Software Entwicklung 2

Sommersemester 2018

MI7

StreberQuiz

Foo Bar, fb001@hdm-stuttgart.de

.....

https://gitlab.mi.hdm-stuttgart.de/?????/StreberQuiz.git

1. Kurzbeschreibung

Bei unserem Projekt handelt es sich um ein Lernkarteiprogramm für das zweite Semester des Studiengangs Medieninformatik an der Hochschule der Medien.

Man legt Fragen in den einzelnen Fächern/Vorlesungen an. Beim Anlegen einer Frage gibt man neben der Frage, eine richtige Antwort und zwei falsche Antworten ein. Die Frage bekommt dann einen Status (anfangs 0) zugewiesen, welcher durch das richtige Beantworten der Frage erhöht werden kann, sich aber auch durch das falsche Beantworten einer Frage wieder senkt. Wenn der Status 3(maximal) ist, gilt die Frage als „gekonnt“. Nach dem Status, sowie nach dem Fach kann man im Spielmodus die Fragen sortieren. So lässt sich neben allen Fragen eines Faches auch nur die gekonnten oder die noch nicht gekonnten Fragen auswählen. In jeder Spielrunde werden fünf zufällig ausgegebene Fragen gespielt. Am Ende jeder Runde wird angezeigt wie viele Fragen in der Runde richtig beantwortet wurden und insgesamt in dem Fach schon richtig (d.h. dreimal hintereinander richtig) beantworten sind.

2. Startklasse

Main-Methode befindet sich in der Klasse **`GameManager`**

3. Besonderheiten

Im Moment gibt es fast nur Fragen im Fach Theoretische Informatik, die wir bereits angelegt haben, um das Programm zu testen. (unter „spielen“ deshalb des Fach „Theoretische Informatik“ auswählen)

Außerdem haben wir um besser Testen zu können ist die richtige Antwort immer an mittlerer Position.

Wir nutzen JDK 9

Eventuell muss in der pom.xml

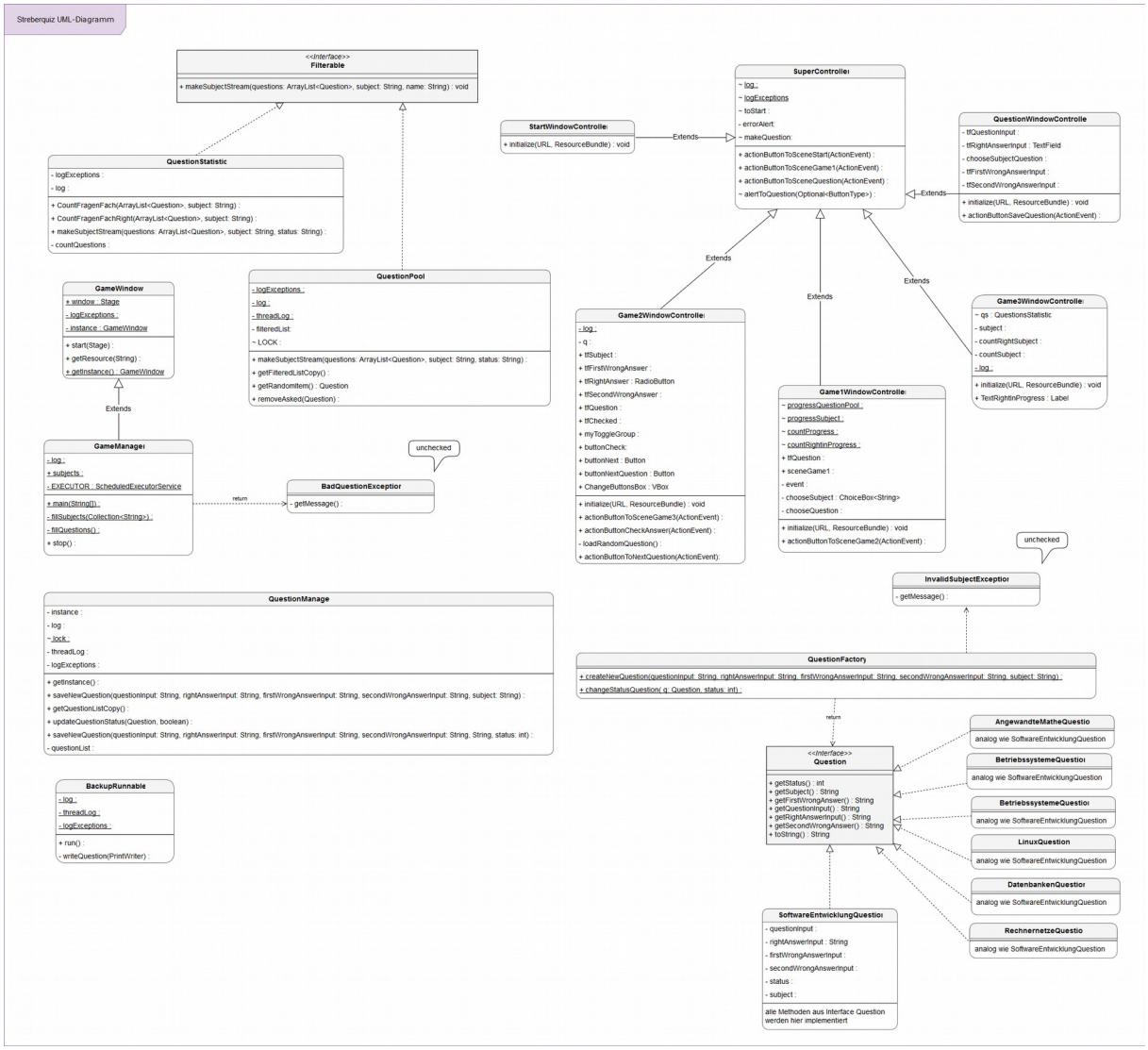
```
<plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-compiler-plugin</artifactId>
   <version>3.7.0</version>
   <configuration>
      <source>1.9.0</source>
      <target>1.9.0</target>
   </configuration>
</plugin>
```

