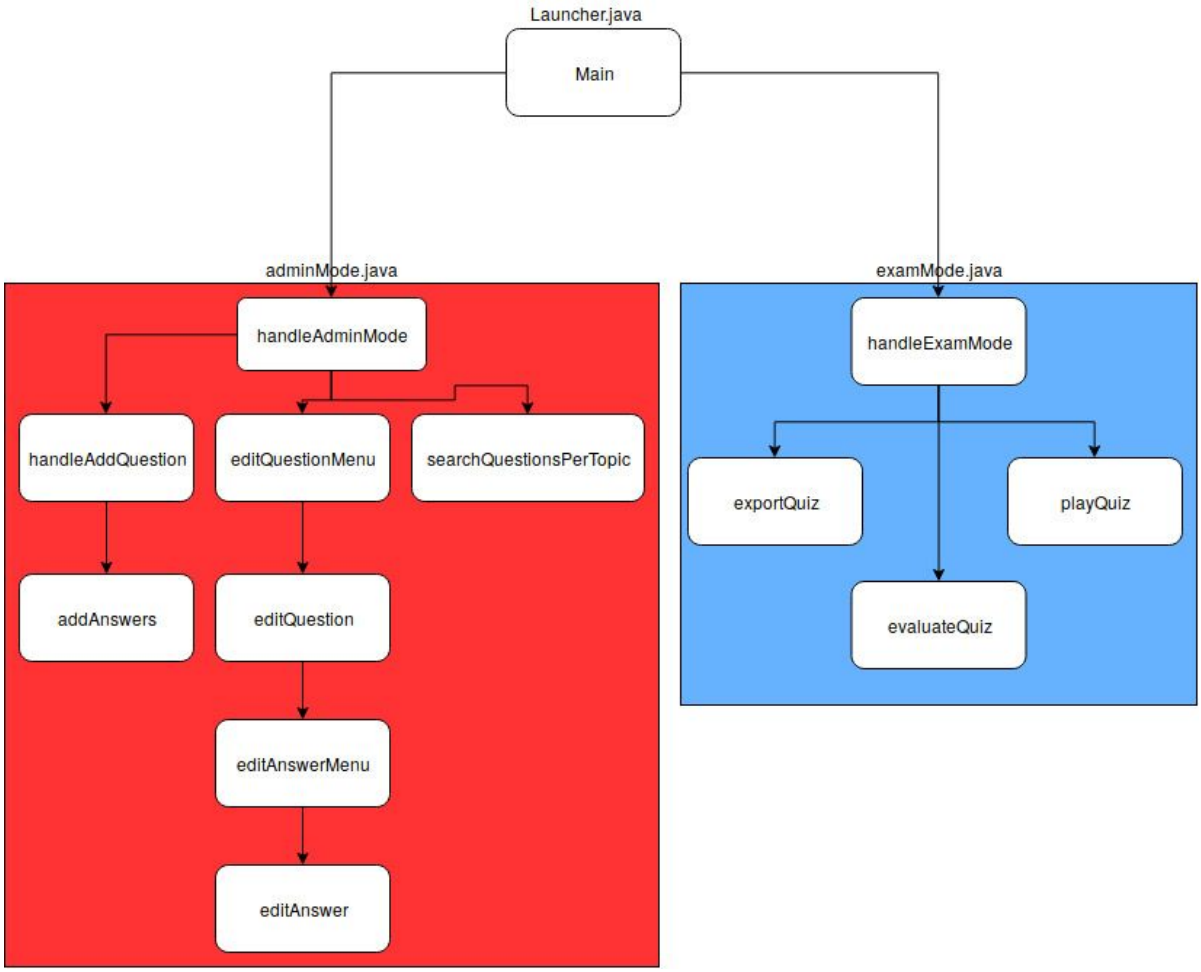
Technical Specification Document

Java Fundamental Project

APPLICATION FLOWCHART



I will first explain the structure of the application. As you can see in the picture above, there are three java classes used for the most important application functions. The first class is of course the launcher class, where the main class is located. This is where the program runs from. There are two other auxiliary functions located in the Launcher class, but the rest of the functions are outside this class.

adminMode class

Let's first focus on the adminMode class. This part of the application is designed for teachers, so they can add, edit, delete and view questions and answers. I have given only a brief overview of the functions below, because they are all relatively standard, and work in a standard way.

handleAdminMode: This is the menu function for this part of the application. The teacher can choose between administrator functions here.

handleAddQuestion: The teacher enters here the question, and its topic, difficulty, and whether or not it is a multiple choice question. It then runs the "addAnswers" function.

addAnswers: The teacher enters the answer to the question here. The functionality of this function depends on if it is a multiple choice question or not.

editQuestionMenu: This function shows a list of all questions in the database. The teacher enters the ID of the question to edit it. The question is not edited in this function, it calls the "editQuestion" function for that.

editQuestion: A specific question can be edited here. The question itself can be edited, or its difficulty, topic or answer(s). It can also be deleted.

editAnswerMenu: If answers needs to be edited, this function looks up the answers and calls the "editAnswer" function, that actually edits the answer. It is not possible to delete an answer, or create more answers for multiple choice questions.

searchQuestionsPerTopic: This function looks up all questions with a given topic in the database.

examMode class

I provided a more detailed overview of this class, because my implementation might not be very standard.

