

Quiz Question Bank

Author: Nogay Viktoria EEAIR24

Description:

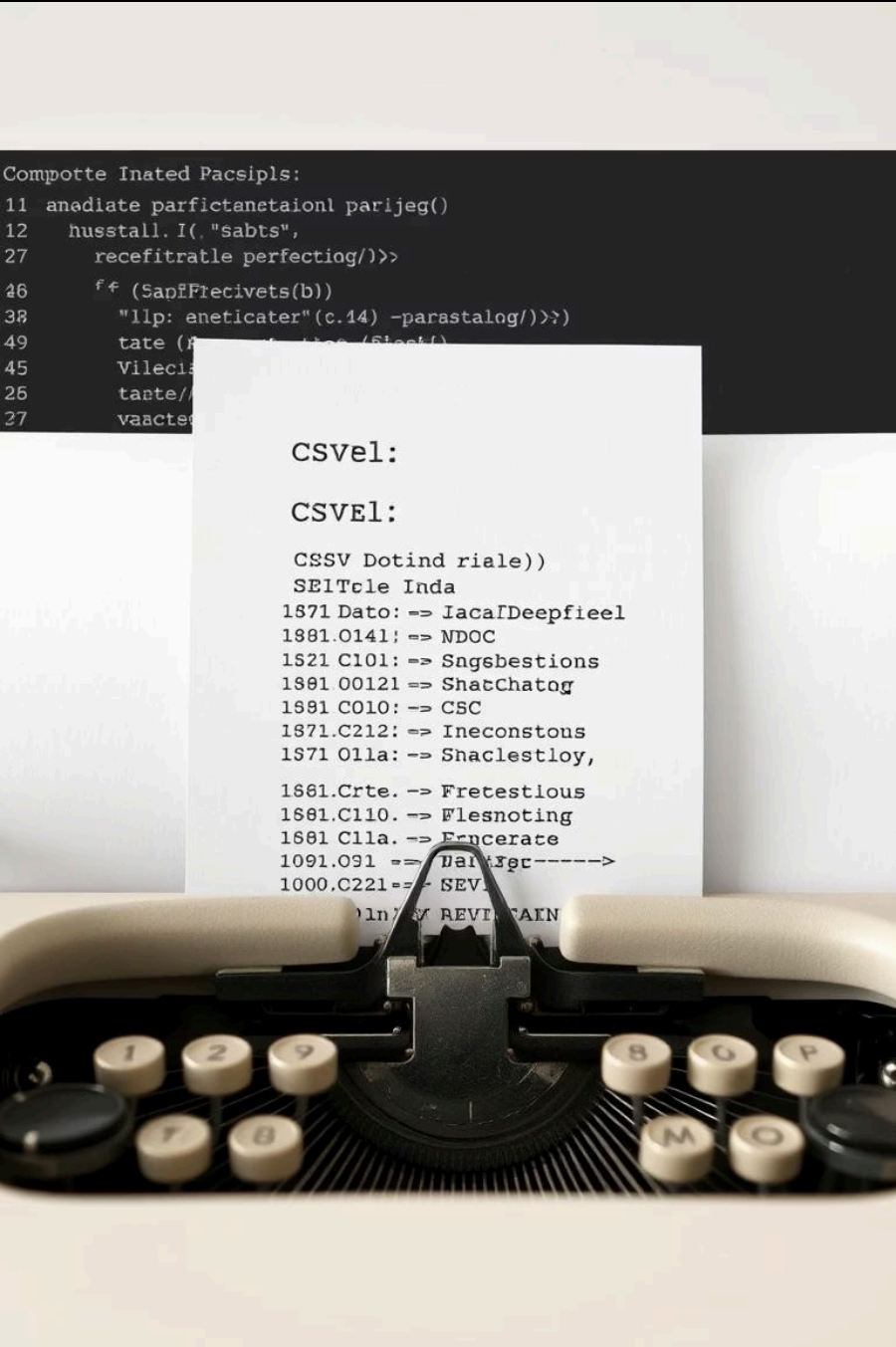
This project is a Quiz Application created in Java. It allows users to create, edit, delete, and take quizzes through a simple text-based interface. The application also supports saving and loading quizzes from a CSV file, take the quiz and receive a score and generates a report of operations performed.

Objectives:

The main goals of this project are:

1. Implement a fully functional CRUD system for quiz management.
2. Practice programming with Java.
3. Introduce file operations (CSV reading/writing).
4. Validate user input and handle edge cases.
5. Track user actions for a simple reporting system.





Project Requirement List

Operations for Quiz Questions

- Add one or multiple questions.
- Delete a question by specifying its index.
- Edit an existing question by specifying its index.
- Display all quiz questions without showing the answers.

CSV Integration

- Save the quiz data to a CSV file.
- Load quiz data from an existing CSV file.

Quiz Functionality

- Allow the user to take the quiz, answer questions, and calculate the score.
- Display correct answers for any wrong responses during the quiz.

Report

- Track the operations performed and provide a summary report.

