



A JAVAProject Report
on
Digital Quiz Management System
[QUIZ MANAGER Project]

Submitted by
Namrata Supe

Contents

<u>1.Subject Description</u>	3
[1.1]Introduction	3
[1.1.1]Overview:	3
[1.1.3]Motivation:	3
[1.1.4] Objective	3
[1.1.5] Development Environment	4
<u>2.Subject analysis</u>	5
[2.1] Major Features	5
[2.2] Application Feasibility	5
[2.3] Data Description	5
[2.4] Expected Results	6
[2.5] Scope and limitations	6
<u>3.Conception</u>	8
[3.1] Chosen Algorithm	8
[3.2] Data Structures	8
[3.3] Global Application Flow	9
[3.4] Class Diagram for Application	10
<u>4.Console operation Description</u>	11
<u>5.Configuration</u>	12
<u>6.Screenshots</u>	13
<u>Bibliography</u>	17

1.SUBJECT DESCRIPTION

[1.1]Introduction

[1.1.1]Overview:

This report contains all the detail from start to end of Quiz Manager project development.

[1.1.2] Background:

The usual problem with digital quiz application while preparing and running evaluation are to constitute an appropriate evaluation corresponding of the required level, be able to reuse former questions, organize sample evaluations and be able to correct MCQ questions automatically.

This Project is an attempt to address the above issues with an intention to learn basic concepts of JAVA. User should be able to take the quiz based on topic and level selected. Once user has provided all the answers, application will display global result.

[1.1.3] Motivation:

This report and its subject, has been greatly influenced by our will to learn the concepts of a new programming tool **JAVA**.

What we encountered is that

- Java is Easy to write
- Readable and Reusable
- Most of the concepts are drew from C++ thus making Java learning simpler.

[1.1.4] Objective

The main goal of this project is to develop an console application using Java to manage digital quiz preparation and execution.

The following are the objectives that the application needs to satisfy:

1. Operates **CRUD** on Open Questions and MCQ Questions
2. **Searches** questions based on topics
3. **Assembles** automatically a quiz based on topic
4. **Exports** quiz under a plain text format
5. **Runs** the evaluation
6. **Provides** automatic mark in the end of this execution

[1.1.5] Development Environment

PLATFORM USED : **MacOS Mojave version 10.14.3**
LANGUAGE USED : **Core Java, SQL**
IDE : **Eclipse**
DATABASE : **H2**
EXTERNAL JARS : **h2-1.4.197.jar and itextpdf-5.5.13.jar**
JRE LIBRARY : **JAVA SE 11.0.1**
JAVA COMPILER : **JAVA 10**

2.SUBJECT ANALYSIS

[2.1] Major Features

- Platform Independent
- Easy to use
- Robust
- Clean separation of various components

[2.2] Application Feasibility

- This application is a prototype for a Digital Quiz management system.
- The costs are much reduced as we do not depend on graphical interface, instead look for a high system performance
- Most of the components used such as the development platform and databases are open source.

[2.3] Data Description

The data description and data access objects are clearly specified below.

1. The Schema for Questions is <ID, TEXT, TYPE, LEVEL>

ID	:	INT, Auto generated, Unique
TEXT	:	STRING
TYPE	:	STRING (M – MCQ or O – Open Questions)
LEVEL	:	INT (1 – Easy, 2 – Average, 3 – Difficult)

2. The Schema for Answers is <ID, TEXT, IS_CORRECT>

ID	:	INT, Auto generated, Unique
TEXT	:	STRING
IS_CORRECT	:	Boolean

3. The Schema for Topics is <ID, TEXT >

ID	:	INT, Auto generated, Unique
TEXT	:	STRING

4. The Schema for **Questions_Topic** is <QUES_ID, TOPIC_ID>
QUES_ID : INT, Foreign Key to ID in Questions
TOPIC_ID : INT, Foreign Key to ID in Topics
5. The Schema for **Question_Answers** is <QUES_ID, ANS_ID>
QUES_ID : INT, Foreign Key to ID in Questions
ANS_ID : INT, Foreign Key to ID in Answers

DAOs

1. **QuestionsService** : This DAO class performs CRUD on Questions
2. **AnswersService** : This DAO class performs CRUD on Answers
3. **TopicsService** : This DAO class performs CRUD on Topics
4. **QuestionAnswersMappingService** : This DAO class performs Insert and delete on Question_Answers
5. **QuestionTopicsMappingService** This DAO class performs Insert and delete on Questions_Topic

