
Betreutes Programmieren 11

- **Korrektur:** Dieses Aufgabenblatt wird nicht korrigiert, kann aber von einem Tutor in der unten genannten Betreuungszeit für einen Bonuspunkt abgenommen werden.
 - **Fragen:** Am Ausgabetag, jeweils mittwochs, können während der Betreuungszeit von 14:00 Uhr bis 17:15 Uhr in den CIP-Pools 1001N, 1002N und 1005N Fragen gestellt werden. Außerhalb dieser Zeit werden Fragen im entsprechenden Digicampus-Forum¹ beantwortet.
 - **Lösung:** Wird voraussichtlich eine Woche nach Ausgabe der Angabe hochgeladen.
-

Aufgabe 1 *** (*Produktverwaltung*)

In dieser Übungsaufgabe geht es darum, eine Java-Anwendung zu erstellen, die in der Lage ist, Daten aus einer MySQL-Datenbanktabelle auszulesen, sie mithilfe von Streams zu manipulieren und die Ergebnisse anzuzeigen. Die Anwendung soll auf eine vorhandene MySQL-Datenbank zugreifen, in der eine Tabelle namens "Product" mit den Spalten "id" (**int**), "name" (**varchar**) und "price" (**double**) vorhanden ist.

Durch die Bearbeitung dieser Übungsaufgabe können Sie Ihre Kenntnisse im Datenbankzugriff mit Java vertiefen und lernen, wie man Streams in diesem Kontext einsetzen kann.

Teilaufgabe a: Implementierung der ProductDAO-Klasse

Erstellen Sie eine Klasse namens `ProductDAO`, die für den Datenbankzugriff verantwortlich ist. Die Klasse soll im Konstruktor eine Verbindung (`Connection`) zur MySQL-Datenbank aufbauen und zwei Methoden bereitstellen:

- `public List<Product> getAllProducts()` – Gibt alle Produkte der Tabelle "Product" zurück. Verwenden Sie die SQL-Abfrage "SELECT * FROM Product".
- `public void updateProductPrice(int productId, double newPrice)` – Updatet den Preis eines Produkts der Tabelle "Product". Verwenden Sie ein `PreparedStatement` Statement mit der SQL-Abfrage "UPDATE Product SET price = ? WHERE id = ?".

Beachten Sie außerdem folgende Punkte:

- Behandeln Sie alle möglichen Exceptions innerhalb der zwei Methoden auf sinnvolle Art und Weise.
- Denken Sie daran, die Streams an geeigneten Stellen zu schließen.
- Geben Sie Erfolgsmeldungen aus, wenn die Operationen erfolgreich verliefen.

¹<https://digicampus.uni-augsburg.de/plugins.php/coreforum/index?cid=27109f15584353149791b88f594776b1>

-
- Die Klasse **Product** wird in Digicampus bereitgestellt. Laden Sie diese herunter und fügen Sie sie anschließend Ihrem Projekt hinzu.
 - Verwenden Sie die in der Vorlesung und im Skript bereitgestellten Zugangsdaten.

Teilaufgabe b: Implementierung der **ProductStatistics**-Klasse

Erstellen Sie eine Klasse namens **ProductDAO**, welche statistische Informationen zur Tabelle "Product" ermittelt. Die Klasse soll drei Methoden bereitstellen:

- `public static double calculateAveragePrice(List<Product> products)` – Berechnet den Durchschnittspreis aller Produkte.
- `public static List<Product> findMostExpensiveProducts(List<Product> products, int count)` – Ermittelt die teuersten Produkte basierend auf dem Preis. Der Parameter `count` stellt die Anzahl der Produkte dar, welche zurückgegeben werden sollen.
- `public static List<Product> findCheapestProducts(List<Product> products, int count)` – Ermittelt die günstigsten Produkte basierend auf dem Preis. Der Parameter `count` stellt die Anzahl der Produkte dar, welche zurückgegeben werden sollen.

Verwenden Sie für alle der drei Methoden keine Schleifen, sondern lediglich Streams. Machen Sie sich hierfür mit der `limit`-Streamoperation vertraut.

Teilaufgabe c: Implementierung der **Main**-Klasse

Erstellen Sie eine Klasse namens **ProductManager**, welche eine `main()`-Methode bereitstellt. In dieser Methode soll folgendes geschehen:

1. Es wird ein **ProductDAO**-Objekt erstellt.
2. Mit dem Objekt werden alle Produkte aus der Tabelle Product geholt.
3. Es wird der durchschnittliche Preis der Produkte mithilfe der **ProductStatistics**-Klasse ermittelt und ausgegeben.
4. Es werden die drei günstigsten Produkte mithilfe der **ProductStatistics**-Klasse ermittelt und ausgegeben.
5. Es werden die drei teuersten Produkte mithilfe der **ProductStatistics**-Klasse ermittelt und ausgegeben.
6. Durch die Verwendung von Benutzereingaben in der Konsole werden eine ID sowie ein Preis ermittelt.
7. Anschließend soll der Preis des Produkts mit der eingegebenen ID geupdatet werden.
Wichtig: Sie werden an dieser Stelle immer eine Exception bekommen, da Sie keine Rechte zur Manipulation von Daten in der Datenbank haben.
8. Im letzten Schritt werden ggf. Streams (z.B. bei der Verwendung der **Scanner**-Klasse) geschlossen.