

Prepared

~ . . .

# Inhaltsverzeichnis

Aufgabenstellung.....	2
Arbeitsaufteilung mit Aufwandschätzung.....	3
Endzeitaufteilung.....	4
Technologiebeschreibung.....	5
Arbeitsdurchführung.....	6
Quellenangaben.....	7

<b>Autor/en:</b>	<b>Özsoy Ahmet Tiryaki Seyyid</b>
<b>Klasse:</b>	4AHITM
<b>Letzte Änderung:</b>	04.05.2016
<b>Version:</b>	1.0

# Aufgabenstellung

## Einleitung

PreparedStatement 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 PreparedStatement [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 PreparedStatement 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.

## Quelle

[1] Apache Commons CLI; Online:

<http://commons.apache.org/proper/commons-cli/>

[2] Java Tutorial JDBC "PreparedStatement"; Online:

<https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html>

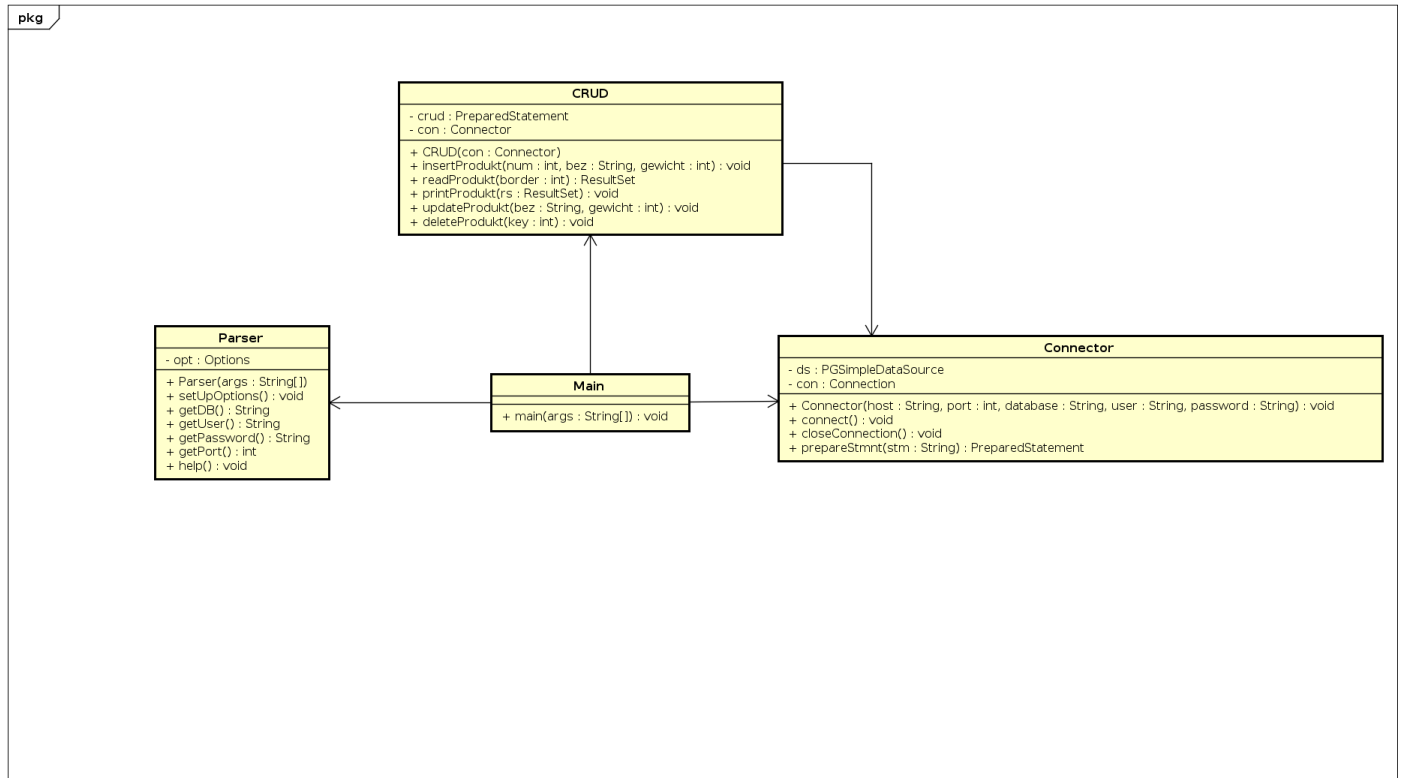
[3] Java Tutorial Properties; Online:

