```java
/**
package webapplication.datalayer;

import java.sql.ResultSet;

/**
 * @author Tobias
 *
 */
public class MySQLProduktBatchDAO implements ProduktBatchDAO {
    SQLMapper map = new SQLMapper();

    public MySQLProduktBatchDAO(){
        try { new Connector(); }
        catch (InstantiationException e) { e.printStackTrace(); }
        catch (IllegalAccessException e) { e.printStackTrace(); }
        catch (ClassNotFoundException e) { e.printStackTrace(); }
        catch (SQLException e) { e.printStackTrace(); }
    }

    @Override
    public ProduktBatchDTO getProduktBatch(int pbId) throws DALException {
        /*
         * We have imported our connector class. It's static,
         * so we can use the methods within it without having to create an instance of it.
         *
         * We can store the result of a query in the class ResultSet
         */
        String statement = map.getStatement("pb_SELECT");
        String[] values = new String[]{Integer.toString(pbId)};
        statement = map.insertValuesIntoString(statement, values);
        System.out.println("Query: "+statement);
        ResultSet rs = Connector.doQuery(statement);
        //Result is stored ^
        try {
            if (!rs.first()) throw new DALException("Produkt batch " + pbId + " findes ikke");
            /*
             * If no rows are returned,
             * that must mean that there is no batch with the given ID ^
             * We throw an exception because there is no object with the given ID.
             */
            return new ProduktBatchDTO (rs.getInt("pb_id"), rs.getInt("status"), rs.getInt("recept_id"));
            //If there is a result returned, then we create a new object from it. ^
        }
        catch (SQLException e) {throw new DALException(e); }
        //We also check for SQL exceptions ^
    }

    @Override
    public List<ProduktBatchDTO> getProduktBatchList() throws DALException {
        /*
         * We return a list of all the product batches.
         * Our query selects all present elements in the table.
         */
        List<ProduktBatchDTO> list = new ArrayList<ProduktBatchDTO>();
        ResultSet rs = Connector.doQuery(map.getStatement("pb_SELECT_ALL"));
        try
        {
            while (rs.next())
            {
                list.add(new ProduktBatchDTO (rs.getInt("pb_id"), rs.getInt("status"), rs.getInt("recept_id")));
            }
        }
        catch (SQLException e) { throw new DALException(e); }
        return list;
    }

    @Override
    public void createProduktBatch(ProduktBatchDTO produktbatch) throws DALException {
        String statement = map.getStatement("pb_INSERT");
        String[] values = new String[]{Integer.toString(produktbatch.getPbId()), Integer.toString(produktbatch.getStatus()), Integer.toString(produktbatch.getReceptId()) };
        statement = map.insertValuesIntoString(statement, values);
        System.out.println(statement);

        Connector.doUpdate(statement);
    }


    @Override
    public void updateProduktBatch(ProduktBatchDTO produktbatch) throws DALException {
        String statement = map.getStatement("pb_UPDATE");
        String[] values = new String[]{Integer.toString(produktbatch.getStatus()), Integer.toString(produktbatch.getReceptId()), Integer.toString(produktbatch.getPbId()) };
        statement = map.insertValuesIntoString(statement, values);
        System.out.println(statement);

        Connector.doUpdate(statement);
    }

}
```