Testat 5

Programming in Java

WS 22/23

Learning objectives:

- **You manage persistency**
- **You separate the database from business logic and from GUI**
- **You better handle exceptions**

Implementing persistency for the program from Testat 4

AGAIN:
First, create UML class diagram, save it as PDF-file `Testat-5_ClassDiagram.pdf` in project subfolder called *docu* (as you did in the attestations before)

Task 1:

Create a persistent data repository for the program from Testat 4.
You can use a free relational DB system to store the objects persistently (permanently) in a database.
If you are not yet familiar with database handling, I prefer this:
You may use **Serialization**: **It is your choice!**

1. The program from Testat 4 is to be extended in such a way that it …
    a. checks at the beginning whether the defined database exists. If the database does not exist, an error message is issued, and the program ends. Otherwise all objects are retrieved from the database, created and stored in ArrayLists. NOTE: Before starting the program, you must create the DB manually and enter a few objects, so that testing your code is possible!
    If you use serialization then write a short program that adds some objects into a Serialization file.
    b. If an object is changed, deleted or created, this change must be made not only in the apps ArrayList, but also in the database/ serialization file.
    c. When user terminates the program, the database/ serialization file must be closed properly before the program really ends.

Task 2:

Extend the program in such a way that a submenu "Settings" is also available in the menu item "File". There, the user should be able to make basic program settings:

- name of the program. (which is displayed in the title bar of the main window)
- version of the program. (Displayed, if you click on "About" in the menu bar under HELP)
- name of the licensee or company using the program. (also shown under HELP/About)

Your task:
These three data are again to be made persistent by **serialization**. They can be displayed in the application window in the title bar or under HELP/About. This data is loaded at program start. There is an appropriate dialog for user's edit, which starts when you click on "Settings". This dialog has two buttons: **`Save`** or **`Cancel`**. If user clicks **`Save`**, the previously changed three values are serialized into a file, the application window is updated, and the dialog disappears. If user clicks **`Cancel`** the dialog disappears without any savings.