<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>

# Designüberlegung



## Arbeitsaufteilung mit Aufwandschätzung

Zuständige Person(en)	Task	Beschreibung	Geschätzte Zeit in h
Özsoy Ahmet	CRUD	PreparedStatements Aufrufe	1 h
Tiryaki Seyyid	Parser	CLI Argumente parsen	1 h
Tiryaki Seyyid, Özsoy Ahmet	Design	UML und Designüberlegung	3 h

Geschätzter Gesamtzeitaufwand

Person	Zeitaufwand in h
Özsoy Ahmet	2,5 h
Tiryaki Seyyid	2,5 h
Summe:	5 h

## Endzeitaufteilung

Zuständige Person(en)	Task	Geschätzte Zeit in h	Tatsächliche Zeit in h	Kommentar
Özsoy Ahmet	CRUD	1	1	
Tiryaki Seyyid	Parser	1	2	
Tiryaki Seyyid, Özsoy Ahmet	Designüberlegung	3	2	

Tatsächlicher Gesamtzeitaufwand

Person	Zeitaufwand in h
Özsoy Ahmet	2
Tiryaki Seyyid	3
Summe:	5

## Technologiebeschreibung

PreparedStatement - Postgres

Apache CLI

## Arbeitsdurchführung

Apache Commons CLI

Beispiel:

```
opt = new Options();  
opt.addOption(Option.builder("H").argName("hostname").desc("Adresse  
des Servers").hasArg().longOpt("host")  
    .numberOfArgs(1).build()));
```

Beispiel für eine getter-Methode:

```
public String getDB() {  
    if (cli.hasOption("d"))  
        return cli.getOptionValue('d');  
    else {  
        System.out.println("Datenbankname fehlt!");  
        printHelper();  
        System.exit(-1);  
        return "";  
    }  
}
```

```
}
```

## Prepared Statements(CRUD)

Beispiel:

```
public void insertProdukt(int num, String bez, int gewicht) {  
    //Erstellen eines Prepared Statements  
    crud = con.prepareStatement("INSERT INTO Produkt VALUES(?,?,?)");  
    try{  
        crud.setInt(1, num);  
        crud.setString(2,bez);  
        crud.setInt(3, gewicht);  
        crud.execute();  
    } catch(SQLException e) {  
        System.out.println("Fehler aufgetreten!");  
        System.out.println(e.getMessage());  
    }  
}
```

## Quellenangaben

[BOR] Titel: Michael Borko. (2016, März). Prepared Statements  
Autor: -  
Online/Quelle: <http://commons.apache.org/proper/commons-cli/>  
geändert am: -  
entnommen am: 09.03.2016

[BOR] Titel: *Michael Borko. (2016, März). Prepared Statements*  
Autor: -  
Online/Quelle:  
*<https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html>*  
geändert am: -  
entnommen am: 09.03.2016

[BOR] Titel: *Michael Borko. (2016, März). Prepared Statements*  
Autor: -  
Online/Quelle:  
*<https://docs.oracle.com/javase/tutorial/essential/environment/properties.html>*  
geändert am: -  
entnommen am: 09.03.2016

[BOR] Titel: *Michael Borko. (2016, März). Prepared Statements*  
Autor: -



---

Online/Quelle: <i><a href="http://stackoverflow.com/questions/1200054/java-library-for-parsing-command-line-parameters">http://stackoverflow.com/questions/1200054/java-library-for-parsing-command-line-parameters</a></i> geändert am: - entnommen am: 09.03.2016
---

---