Documentation



Data Structures

HashMap: Used to track the number of times each operation is performed. This allows easy reporting of the frequency of actions such as adding, deleting, and changing questions. ArrayList: Used to store quiz questions and options dynamically. It provides flexibility in managing the questions, adding new ones, and modifying existing ones.



Functions

Main Class: Contains the main logic of the application, including the menu system and interaction with the user. It calls various methods to manage the quiz. Quiz Class: Responsible for managing the quiz questions, including adding, deleting, editing, and displaying them. Question Class: Represents individual quiz questions, including the question text, options, and correct answer. CSV Operations: Methods for saving and reading quiz data to/from a CSV file. This includes serialization and deserialization of quiz data. Reporting: A method to track and print a summary of operations performed on the quiz.



Challenges

At first I couldn't save questions as a separate quiz. So I decided to create a separate class for this. The next problem was in the csv files. I tried for a long time to figure out how to save them and how to output them. In the end, I managed to solve all these issues.

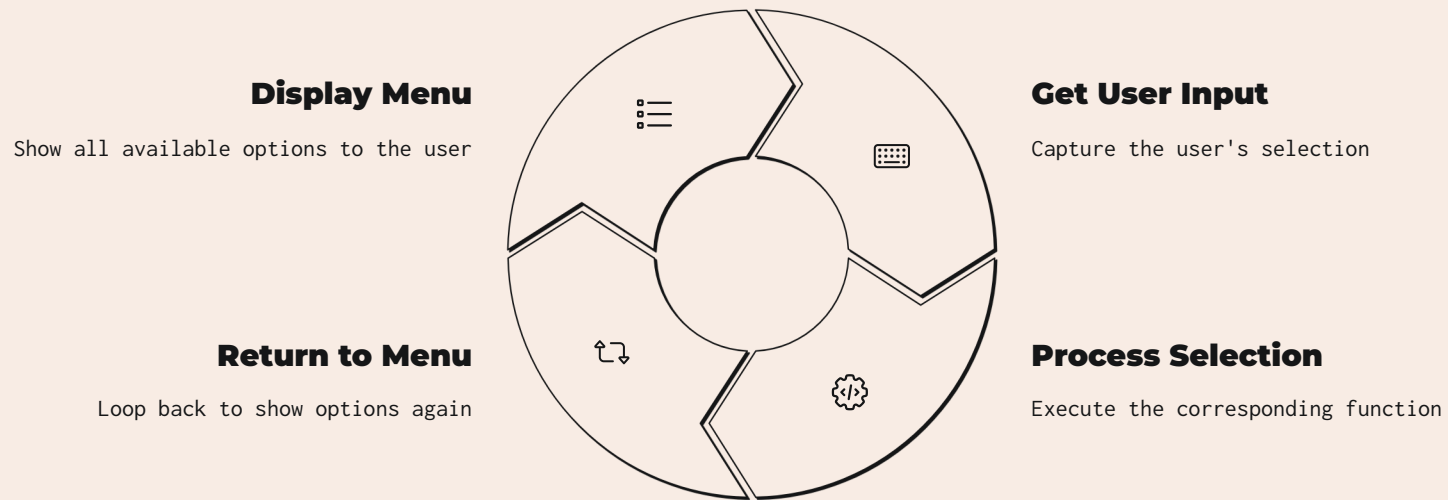
Purpose of the Main Class

The Main class is the **central controller** of the quiz application. It:

- Displays the **menu**.
- Reads **user input**.
- Performs **actions** (add, delete, edit, take quiz, etc.).
- Tracks how many times each operation is performed.
- Coordinates with other classes like Quiz and Question.



Menu

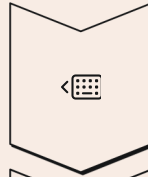


In the `main()` method, there's an infinite loop:

```
while (true) { // display menu and get user input  
    int number = input.nextInt(); // user's choice
```

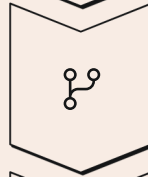
The user selects an action from the menu (like add question, take quiz, etc.).

Switch-Case Block



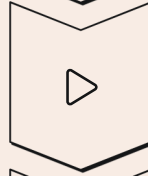
User Enters Choice

A number corresponding to a menu option



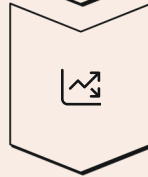
Switch Statement Evaluates

Determines which case to execute



Case Block Executes

Runs the appropriate function



Operation Tracked






Records the action for reporting

User choices are handled by a switch:

```
switch (number) { case 1 -> { quiz.addque(createQuestion(input)); Operation("Add one question"); } }
```

Depending on the number, the program runs the appropriate feature (like adding a question, saving to file, etc.).