[2.4] Expected Results

- To be able to assemble the quiz based on selected topic and level
- To be able to run the quiz and get the answers from the user. All the answers should be recorded and at the end of the quiz the global result should be displayed to the user
- To be able to save the quiz in PDF format

[2.5] Scope and limitations

Scope:

- Centralization of user administration
- Used in institutions to evaluate students
- Reduce evaluation time for exams and paper
- Reusable

Limitations:

- Lack of user GUI/web interface
- Lack of history for a student
- Lack of authentication system

Evolution:

We are working to enhance the Quiz manager as a complete web-based software, users and login and keep details, export details, import to new system.

3.CONCEPTION

[3.1] Chosen Algorithm

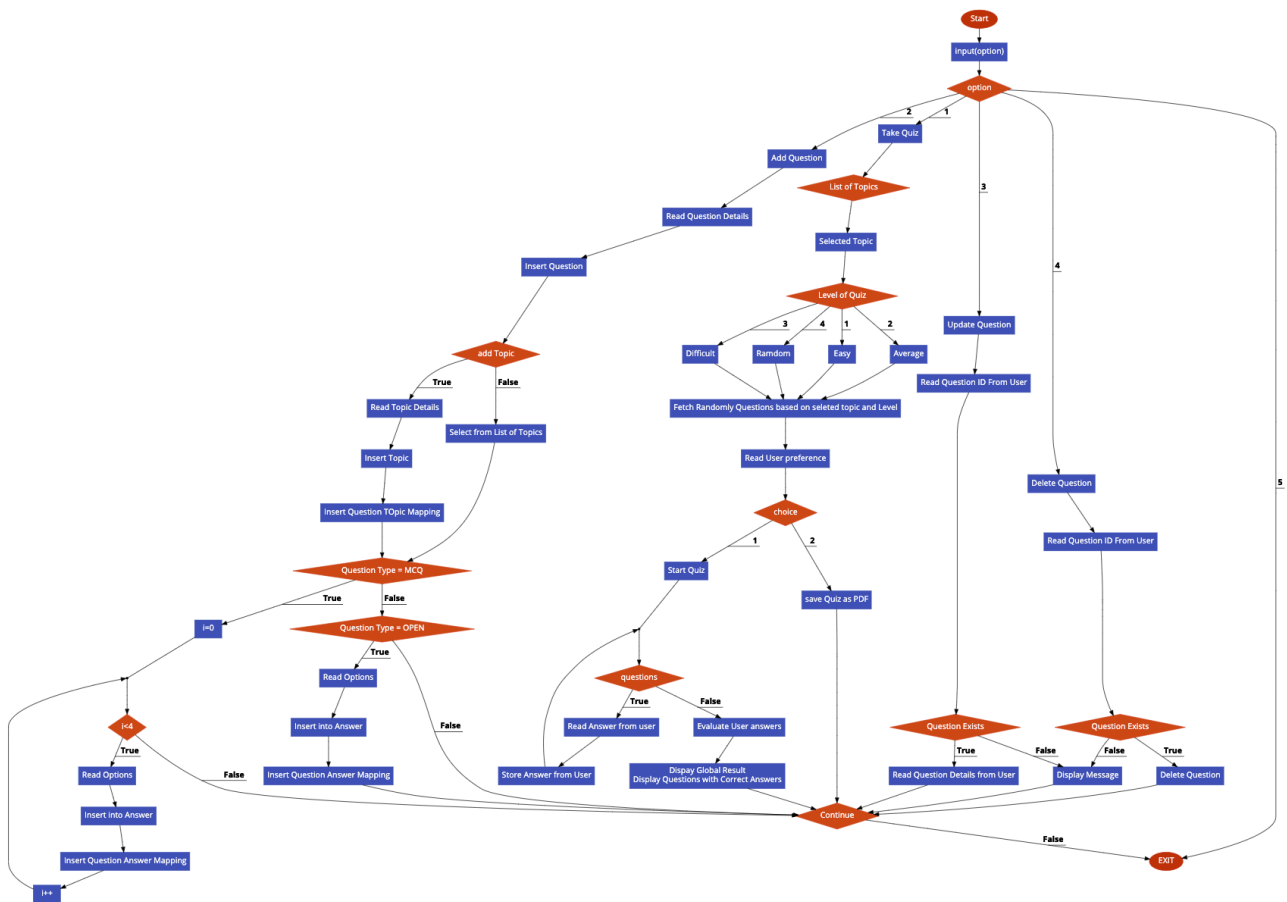
The algorithm that we have used is the exact match algorithm where searching is based on **ID**

