

**Übungsprotokoll**

**Prepared Statements**

**Informationssysteme**

**4CHITT 2015/16**

**Moritz Mühlehner, Lukas Sträßler**

|  |  |
| --- | --- |
|  | **Version 0.1** |
| **Note:** | **Begonnen am 18. Februar 2016** |
| **Betreuer: Borko Michael** | **Beendet am 24. Februar 2016** |

Inhaltsverzeichnis

[1 Einführung 3](#_Toc444015808)

[1.1 Ziele 3](#_Toc444015809)

[1.2 Aufgabenstellung 3](#_Toc444015810)

[1.3 Quellen 3](#_Toc444015811)

[2 Ergebnisse 4](#_Toc444015812)

[2.1 Property-Files 4](#_Toc444015813)

[3 Teamarbeit 4](#_Toc444015814)

# Einführung

PreparedStatements sind in JDBC eine Möglichkeit SQL-Befehle vorzubereiten um SQL-Injections zu vermeiden. Die Typüberprüfung kann somit schon bei der Hochsprache abgehandelt werden und kann so das DBMS entlasten und Fehler in der Businesslogic behandelbar machen.

## Ziele

Es ist erwünscht Konfigurationen nicht direkt im Sourcecode zu speichern, daher sollen Property-Files [3] zur Anwendung kommen bzw. CLI-Argumente (Library verwenden) [1,4] verwendet werden. Dabei können natürlich Default-Werte im Code abgelegt werden.  
Das Hauptaugenmerk in diesem Beispiel liegt auf der Verwendung von PreparedStatements [2]. Dabei sollen alle CRUD-Aktionen durchgeführt werden.

## Aufgabenstellung

Verwenden Sie Ihren Code aus der Aufgabenstellung "Simple JDBC Connection" um Zugriff auf die Postgresql Datenbank "Schokofabrik" zur Verfügung zu stellen. Dabei sollen die Befehle (CRUD) auf die Datenbank mittels PreparedStatements ausgeführt werden. Verwenden Sie mindestens 10000 Datensätze bei Ihren SQL-Befehlen. Diese können natürlich sinnfrei mittels geeigneten Methoden in Java erstellt werden.

Die Properties sollen dabei folgende Keys beinhalten: host, port, database, user, password

Vergessen Sie nicht auf die Meta-Regeln (Dokumentation, Jar-File, etc.)! Die Testfälle sind dabei zu ignorieren. Diese Aufgabe ist als Gruppenarbeit (2 Personen) zu lösen.

## Quellen

[1] Apache Commons CLI; Online: http://commons.apache.org/proper/commons-cli/  
[2] Java Tutorial JDBC "Prepared Statements"; Online: https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html  
[3] Java Tutorial Properties; Online: https://docs.oracle.com/javase/tutorial/essential/environment/properties.html  
[4] Overview of Java CLI Libraries; Online: http://stackoverflow.com/questions/1200054/java-library-for-parsing-command-line-parameters

# Ergebnisse

## Property-Files

Um Properties in Java zu verwenden gibt es ein eigenes Properties Objekt.  
Dazu muss java.util.Properties importiert werden.  
Die Properties Klasse beinhaltet mehrere wichtige Methoden wie z.B. setProperty(„key“,“value“), getProperty(„key“), remove(„key“), store(FileOutputStream, String comment) und load(FileInputStream).

Die Properties werden in dem File in dem Format key=value gespeichert.

# Teamarbeit

Der tatsächliche Aufwand ist wie folgt aufgeschlüsselt:

**Moritz Mühlehner:**

|  |  |  |
| --- | --- | --- |
| **Durchgeführte Arbeit** | **Datum** | **tatsächlicher Aufwand** |
| Übungsdurchführung | 19.02.2016 | ? h |
| Protokoll | 21.02.2016 | ? h |
| **Gesamt** | **19 - 21.02.2016** | **? h** |

**Lukas Sträßler:**

|  |  |  |
| --- | --- | --- |
| **Durchgeführte Arbeit** | **Datum** | **tatsächlicher Aufwand** |
| Übungsdurchführung | 19.02.2016 | ? h |
| Protokoll | 22.02.2016 | ? h |
| **Gesamt** | **19 - 22.02.2016** | **? h** |

**Gesamt:**

|  |  |
| --- | --- |
| **Durchführender** | **tatsächlicher Aufwand** |
| Lukas Sträßler | ? h |
| Moritz Mühlehner | ? h |
| **Gesamt** | **? h** |