Quiz Class

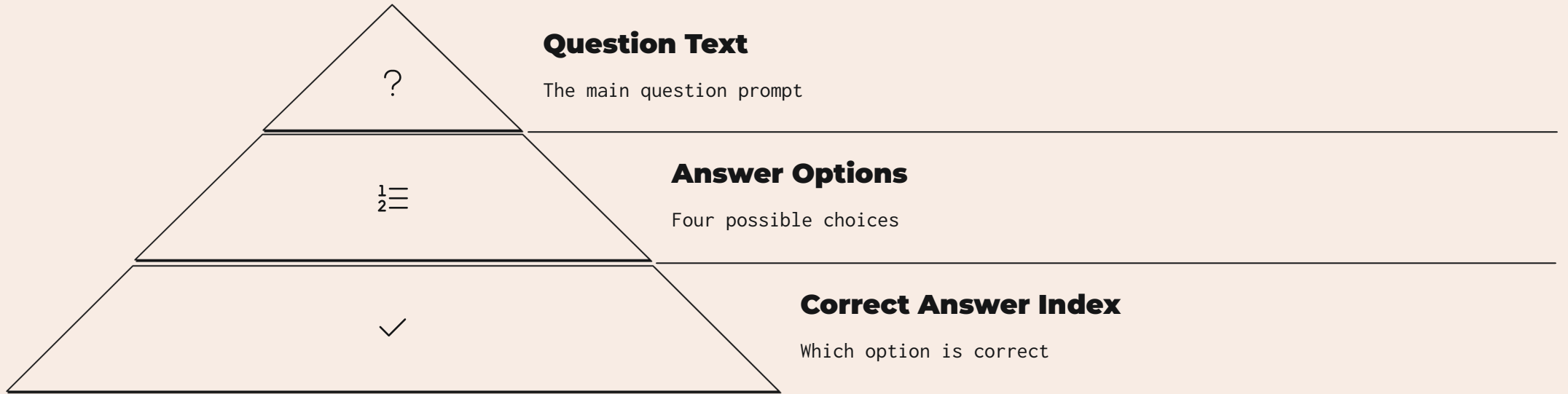
| | |
|---|--|
|  | Class Definition Defines the Quiz structure with ArrayList |
|  | Add Question Method Adds new questions to the quiz |
|  | Delete Question Method Removes questions by index |
|  | Edit Question Method Modifies existing questions |
|  | Display Method Shows questions without answers |

```
class Quiz { ArrayList quiz = new ArrayList<>();
```

This class holds a list of questions and provides methods to:

- addque – add a question.
- deletque – delete a question.
- changeque – edit a question.
- printQuizWithoutAns – print all questions without answers.

Question Class



```
class Question { String que; // question text ArrayList options; // answer choices int ans; // index of the correct answer
```

This class represents a single question with:

- Text of the question.
- 4 options.
- Index of the correct answer.

createQuestion() Method

Prompt for Question Text

Ask the user to enter the question

- Captures the main question prompt
- Validates input is not empty

Collect Answer Options

Get all four possible answers

- Prompts for each option one by one
- Stores options in an ArrayList

Identify Correct Answer

Ask which option is correct

- User specifies which option number is correct
- Validates input is between 1-4

Saving to CSV

Method Declaration

Define the saveToCSV method with filename and quiz parameters



Create File Writer

Open a connection to the specified CSV file



Iterate Through Questions

Loop through each question in the quiz



Format and Write Data

Convert each question to CSV format and write to file



Close Resources

Properly close the file connection



```
public static void saveToCSV(String filename, Quiz quizis)
```

This saves the quiz to a file called quiz.csv, with each question written like this:

```
question,option1,option2,option3,option4,correctOptionNumber
```

Loading from CSV

1

Read File

Open and read the CSV file line by line

2

Parse Data

Split each line into components using commas

3

Create Objects

Convert CSV data into Question objects

4

Return Collection

Return the complete ArrayList of questions

```
enb: 1 case

aptunal Weassenill)

< it: lisse case >

< sy: by the case(i))

< inb: 1 case case,

< tiles case accsall prarchi()

:: ilke case, lacking, return()

staile! thy the set (egs)

spethacopl.

>: cillesct i) terray (egs)

< the thg whitee, Stating tax)

(fobuss contatid wanc is merrang merrang)

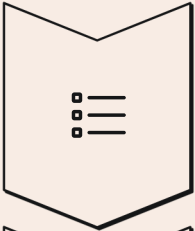
< casel thg by east, class)

elles thiset the entere from, the merrang

the tim. thres acates).

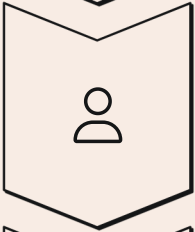
staelabros, merrang, merrang
```

Taking the Quiz



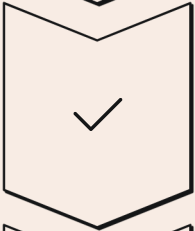
Display Questions

Show each question with its options.



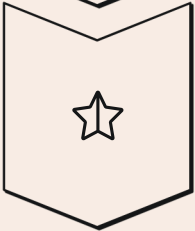
User Input

The user selects an answer from the options.



Check Answer

Program verifies answer correctness.



Calculate Score

The system calculates the quiz score.

Operation Tracking (Report)

HashMap for Actions

Tracks how many times each action was performed.

Operation Method

Records action frequency.

Show Report

Prints operation statistics.

We use a static HashMap called operations to track actions.

The Operation() method records it, and showReport() prints the stats.

Thank you for your attention!