Angepasst werden, hier hat bei manchen von uns je nach JDK-version (9.x.x) 1.9 oder 1.9.0 funktioniert. Ich vermute das muss mit dem, was in der misc.xml (nicht im git) steht, übereinstimmen.

```
<component name="ProjectRootManager" version="2" languageLevel="JDK_1_9" default="false"
project-jdk-name="9.0" project-jdk-type="JavaSDK">
  <output url="file://$PROJECT_DIR$/out" />
</component>
```

4. UML-Klassendiagramm

Das UML-Klassendiagramm finden Sie zusätzlich im git-Repository als: `Streber_Quiz_uml_<Datum>.xml` *(bitte mit draw.io öffnen)*



UML Use Case Diagram:

<diag>

Sequence Diagramm: Nur EINEN Flow !!!!

<diag>

5. Stellungnahme

Architektur

- Interfaces: 1. (… implementations > **`Filterable`**) wird implementiert von (… implementations > **`QuestionPool`**) und (… implementations > **`QuestionStatistic`**)

2. (… implementations > questions > **`Question`**) wird implementiert von (… > implementations > questions > **`'Fach'Question`**) es wir nur über die *Factory* **`QuestionFactory`** als <Question> auf die Questions zugegriffen

- Vererbung:

(…gameLogic > **`GameManager`**) erbt von (Gui > **`GameWindow`**) dadurch werden GUI und Gamelogik verknüpft

- im Ordner SoftDev2 unter Java haben wir vier Unterordner:

(1) *CustomException* beinhaltet unsere eigene Exception
(2) *gameLogic* (hier sind der **`GameManager`** und **`QuestionManager`**, sowie der **`BackupRunnable`** enthalten) ✉ Implementation der Hauptsteuerung der Geschäftslogik
(3) *Gui* enthält den Ordner Controller mit den einzelnen **`GameWindowControllern`**, die die Gui steuern, sowie dem **`GameWindow (extends Application + start-Methode)`**
(5) implementations enthält die Implementierungen unserer Gamelogik, sowie Interfaces und Factory

- die *fxml*-Dateien für die Gui befinden sich im Ordner resources

Clean Code: Es sind nirgends public member verwendet in den Kernklassen, statische Methoden waren nötig in … Referenzen werden meist kopiert vor einem getter() Aufruf

Exceptions

- (Unchecked) InvalidSubjectException: (implementation > **`questions > QuestionFactory`**)
  ✉ wird an Aufrufer zur Fehlerbehandlung weitergegeben: (gameLogic > **`QuestionManager`**)

Graphische Oberfläche

- haben wir mit *fxml*-Dateien erstellt, die sich im Ordner **resources** befinden. Im Ordner **GUI** > **Controller** befinden sich die Controller für die verschiedenen Scenes.

