dokumentation, Kap. 8.1.4), alles andere vom Typ TEXT (ebd. Kap. 8.3.). Straße und Hausnummer werden vor dem Einfügen zu einem einzigen String verkettet.



Achten Sie darauf, die Fehlerbehandlung in der main()-Methode durchzuführen. Nutzen Sie finally (vgl. Exceptions-Kapitel) um sicherzustellen, dass die DB-Verbindung auf jeden Fall geschlossen wird.

Fügen Sie 10 000 Personen in die Datenbank ein und überzeugen Sie sich davon, dass die Datensätze tatsächlich in der Datenbank ankommen.