Home exam (hjemmeeksamen)

PGR112 Object-oriented Programming

Spring 2022

Home exam, individual

Notice:

- The main topics in this exam are Java, JDBC, and diverse Object-oriented programming theory as given through the course.
- All code that is not from a course related git repository or from any of the lecture slides, must be coded by you, and you alone. If you do use existing
- code, you must clearly show (in comments) what the source was
- You should put all the files that the delivery contains in one folder and zip before uploading to WISEFlow.
- The solution should be able to run properly in IntelliJ (i.e. not have to run with any other special technology/IDE)
- If you are unsure about something regarding the exam, re-read the instructions and try to make a best assumption. Clearly express what assumptions you made and why in the report (see below).

The delivery will contain three things:

1. The project folder with Java, folders, txt or json files if any, etc.
2. The database schema files you have created (see SQL files in repository for examples).
3. A small report (pdf or word document) that shortly describes the major OOP concepts and how you have implemented them through your design.

Case QuizGame

You are given the task to implement an interactive Quiz game application. You will select two Quiz topics which you find interesting and include them in this application. Examples: sports, football, boxing, tennis, games, movies, music, history, etc. One Quiz topic shall contain only multichoice questions (questions with 4 answers), and the second Quiz topic shall contain only binary questions (questions with yes/no answers). Number of questions for each Quiz topic shall be at least 4.

You shall create all Quiz questions and store them in database. You shall design an interactive user interface, which reads user inputs, communicates with database, and gives response to users. It should also be easy to choose and change Quiz topics or quit the game.

You need to sign up as a user before start playing the Quiz game. The Quiz game has a scoring board. The scoring board will show score history to the user. The scores are stored in database.

In addition, you shall make a report (200-400 words) in a pdf or word document about the following things:

- What is Objected-oriented programming? What concepts did you use in your solution? How did you use these concepts to develop your solution?
- What is JDBC? How did you use JDBC?
- Examples of OOP concepts that you have implemented in your QuizGame design.

What are you to make?

- **Database and tables**
  - You will create a database called quizDb.
  - You will create two tables that stores Quiz questions. One table is called multichoiceQuiz, the second table is called binaryQuiz. You shall insert Quiz questions into tables using JDBC connection.
    - The multichoiceQuiz table contains following columns for each record: id; question; answerA; answerB; answerC; answerD; correctAnswer.
    - The binaryQuiz table contains following columns for each record: id; question; correctAnswer.
  - You will create a table that stores the score history. You shall insert a new score record at each round using JDBC connection.
    - The score table contains following columns for each record: id; user; score; topic
  - Other things you think should be included
- **Interactive Java program**
  - You shall use Java.Util.Scanner to build the interactive QuizGame Java program.
  - You need to sign up as a user before start playing the Quiz game.
  - You shall be able to play the QuizGame for unlimited times unless you want to quit the game.
  - You shall be able to choose Quiz topics before start. Once you start, you are not allowed to switch topic.
  - The Quiz game has a scoring system. If your answer is correct, add one point to the score.
  - You will get a final score each time you finish playing a round.
  - You will also be displayed with a scoring board each time you finish playing a round. The scoring board shows the score history where each record presents “user, score, quiz topic” and listed in an descending order sorted by score.
  - Random questions at each play.

As an example, your QuizGame terminal display could be about the following steps:

1. Sign up a new user:
   Enter your user name at terminal
2. Choose Quiz topic:
   A. Tennis (Multichoice Quiz)
   B. Football (Binary Quiz)
   Choose A for Tennis Quiz
3. Question 1: Who ranks No.1 in Men’s Tennis ATP Ranking 2022?
   A. Novak Djokovic
   B. Rafael Nadal

C. Casper Ruud
D. Roger Federer

Choose A as answer

4. Your answer is correct! Would you like to
   A: Continue with next question
   B: Exit current round

   Choose A, continue until all Quizs are done
5. Your answer is wrong!
   Your final score is 3 out of 4! The score board is as below:

   User1 3 Tennis

   User2 2 Football

   You have following options:
   A. Continue with another round
   B. More options
   C. Quit the game

   Choose B for more options
6. You have below options:
   A. List all scores for a user (please enter the user name):
   B. List all score records >= 3 for Tennis
   C. Continue with another round
   D. Quit the game

   Choose D to quit
7. Bye and Welcome back!

Note that the example is just for your reference. You have a lot of freedom regarding the user interactive program and how to show information. You want to add as much details and complexity as you can.

Other details

- You can create a Quiz class as an abstract class or interface. You also create MultichoiceQuiz and BinaryQuiz classes as child classes of Quiz class. In Quiz class, you will define at least but not limited to below functions:
  - showQuestion() – it will show current Quiz question.
  - isCorrectAnswer(String answer) – it will add one point to final score if the answer is correct.

  The function implementations are expected to be different in MultichoiceQuiz and BinaryQuiz classes. Think about inheritance and abstraction concepts we have learned in this course.

- You can create a UserScore class, and initiate a new object each time when we start a Quiz round. The UserScore class is to update score for the current user for a given selected Quiz topic.

- The Quiz questions and score records are stored in the database.
  - You shall insert all Quiz questions into database using JDBC connections.

- You can fetch all Quiz questions from database at once, or fetch one Quiz question each time.
- You shall insert a new score record into database as long as you finish a Quiz round.
- You will be able to a) list all scores, b) list all scores for one user, c) list all scores above a certain value, etc. Try to use Java techniques such as Lambda, Optional, iterators etc to solve these tasks.

Assessment criteria, for students and assessors

The following will be the main points to assessed:

- Java, Object-oriented programming, and JDBC is what is assessed in this exam.
- The basic OOP concepts such as Inheritance, Polymorphism, Abstraction and Encapsulation.
- Data types such as ArrayList, HashMap, etc.
- Use of advanced Java techniques such as Iterators, Optional and Lambda expressions.
- Exception handler to handle exceptions, and input validation to validate user input.
- Amount of (good) code and complexity are of importance
- Folder structure, tidy code, naming convention

Guideline for how much each part counts

The percentage numbers below present approximate numbers on how much each part will count

- User Interface logic: ca. 20%
- Use of OOP concepts: ca. 20%
- Use of JDBC: ca 20%
- Java techniques (data type, advanced techniques, exception handler etc): ca. 15%
- Report: ca. 10%
- Code structure, tidy code, good names on variables and functions, etc.: ca. 10%
- *Amount of code and complexity: ca. 5%

*All points affect each other. The assessment will require a holistic view of techniques applied in the exam delivery.

--End of exam text--