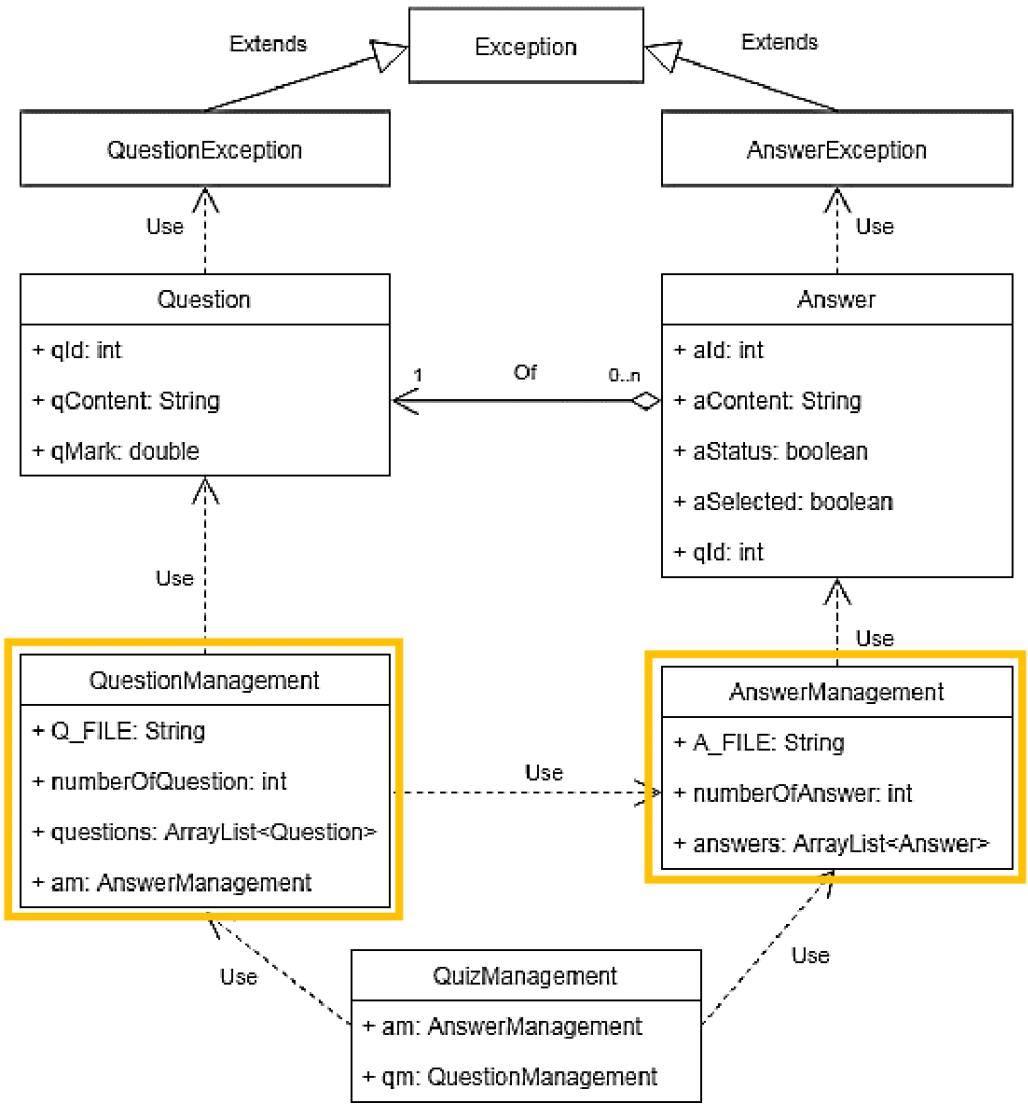
Session 08

Basic IO (1)

A. STEPS BY STEPS

I. Problem definition

Multiple choice testing is an effective way to get students' level. Your task is to build a simple testing system as below:



Special requirements:

- The data of questions and answers must be **saved** after each run and **reloaded** at the next run **automatically**.
- User can add new question and answer, view question bank and testing.
- Questions and answers of quiz can be **shuffled**.