Logging

- Oben angegebene Exceptions werden mit Level fatal geloggt (z.B. (gameLogic > **`QuestionManager`**))
- allgemeines debugging (z.B. (Gui > Controller > **`Game3WindowController`**))
- debugging von Threads (z.B. (gameLogic > **`BackupRunnable`**))
- Es wird in unterschiedliche files und zusätzlich in ein gesamt-File zur Übersicht geloggt

- Konfiguration in resources > **`log4j2.xml`**

Threads

`BackupRunnable` führt ein „Backup“ im Hintergrund durch, indem es die gesamte Liste aus **`QuestionManager`**, die die Fragen enthält in ein *txt*-File schreibt. Um die Liste *threadsafe* zu halten wird eine Kopie der Liste übergeben und alle Funktionen in **`GameManager`**, die auf die Liste zugreifen, werden mit einem *Reentrantlock* gelockt.

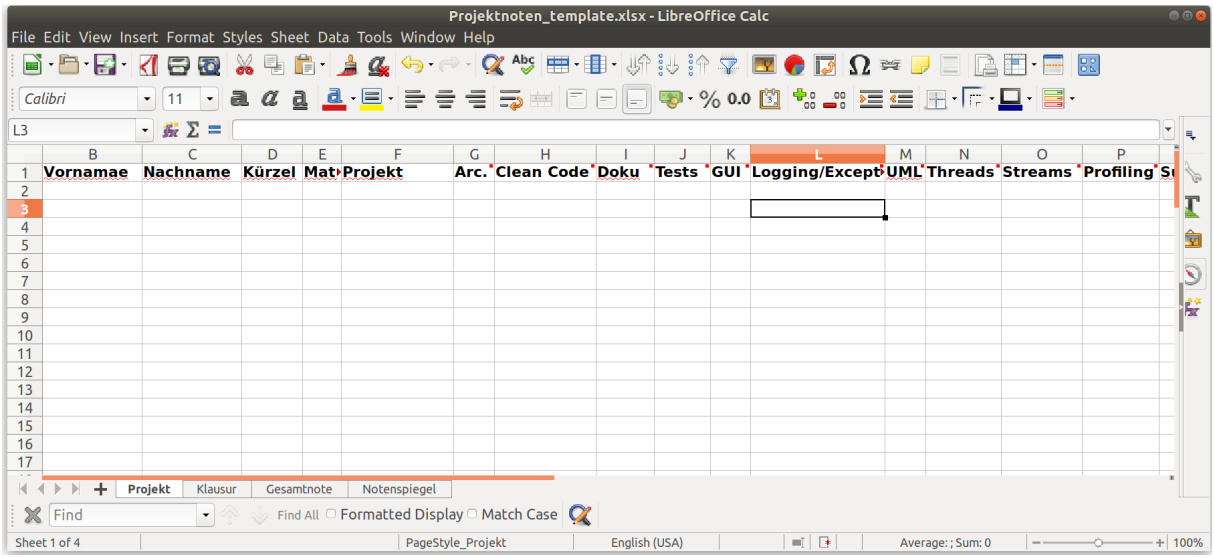
`Die Threads (BackupThread und Application Thread) werden im GameManager (main) gestartet.`

Streams und lambda-Funktionen

- Parallelstream in (implementations > **`QuestionPool`**), welcher die Fragen nach Fach und Status filtert und als neue *List(<Question>)* ausgibt. Der Konstruktor wird durch die Verwendung in (Gui > Controller > **`Game1WindowController`**) aufgerufen, nachdem der Benutzer ausgewählt hat, welche Fragen er spielen will.
  Die Liste wird am Ende Threadsafe mit einem *Collector* eingesammelt.
- Stream in **`QuestionStatistic`**, der den Status nach jeder Runde neu generiert

Dokumentation und Testfälle

- wichtige Methoden wurden kommentiert
- für die Tests haben wir mit *JUnit* Tests im Ordner **test** die zu den Klassen dazugehörige Testklassen erstellt, mit denen die einzelnen wichtigsten Methoden aus den Klassen auf ihre Funktion getestet werden können
  NegativTest in **`QuestionFactoryTest`**: testet, ob die *CustomException* **`InvalidSubjectException`** wie erwartet geworfen wird


Projektnoten_template.xlsx - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Calibri
11
L3
Vornamae
Nachname
Kürzel
Mat
Projekt
Arc.
Clean Code
Doku
Tests
GUI
Logging/Except
UML
Threads
Streams
Profiling
Projekt
Klausur
Gesamtnote
Notenspiegel
Find
Find All
Formatted Display
Match Case
Sheet 1 of 4
PageStyle_Projekt
English (USA)
Average: ; Sum: 0
100%

Bilder der Profiling Analysen zu Memory und Performance

(nicht solange Online Vorlesung, stattdessen Nachdenkzettel vom Team gemeinsam ausgefüllt im Repo)