Lab 2

In this lab you learn how to connect to a real database using Java.

In the previous lab you created a web application in which you could manage users and products. At that point you used an in-memory database: the data where stored in memory, using the Java class "`HashMap`". This is a good practice: it allows you to focus on your web application without having to worry about database-issues.
Nevertheless, if you actually want to start using your application, you will need a real database. You can connect to the PostgreSQL database that is available for you at `gegevensbanken.khleuven.be`.

Preparation

If you already have a database (from previous semesters), you can skip the preparation-steps and use this database for the next paragraphs.

Create a database schema for your project in the database of your class. Use your student number (like rxxxxxxx) as name for your schema. If necessary, follow these steps:

1. Connect with `gegevensbanken.khleuven.be`
   database: *administration*
   port: 5+abbreviation of the academic year (e.g. 51718 for 2017-2018)
   schema: user_administration
2. Execute: *select user_administration.set_session _svg('replaceWithDbName','replaceWithClass')*
   →e.g. *select user_administration.set_session_svg('webontwerp','2TX31')*

More information: https://kibo.ucll.be/projects/4testing/wiki/Databankgebruik

Tables

To create tables, etc. you can the desktop version of *PGAdmin* or the online version via *https://gegevensbanken.khleuven.be/*.

1. Create a table "`person`" in your schema.
2. In the table, create the necessary columns. Do not forget the primary key!
3. Add some test persons
4. Do the same for "`product`".
5. Grant access to `u0082726`, `u0015529` and `u0034562`
   - to your schema
   - to each table
   - to each sequence

Stories

See next page...

Persons – Very High Priority

Implement the stories:

1. StoryW08_PersonOverviewDesktop.
2. StoryW09_AddPersonDesktop.

Refactor

Review the code you wrote for story W08 and W09. The main method is only allowed to throw a **DbException**. Refactor your code if necessary:

1. In the package *db*, create the exception “`DbException`”. Make it an unchecked exception.
2. Catch all exceptions and throw them again as a “`DbException`”.