In this session, you will create the second part of **QuizManagement** Project includes:

- **AnswerManagement**
- **QuestionManagement**

AnswerManagement.java

```
package quizmanagement;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;

/**
 * The class AnswerManagement is used to manage answer bank
 * @author KhanhVH@fe.edu.vn
 */
public class AnswerManagement {
    private String A_FILE;              //The URL of data file that stores all answers
    private int numberOfAnswer;         //Number of answers that stored in data file
    private ArrayList<Answer> answers;  //All instances of answers

    /**
     * Creates instance for answer management
     * @param A_FILE
     * @throws quizmanagement.AnswerException
     */
    public AnswerManagement(String A_FILE) throws AnswerException {
        if (A_FILE.equals("")) {
            throw new AnswerException("The URL of answer data file can't be empty!");
        } else {
            this.A_FILE = A_FILE;                   //Inits the URL of data file thats stores answer bank
            this.answers = new ArrayList<Answer>(); //Creates empty answer bank
            this.numberOfAnswer = 0;                //So, the number of answer is 0
        }
    }
```

```
/**
 * Loads data of answers from data file and stored it into ArrayList
 * @throws IOException
 * @throws AnswerException
 */
public void loadAnswers() throws IOException, AnswerException {
    File aFile = new File(A_FILE);

    if(!aFile.exists()){       //Checks is file created
        aFile.createNewFile(); //If not, creates new file
        System.out.println("The data file answers.txt is not exits. " +
                           "Creating new data file answers.txt... " +
                           "Done!");
        this.numberOfAnswer = 0; //New data file with the number of answer is 0
    }else{
        //If file is existed, so loading this data file
        System.out.print("\nThe data file answers.txt is found. " +
                         "Data of answers is loading...");
        //Loads text file into buffer
        try (BufferedReader br = new BufferedReader(new FileReader(A_FILE))) {
            String qId, aId, aContent, aStatus;

            this.numberOfAnswer = Integer.parseInt(br.readLine()); //Reads number of answers

            for (int i = 0; i < this.numberOfAnswer; i++) {
                //Reads answer's information
                aId      = br.readLine();
                aContent = br.readLine();
                aStatus  = br.readLine();
                qId      = br.readLine();

                //Create new instance of Answer and adds to answer bank
                this.answers.add(new Answer(Integer.parseInt(aId),
                                            aContent,
                                            Boolean.parseBoolean(aStatus),
                                            Integer.parseInt(qId)));
            }
        }
        System.out.println("Done! [" + this.numberOfAnswer + " answers]");
    }
}
```

```java
/**
 * Adds new answer to answer bank
 * @param aContent
 * @param aStatus
 * @param qId
 * @return
 * @throws AnswerException
 */
public int addAnswer(String aContent, boolean aStatus, int qId) throws AnswerException {
    this.answers.add(new Answer(++this.numberOfAnswer, aContent, aStatus, qId));
    return this.numberOfAnswer; //answer id
}

/**
 * Finds answer by answer id and return the index of this answer
 * @param aId
 * @return
 */
public int findAnswer(int aId) {
    for (int i = 0; i < this.answers.size(); i++) {
        Answer a = this.answers.get(i);
        if (a.getAId() == aId) {
            return i;
        }
    }
    return -1;
}

/**
 * Finds the answer instance by answer id
 * @param aId
 * @return
 */
public Answer getAnswer(int aId) {
    int idx = this.findAnswer(aId);
    if (idx == -1) {
        return null;
    } else {
        return this.answers.get(idx);
    }
}
```

```
/**
 * Saves answer bank (ArrayList) into data file
 * @throws IOException
 */
public void saveAnswers() throws IOException {
    //Overwrite data file
    FileWriter fw = new FileWriter(new File(A_FILE), false);

    try {
        System.out.print("\nAnswers is saving into data file answers.txt...");

        //Writes number of answer
        fw.append(String.valueOf(this.numberOfAnswer) + "\n");

        for(int i=0; i<this.numberOfAnswer; i++){
            //Inits answer's information
            int aId           = this.answers.get(i).getAId();
            String aContent   = this.answers.get(i).getAContent();
            boolean aStatus   = this.answers.get(i).getAStatus();
            int qId           = this.answers.get(i).getQId();

            //Writes answer's information into data file
            fw.append(String.valueOf(aId) + "\n");
            fw.append(aContent + "\n");
            fw.append(String.valueOf(aStatus) + "\n");
            fw.append(String.valueOf(qId) + "\n");
        }
    } finally {
        //Saves data file (from RAM into HDD)
        fw.close();
        System.out.println("Done! [" + this.numberOfAnswer + " answers]");
    }
}
```

```
    /**
     * Gets all answer that belongs to question that identifies by question id
     * @param qId
     * @return
     */
    public ArrayList<Answer> getAnswers(int qId, boolean isShuffle) {
        ArrayList<Answer> aList = new ArrayList<Answer>();

        for (int i = 0; i < this.answers.size(); i++) {
            Answer a = this.answers.get(i);
            if (a.getQId() == qId) {
                aList.add(a);
            }
        }

        //Inits the index of all answer
        int[] idx = new int[aList.size()];
        for (int i = 0; i < aList.size(); i++) {
            idx[i] = i;
        }

        if (isShuffle) { //if the random mode is turned on
            int newIdx, tmp;
            Random ran = new Random();

            //Randomizes indexes of answer bank
            for (int i = 0; i < aList.size(); i++) {
                newIdx = ran.nextInt(aList.size());
                tmp = idx[i];
                idx[i] = idx[newIdx];
                idx[newIdx] = tmp;
            }
        }

        ArrayList<Answer> result = new ArrayList<Answer>();
        for (int i = 0; i < aList.size(); i++) {
            result.add(aList.get(idx[i]));
        }
        return result;
    }
}
```