The teacher starts with entering the desired quiz topics and difficulty. All questions are fetched from the database that fulfill these requirements and are put in a list. The teacher then chooses if the quiz is to be printed or to be taken on the computer.

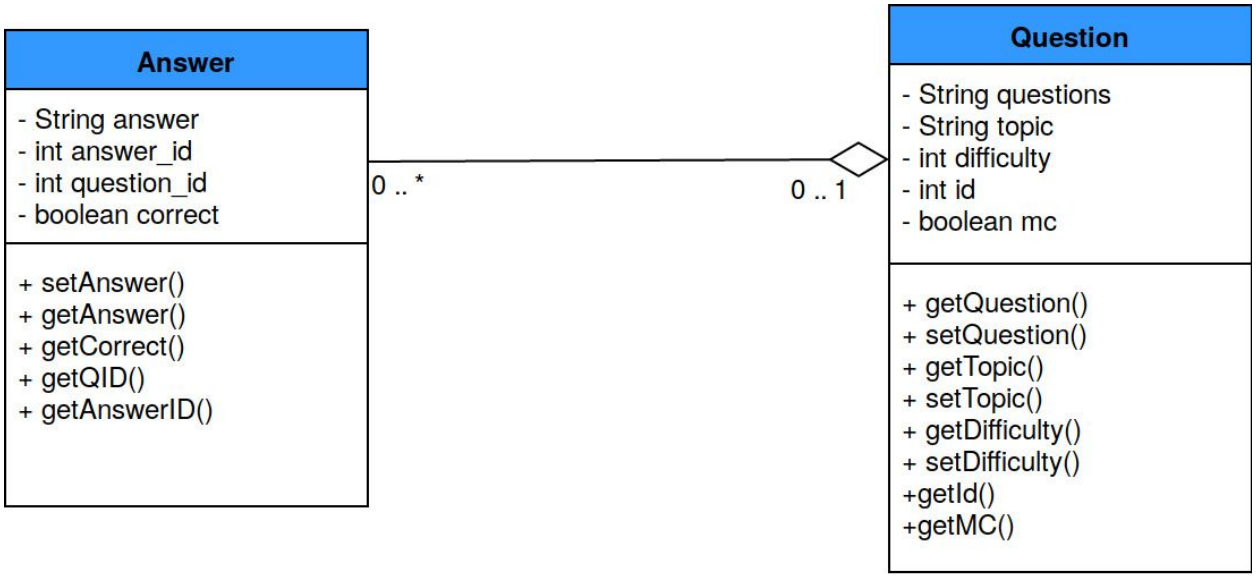
If the quiz is to be printed, the function calls "exportQuiz". This function writes the questions and multiple choice answers to a text file so the teacher can print this file. The name of this file is "Quiz-" followed by the selected topics and difficulty.

If the quiz is taken in the application, the function asks for student name and ID. This data is saved in a list and is used later. The function “playQuiz” is then called. It loops over this question list and asks the student for an answer every iteration. If a question is a multiple choice question, its multiple choice answers are fetched from the database. The student answers are saved in another list.

This function, “playQuiz”, returns the answer list and then the function “evaluateQuiz” is called to calculate the quiz result. A text file is generated, containing all quiz questions and student answers. All multiple choice questions are corrected and the partial result is shown at the bottom of the file. The student answers to non-multiple choice questions are not evaluated, and are just written to the file. Database answers to these questions are also written to the file so the teacher can correct the quiz more easily. The name of the file is the ID and name of the student. The teacher can print this file and correct it with ease.

The partial result of the multiple choice questions is also printed to the console immediately after the quiz for the student to see.

UML DIAGRAM



I thoroughly simplified the standard UML diagram that was given in this assignment of the project. Instead of creating classes for Quiz and Student, I simply processed the Quiz as a list of Question class objects, and I temporarily saved the required student data in a String list.

I combined MCQQuestion and Question by creating a boolean variable that indicated whether a question was a multiple choice question or not. To combine MCQAnswer and Answer, I added

the variable “question_id” to each Answer object, while also providing objects with their own unique ID. This way, I can easily find a group of answers for a multiple choice question by searching for a question_id, but I can still edit individual answers by looking for their unique answer_id. I also added the boolean “correct” for multiple choice answers.

APPLICATION LIMITATIONS

There are several features that I wanted to implement, but I did not accomplish this due to time constraints. For example, it is currently not possible to delete or add multiple choice answers. The answers can be changed, but the number of answers for a particular question cannot be changed. There is also no control implemented to check if the teacher accidentally labels more than one multiple choice answer as correct. In general, there is a lack of user input control in the application. Also, questions can only have one topic.

I considered these things to be of secondary importance, and I first wanted to focus on implementing all the required features.

SOURCES

https://stackoverflow.com/questions/10631715/how-to-split-a-comma-separated-string