[3.2] Data Structures

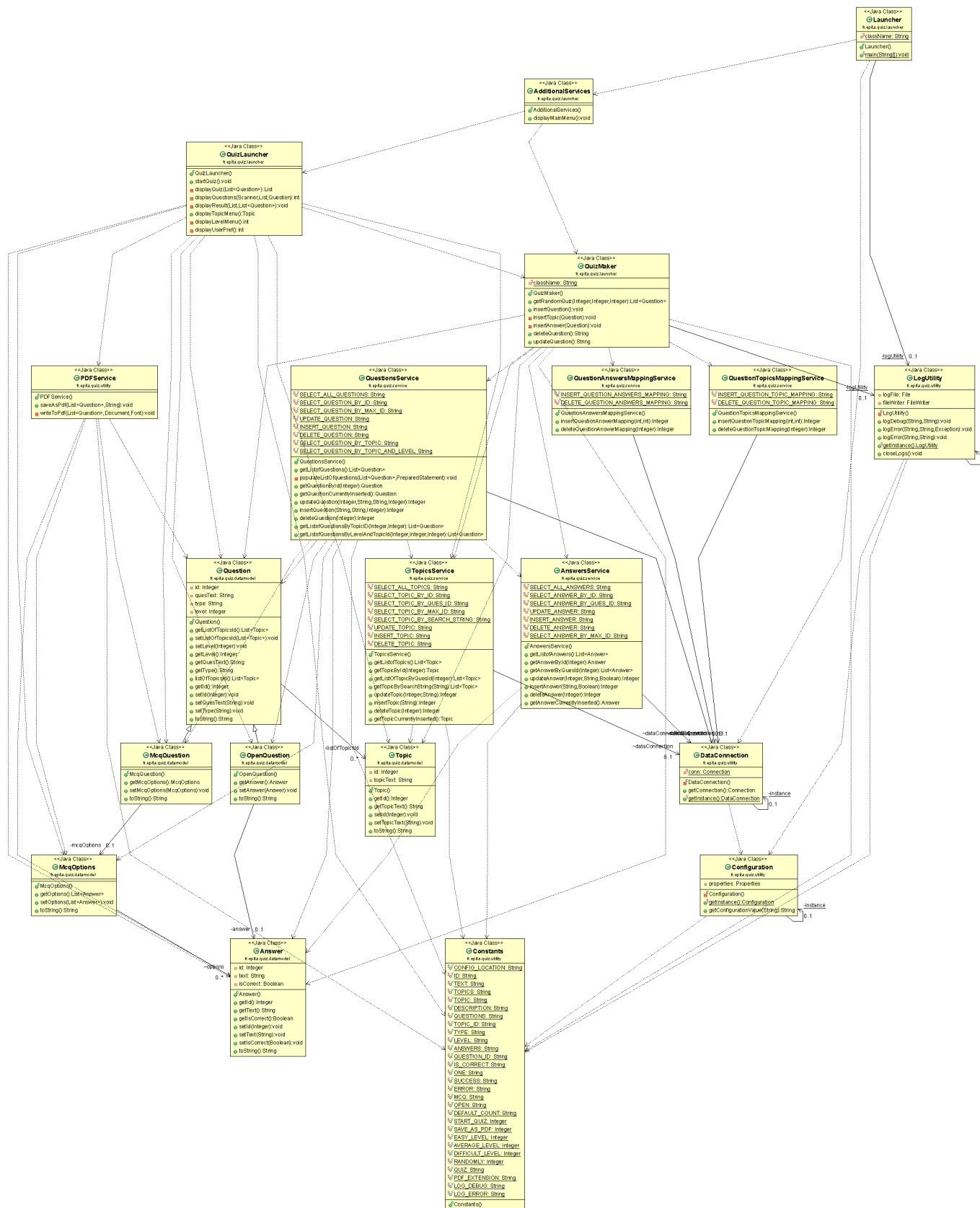
The data structures provided by the Java utility package are very powerful and perform a wide range of functions. However the data structure used in this project is **LIST** ,which is a collection of elements.

String and **Boolean** are also a widely used data structure in this project.