QuestionManagement.java

```
package quizmanagement;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;

/**
 * The class QuestionManagement is used to manage question bank
 * @author KhanhVH@fe.edu.vn
 */
public class QuestionManagement {
    private String Q_FILE;                  //The URL of data file that stores all questions
    private int numberOfQuestion;           //Number of questions that stored in data file
    private ArrayList<Question> questions;  //All instances of questions
    private AnswerManagement am;            //Instance of AnswerManagement

    /**
     * Creates instance for question management
     * @param Q_FILE
     * @param am
     * @throws quizmanagement.QuestionException
     */
    public QuestionManagement(String Q_FILE, AnswerManagement am) throws QuestionException {
        if (Q_FILE.equals("")) {
            throw new QuestionException("The URL of answer data file can't be empty!");
        } else {
            this.Q_FILE = Q_FILE; //Inits the URL of data file thats stores question bank

            this.questions = new ArrayList<Question>(); //Creates empty question bank

            this.numberOfQuestion = 0; //So, the number of question is 0

            this.am = am; //Inits the answer management
        }
    }
```

```
/**
 * Loads data of questions from data file and stored it into ArrayList
 * @throws IOException
 * @throws QuestionException
 */
public void loadQuestions() throws IOException, QuestionException {
    File qFile = new File(Q_FILE);

    if(!qFile.exists()){ //Checks is file created
        qFile.createNewFile(); //If not, creates new file
        System.out.println("The data file questions.txt is not exits. " +
                           "Creating new data file questions.txt..." +
                           "Done!");
        this.numberOfQuestion = 0; //New data file with the number of question is 0
    }else{
        //If file is existed, so loading this data file
        System.out.print("\nThe data file questions.txt is found. " +
                         "Data of questions is loading...");

        BufferedReader br = new BufferedReader(new FileReader(Q_FILE)); //Loads text file into buffer
        try {
            String qId, qContent, qMark;

            this.numberOfQuestion = Integer.parseInt(br.readLine()); //Reads number of answers
            for (int i = 0; i < this.numberOfQuestion; i++) {
                //Reads answer's information
                qId      = br.readLine();
                qContent = br.readLine();
                qMark    = br.readLine();

                //Create new instance of Answer and adds to answer bank
                this.questions.add(new Question(Integer.parseInt(qId),
                                                Double.parseDouble(qMark),
                                                qContent));
            }
        } finally {
            br.close();
        }
        System.out.println("Done! [" + this.numberOfQuestion + " questions]");
    }
}
```

```
/**
 * Gets number of questions
 * @return
 */
public int getSize() {
    return this.numberOfQuestion;
}



/**
 * Adds new question to question bank
 * @param qMark
 * @param qContent
 * @return
 * @throws QuestionException
 */
public int addQuestion(double qMark, String qContent) throws QuestionException {

    this.questions.add(new Question(++this.numberOfQuestion, qMark, qContent));

    return this.numberOfQuestion;
}



/**
 * Finds question by question id and return the index of this question
 * @param qId
 * @return
 */
public int findQuestion(int qId) {
    for (int i = 0; i < this.questions.size(); i++) {
        Question q = this.questions.get(i);
        if (q.getQId() == qId) {
            return i;
        }
    }
    return -1;
}
```

```java
/**
 * Finds the question instance by question id
 * @param qId
 * @return
 */
public Question getQuestion(int qId) {
    int idx = this.findQuestion(qId);
    if (idx == -1) {
        return null;
    } else {
        return this.questions.get(idx);
    }
}

/**
 * Saves question bank (ArrayList) into data file
 * @throws IOException
 */
public void saveQuestions() throws IOException {
    FileWriter fw = new FileWriter(new File(Q_FILE), false); //Overwrite data file

    try {
        System.out.print("\nQuestions is saving into data file questions.txt...");

        fw.append(String.valueOf(this.numberOfQuestion) + "\n"); //Writes number of question
        for(int i=0; i<this.numberOfQuestion; i++) {
            //Inits question's information
            int qId         = this.questions.get(i).getQId();
            String qContent = this.questions.get(i).getQContent();
            double qMark    = this.questions.get(i).getQMark();

            //Writes quesiton's information into data file
            fw.append(String.valueOf(qId) + "\n");
            fw.append(qContent + "\n");
            fw.append(String.valueOf(qMark) + "\n");
        }
    } finally {
        fw.close(); //Saves data file (from RAM to HDD)
        System.out.println("Done! [" + this.numberOfQuestion + " questions]");
    }
}
```

```java
/**
 * Checks that the user's answer is correct or incorrect
 * @param qId
 * @param answers
 * @return
 */
public boolean isQuestionCorrect(int qId, ArrayList<Answer> answers) {
    boolean isCorrect = true;

    for (int i = 0; i < answers.size(); i++) {
        //the answer of user is correct even if the user's selected is the same with answer's status
        isCorrect = isCorrect && answers.get(i).isCorrect();
    }

    return isCorrect;
}

/**
 * Gets the question formatted string that includes question content and all answers
 *      that comes with random mode
 * @param qId
 * @param isShuffle turn on/off random display answer mode
 * @return
 */
public String showQuestion(int qId, boolean isShuffle) {
    Question q = getQuestion(qId);
    ArrayList<Answer> aList = am.getAnswers(qId, isShuffle);

    String str = "";

    str += q.toString();

    char aNo = 'a';

    for (int i = 0; i < aList.size(); i++, aNo++) {
        str += "    " + aNo + ". " + aList.get(i).toString();
    }

    return str;
}
```

```
/**
 * Gets the question formatted string that includes question content and all answers
 *      that comes with a list of answers
 * @param qId
 * @param aList
 * @return
 */
public String showQuestion(int qId, ArrayList<Answer> aList) {
    Question q = getQuestion(qId);

    String str = "";

    str += q.toString();

    char aNo = 'a';

    for (int i = 0; i < aList.size(); i++, aNo++) {
        str += "   " + aNo + ". " + aList.get(i).toString();
    }

    return str;
}

/**
 * Displays all question of question bank
 */
public void showQuestionBank() {
    int qNo = 1;

    for (int i = 0; i < this.questions.size(); i++, qNo++) {
        Question q = this.questions.get(i);

        System.out.println(qNo + ". " + showQuestion(q.getQId(), false));
    }
}
```

```
    /**
     * Gets the first qNumber of question bank
     * @param qNumber
     * @param isShuffle
     * @return
     */
    public ArrayList<Question> getQuestionBank(int qNumber, boolean isShuffle) {
        ArrayList<Question> qList = new ArrayList<Question>();

        //Inits the index of all answer
        int[] idx = new int[questions.size()];

        for (int i = 0; i < questions.size(); i++) {
            idx[i] = i;
        }

        if (isShuffle) { //if the random mode is turned on
            int newIdx, tmp;
            Random ran = new Random();

            //Randomizes indexes of answer bank
            for (int i = 0; i < questions.size(); i++) {
                newIdx = ran.nextInt(questions.size());

                tmp = idx[i];
                idx[i] = idx[newIdx];
                idx[newIdx] = tmp;
            }
        }

        for (int i = 0; i < qNumber; i++) {
            qList.add(questions.get(idx[i]));
        }

        return qList;
    }
}
```

