Project and Training 2
Mathematics, Programming, and Databases

BSc Computer Science, Spring Semester 2020

Version: April 23, 2020

1. Goal

The Goal of this exercise is to implement a program that calculates and visualizes a pie chart diagram based on values persisted in a relational database. An existing library for graphical visualizations may not be used for this. The intention is that the underlying mathematical concepts are understood and can be implemented in a corresponding Java/Kotlin program.

2. Project template and Database

There is a project template given as zip file on moodle. GIT repos will be provided too. The project template is a working JavaFX program (with an empty window).

The database to be used is the same as in block 1. The structure of the 3 tables to be used is given later in this document.

- DBMS: MS SQL Server
- Driver: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- URL: `jdbc:sqlserver://corpus.bfh.ch:55783;Database=pt2_2020`
- User: `GroupXY` (where XY is your group number)
- Password: Received by e-mail during block 1
- VPN: Please note that the database can only be used with a **VPN connection**.

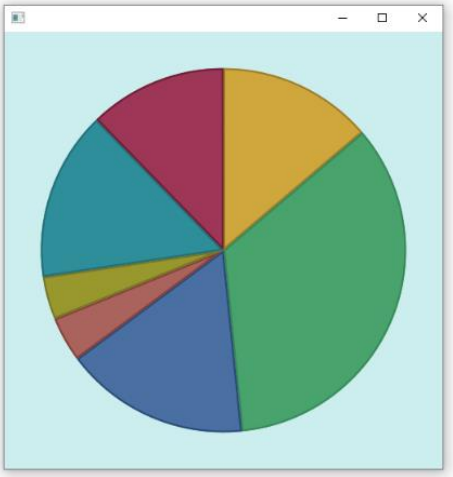
3. Tasks and Points

- Implement the class **`Matrix`**. Instances of this class represent matrices with decimal values and methods providing operations with matrices.
  - Methods / operations to be implemented:
    - Transpose
    - Determinant
    - Inverse
    - Multiplication with scalar
    - Multiplication with other matrix
    - Addition of another matrix
  - Furthermore, implement:
    - The 2 constructors
    - Getter methods for number of lines, number of columns, and values
    - The methods **`equals`** and **`hashCode`**
    - The **`toString`** method is optional but could be useful for debugging
  - Matrix objects have to be immutable

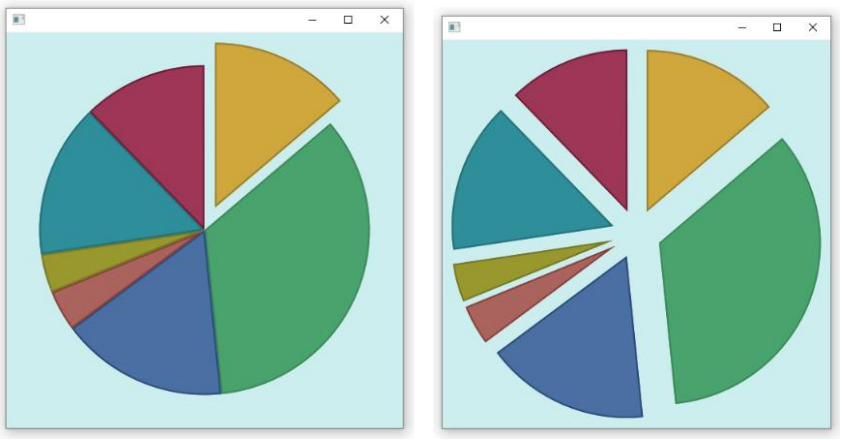
- When a method cannot run correctly due to non-satisfied conditions, it has to throw a **`MatrixException`** or one of its descendants
- The Matrix class has to be tested with the **`MatrixTest`** class. The achieved score corresponds to the number of successfully passed test methods **8 points**

- Implement Java/Kotlin code to do the following:
  (can be integrated into the application in the next step)
  - Read a set of integer values from the relation **`SalesValue`** **2 points**
  - Calculate the pie chart by calculating angles and positions of the slices depending on the values read before. Use your Matrix class. Graphical transformations (rotation, scaling, translation, …) can easily be calculated with matrix operations **2 points**
  - The positions have to be in a coordinate system with 2'400 x 2'400 points. The pie chart has its center at position (1'200, 1'200) and has a radius of 1'000 points. Angles are calculated in radian, start at 0.0 and rotate clockwise.
  - Write the calculated values back into the relations **`PieChartSlice`** and **`PieChartSlicePos`** **2 points**
  - This code can be integrated into the next step.

- Implement a JavaFX/TornadoFX application to visualize the pie chart
  - Read the data from relations **`PieChartSlice`** and **`PieChartSlicePos`** and create corresponding **`CircleSegment`** objects **2 points**
    (do not use classes from the javafx.scene.chart package)



  - Make that the size of the chart automatically fits the size of the window when the window size changes **2 points**
  - With a click on a segment, detach it from the chart and move it away from the center by an absolute distance of 150 (in the 2'400 x 2'400 coordinate system). Another click moves it back to its original position. Multiple segments may be detached at the same time. **2 points**



4. Deliverables

- Full source code
- Screen shots of your pie chart, one without and another one with detached segments if possible

5. Evaluation

The points mentioned in chapter 3 sum up to 20. The exercise is passed if you got at least **16 points**.

6. Database Tables

```
CREATE TABLE SalesValue (
    ProductId     INT NOT NULL
,   CONSTRAINT PK_Sales PRIMARY KEY (ProductId)
,   Value         INT
);

CREATE TABLE PieChartSlice (
    ProductId     INT NOT NULL
,   CONSTRAINT PK_PieSlice PRIMARY KEY (ProductId)
,   CONSTRAINT FK_PieSlice_SalesValue
      FOREIGN KEY (ProductId) REFERENCES SalesValue
,   StartAngle    DOUBLE PRECISION NOT NULL
,   EndAngle      DOUBLE PRECISION NOT NULL
);

CREATE TABLE PieChartSlicePos (
    ProductId     INT NOT NULL
,   CONSTRAINT FK_PieSlicePos_PieSlice
      FOREIGN KEY (ProductId)
      REFERENCES PieChartSlice
,   Pos           INT NOT NULL
,   CONSTRAINT PK_PieSlicePos PRIMARY KEY (ProductId, Pos)
,   X             INT NOT NULL
,   Y             INT NOT NULL
);
```