[3.3] Global Application Flow



[3.4] Class Diagram for Application



4.CONSOLE OPERATIONS DESCRIPTIONS

Console Operations Implemented in this Systems are

1. Assemble quiz based on Selected Topic and level
2. Run Quiz and get answers from user
3. Evaluate users and provide global results with correct answers
4. Save the quiz as PDF file
5. Insert new question
6. Update question on ID
7. Delete a question based on ID

Each Operations are explained below:

- **Assemble quiz based on Selected Topic and level**

A set of question is aseembled by an **getRandomQuiz** method , that takes input as topic id , count and level, and calls a **getListofQuestionsByLevelAndTopicId** method which connects to the database and fetches the list of questions.

- **Insert Question**

This Allows to add a new question with Answers and Topics to the database.

- **Update Question**

This Allows to update question based on Question ID

- **Delete Question**

This Allows to Delete question based on Question ID

5.CONFIGURATION

- Username : root (Case sensitive)
- Password : root(Case Sensitive)
- Database : **H2**
- Drivers : org.h2.Driver
- Run configuration arguments : -Dconfig.location="app.properties"
- Add h2 and itext jars

6.SCREENSHOTS

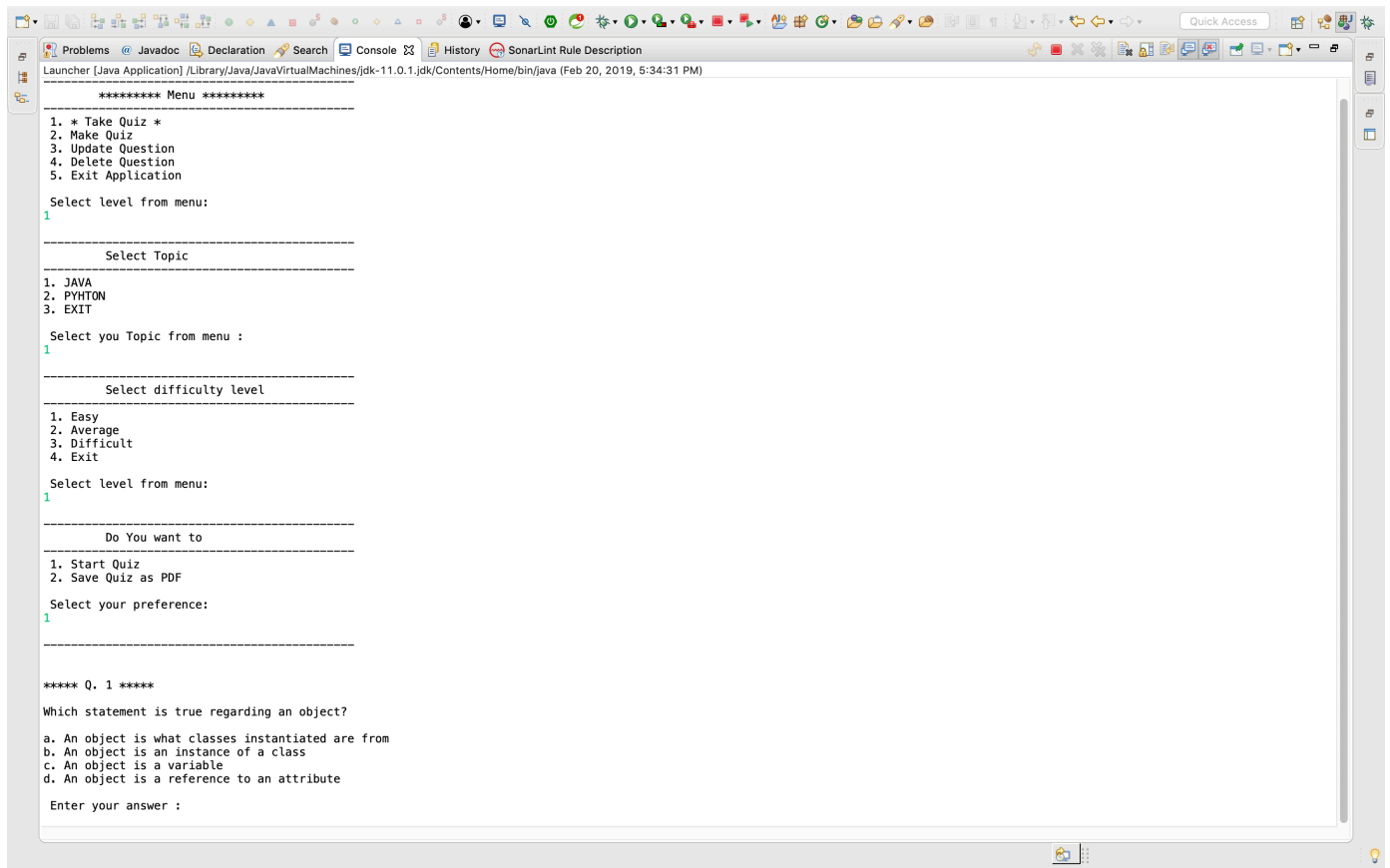


Fig 6.1: Display Menu and read user choice to start Quiz

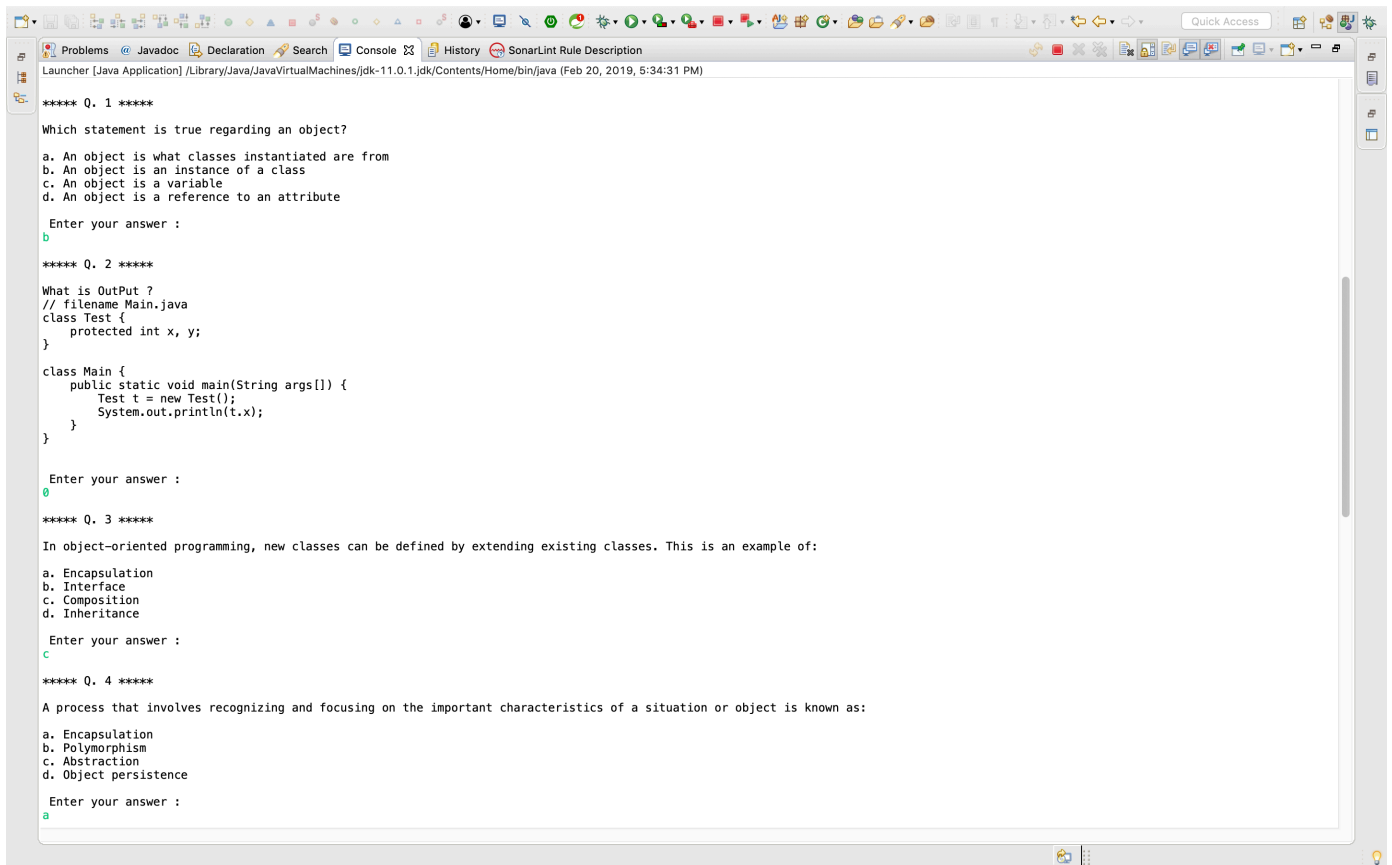


Fig 6.2: Display questions and read answers from user

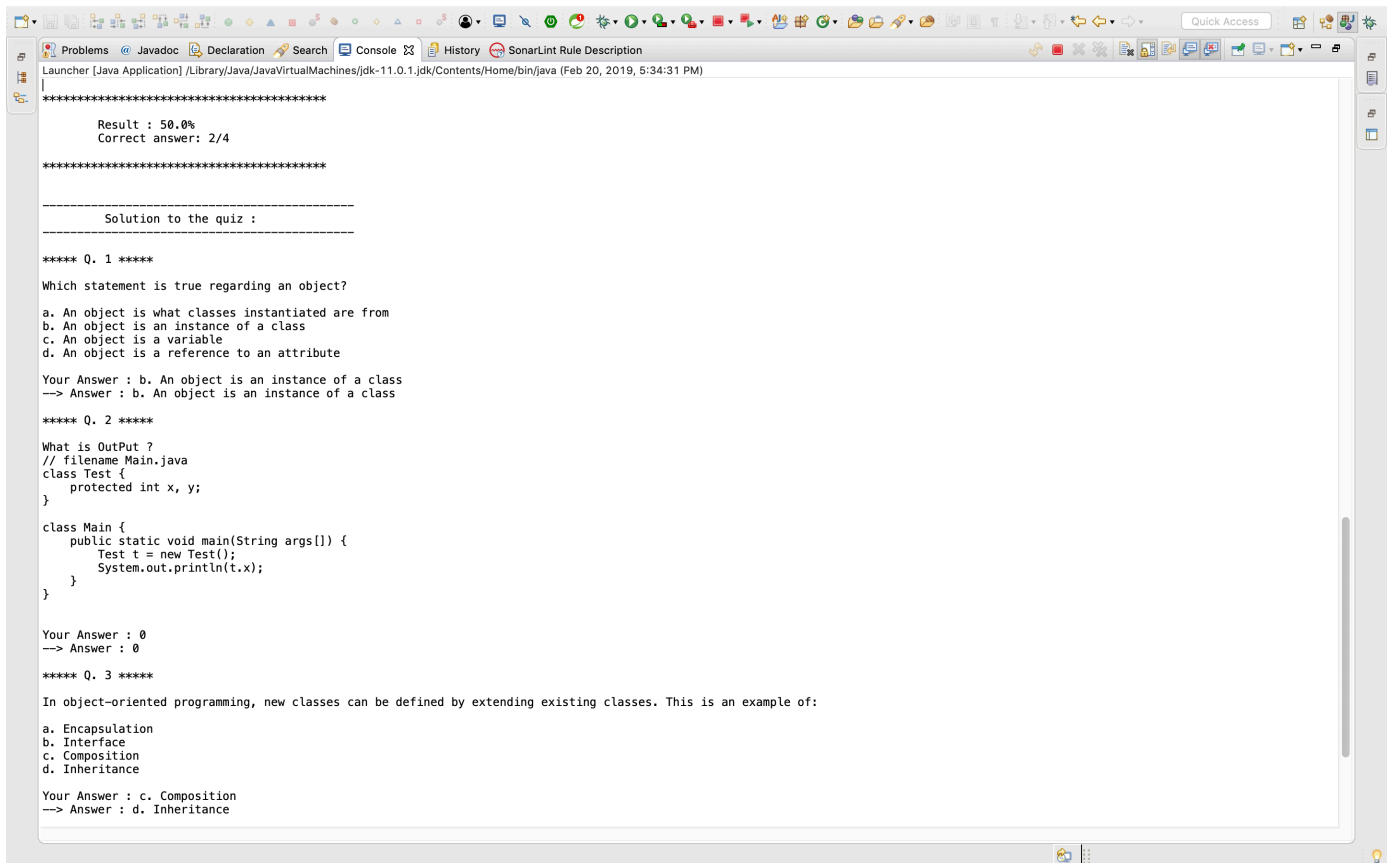


Fig 6.3: Display Result and Correct Answers per question

BIBLIOGRAPHY

- <http://thomas-broussard.fr/>
- <http://www.tutorialspoint.com/>