Example of runtime

```
run:
The data file answers.txt is found. Data of answers is loading...Done! [28 answers]
The data file questions.txt is found. Data of questions is loading...Done! [7 questions]


---------- QUIZ MANAGEMENT ----------
1. Add question.
2. Show question bank.
3. Create quiz.
4. Quit.


   Please select a function: 1
----- QUIZ MANAGEMENT [ADD NEW QUESTION] -----
Please enter content of question: What is the HTML element to display an image?
Please enter mark of question: 1
Your question is created!
+++ [ADD ANSWERS FOR QUESTION] +++
... Answer 1 ...
Please enter content of answer 1: <image>
Is this answer True or False? (True/False) False
```

```
Do you want to add more answer? (Yes/No) Yes
... Answer 2 ...
Please enter content of answer 2: <picture>
Is this answer True or False? (True/False) False

Do you want to add more answer? (Yes/No) Error: You must type 'Yes' or 'No'!
Do you want to add more answer? (Yes/No) Yes
... Answer 3 ...
Please enter content of answer 3: <pic>
Is this answer True or False? (True/False) fal
Error: You must type 'True' or 'False'!
Is this answer True or False? (True/False) False
Do you want to add more answer? (Yes/No) Yes
... Answer 4 ...
Please enter content of answer 4: <img>
Is this answer True or False? (True/False) True
Do you want to add more answer? (Yes/No) No

---------- QUIZ MANAGEMENT ----------
1. Add question.
```

```
2. Show question bank.

3. Create quiz.

4. Quit.


   Please select a function: 2

----- QUIZ MANAGEMENT [QUESTION BANK] (8 questions) -----

1. Which of the following is an example of web browser?

   a. Google

   b. Bing

   c. Baidu

   d. Opera


2. Which of the following HTML tag is used to link the URL?

   a. <style>

   b. <link>

   c. <a>

   d. <hyperlink>


3. Link URL in HTML is specified using _____ attribute

   a. src
```

