- **What is Objected-oriented programming? What concepts did you use in your solution? How did you use these concepts to develop your solution?**

Object oriented programming is a computer language created to best organize and utilize data. With this type of programing we are able to compartmentalize data into packages or sections, for easy use.

I experimented with a couple of different layouts with regards to the file structure. I divided the task into 3 sections. I wanted to isolate the run-file “game” and keep this file as simplistic as possible, as this is the only file the user have to directly interact with. The remaining sheets are divided into the packages Quiz which contain the two different games, and DBconfig for initial setup of the database.

Some of the techniques used in this task are:

- java.util.scanner, which was necessary in order to create a responsive interface for the quiz-participants.
- java.sql statements for in order to interact with the mySQL database.

- **What is JDBC? How did you use JDBC?**

JavaDataBaseConnectivity is an API for java, which enables connectivity between applications running java programming language (e.g IntelliJ) and database handling applications such as mySQL.

I downloaded JDBC from the mySQL-website (added a copy to the exam folder).

I used the JDBC run:

- Queries, to check if the typed answer was correct.
- Import statements to import new question rows from the table of both quiz programs.
- Insert statements to add users to the scoreTable.

- **Examples of OOP concepts that you have implemented in your QuizGame design.**

- Polymorphism in the quiz sheets, handling of import rows with different datatypes.
- Abstraction, for instance moving all code not essential in the main sheet “Game”, and seperate the two different game modes.
- Encapsulation in the quiz sheets, where the import statements is only mentioned one time even though they are repeated until the last row.