b. link

c. rel

d. href

4. HTML is considered as

a. High Level Language

b. OOP Language

c. Programming Language

d. Markup Language

5. _____ attribute is used to specify where to open the linked document

a. target

b. coords

c. rel

d. none of the others

6. Which of the anchor attribute is used to specify the language of the linked document?

a. alang

b. lang

c. hreflang

```
   d. all of the others


7. Which of the following is used to open the document in new window?

   a. <a target="_top">Link</a>

   b. <a target="_blank">Link</a>

   c. <a target="_new">Link</a>

   d. <a target="_parent">Link</a>


8. What is the HTML element to display an image?

   a. <image>

   b. <picture>

   c. <pic>

   d. <img>



---------- QUIZ MANAGEMENT ----------

1. Add question.

2. Show question bank.

3. Create quiz.

4. Quit.
```

```
   Please select a function: 3
----- QUIZ MANAGEMENT [EXAMINATION] (8 questions) -----
How many question of the test: 3
Do you want to shuffle the test? (True/False) True
+++ The test is generating... Done! +++

###################
#     TESTING     #
###################
1. Which of the following HTML tag is used to link the URL?
   a. <style>
   b. <a>
   c. <hyperlink>
   d. <link>
   >>> Please select answer: b
   +++ Congratulation! Your answer is CORRECT!!!

###################
2. _____ attribute is used to specify where to open the linked document
```

```
   a. target
   b. rel
   c. coords
   d. none of the others
   >>> Please select answer: a
   +++ Congratulation! Your answer is CORRECT!!!

###################
3. Which of the following is used to open the document in new window?
   a. <a target="_parent">Link</a>
   b. <a target="_top">Link</a>
   c. <a target="_new">Link</a>
   d. <a target="_blank">Link</a>
   >>> Please select answer: d
   +++ Congratulation! Your answer is CORRECT!!!

++++++++++++++++++++
You are FINISH!!!
Correct rate is 3/3 (100.00%)
Total mark is 3.00/3.00 (100.00%)
```

```
---------- QUIZ MANAGEMENT ----------
1. Add question.
2. Show question bank.
3. Create quiz.
4. Quit.

   Please select a function: 4
--------------------
Thank for using our software!
See you again!
Answers is saving into data file answers.txt...Done! [32 answers]
Questions is saving into data file questions.txt...Done! [8 questions]
BUILD SUCCESSFUL (total time: 2 minutes 39 seconds)
```