**Designing The TakeTestFrame**

****

**STEPS TO BE DONE IN** **TakeTestFrame**

In the **TakeTestFrame** we need to do following steps:

1. Handle the constructor for proper initialization of class members

2.In the constructor we also have to **load** the questions from the Database and **display** the first question as well as it's options

3. Write code for the "**Next**" Button . When this button is clicked it should:

**a. Create an Answer POJO object and initialize it with ExamId,Language,Qno,UserAnswer and CorrectAnswer**

**b. Store or replace the Answer object in the astore i.e. AnswerStore**

**c. Load and show the next question**

5. Write code for the "**Previous**" Button . When this button is clicked it should:

**a. Create an Answer POJO object and initialize it with ExamId,Language,Qno,UserAnswer and CorrectAnswer**

**b. Store or replace the Answer object in the astore i.e. AnswerStore**

**c. Load and show the previous question**

6. Write code for the "**Done**" Button . When this button is clicked it should:

**a. Create an Answer POJO object and initialize it with ExamId,Language,Qno,UserAnswer and CorrectAnswer**

**b. Store or replace the Answer object in the astore i.e. AnswerStore**

**c. Calculate unattempted, right and wrong answers given by the student**

**d.Create an object of Performance POJO , initialize it with exam details,username and user's performance.**

**e. Add Performance POJO to the PERFORMANCE table using PerformanceDAO.**

**f. Open ChoosePaperFrame and dispose current frame.**

7. Write code for the "**Cancel**" Button . When this button is clicked it should:

a**. Dispose the current frame and open the ChoosePaperFrame**

**THE TABLES USED IN TakeTestFrame**

**1. QUESTIONS**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **EXAMID** | **Varchar2(10)** | **Contains id of the paper** |
| **QNO** | **Number(5)** | **Represents the Question No.** |
| **QUESTION** | **Varchar2(100)** | **Represents Question Description** |
| **ANSWER1** | **Varchar2(30)** | **Option 1** |
| **ANSWER2** | **Varchar2(30)** | **Option 2** |
| **ANSWER3** | **Varchar2(30)** | **Option 3** |
| **ANSWER4** | **Varchar2(30)** | **Option 4** |
| **CORRECT\_ANSWER** | **Varchar2(10)** | **The actual correct answer** |
| **LANGUAGE** | **Varchar2(10)** | **Programming language name** |

**2.PERFORMANCE**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **USERID** | **Varchar2(10)** | **Contains Admin Provided User ID** |
| **EXAMID** | **Varchar2(10)** | **Contains id of the paper** |
| **RIGHT** | **Number(4)** | **Number of correct answers** |
| **WRONG** | **Number(4)** | **Number of wrong answers** |
| **UNATTEMPTED** | **Number(5)** | **Number of unattempted questions** |
| **PER** | **Number(5,2)** | **Percentage obtained** |
| **LANGUAGE** | **Varchar2(10)** | **Programming language name** |

**THE POJO CLASSES USED IN TakeTestFrame**

1. The **Answer** POJO

2. The **AnswerStore** POJO

3. The **Question** POJO

4. The **QuestionStore** POJO

5.The **Exam** POJO

6.The **Performance** POJO

**1. The Answer POJO**

***public class Answer {***

***private String examId;***

***private String subject;***

***private int qno;***

***private String chosenAnswer;***

***private String correctAnswer;***

***@Override***

***public String toString() {***

***return "Answer{" + "examId=" + examId + ", subject=" + subject + ", qno=" + qno + ", chosenAnswer=" + chosenAnswer + ", correctAnswer=" + correctAnswer + '}';***

***}***

***@Override***

***public boolean equals(Object obj) {***

***Answer other = (Answer) obj;***

***if (this.examId.equals(other.examId)==false) {***

***return false;***

***}***

***if (this.subject.equals(other.subject)==false) {***

***return false;***

***}***

***if (this.qno != other.qno) {***

***return false;***

***}***

***if (this.chosenAnswer.equals(other.chosenAnswer)==false) {***

***return false;***

***}***

***if (this.correctAnswer.equals(other.correctAnswer)==false) {***

***return false;***

***}***

***return true;***

***}***

***public String getExamId() {***

***return examId;***

***}***

***public void setExamId(String examId) {***

***this.examId = examId;***

***}***

***public String getSubject() {***

***return subject;***

***}***

***public void setSubject(String subject) {***

***this.subject = subject;***

***}***

***public int getQno() {***

***return qno;***

***}***

***public void setQno(int qno) {***

***this.qno = qno;***

***}***

***public String getChosenAnswer() {***

***return chosenAnswer;***

***}***

***public void setChosenAnswer(String chosenAnswer) {***

***this.chosenAnswer = chosenAnswer;***

***}***

***public String getCorrectAnswer() {***

***return correctAnswer;***

***}***

***public void setCorrectAnswer(String correctAnswer) {***

***this.correctAnswer = correctAnswer;***

***}***

***public Answer(String examId, String subject, int qno, String chosenAnswer, String correctAnswer) {***

***this.examId = examId;***

***this.subject = subject;***

***this.qno = qno;***

***this.chosenAnswer = chosenAnswer;***

***this.correctAnswer = correctAnswer;***

***}***

***}***

**The AnswerStore POJO**

***public class AnswerStore {***

***ArrayList <Answer> answerList;***

***public AnswerStore(){***

***answerList=new ArrayList<>();***

***}***

***public void addAnswer(Answer answer){***

***answerList.add(answer);***

***}***

***public Answer getAnswer(int pos){***

***return answerList.get(pos);***

***}***

***public void removeAnswer(int pos){***

***answerList.remove(pos);***

***}***

***public void setAnswerAt(int pos,Answer answer){***

***answerList.add(pos, answer);***

***}***

***public ArrayList<Answer> getAllAnswers(){***

***return answerList;***

***}***

***public int getCount(){***

***return answerList.size();***

***}***

***public Answer getAnswerByQno(int qno){***

***for(Answer answer: answerList){***

***if(answer.getQno()==qno)***

***return answer;***

***}***

***return null;***

***}***

***public int removeAnswer(Answer answer){***

***int pos=answerList.indexOf(answer);***

***answerList.remove(pos);***

***return pos;***

***}***

***}***

**The Performance POJO**

***public class Performance {***

***private String examId;***

***private String language;***

***private String userId;***

***private int right;***

***private int wrong;***

***private int unattempted;***

***private double per;***

***public String getExamId() {***

***return examId;***

***}***

***public void setExamId(String examId) {***

***this.examId = examId;***

***}***

***public String getLanguage() {***

***return language;***

***}***

***public void setLanguage(String language) {***

***this.language = language;***

***}***

***public String getUserId() {***

***return userId;***

***}***

***public void setUserId(String userId) {***

***this.userId = userId;***

***}***

***public int getRight() {***

***return right;***

***}***

***public void setRight(int right) {***

***this.right = right;***

***}***

***public int getWrong() {***

***return wrong;***

***}***

***public void setWrong(int wrong) {***

***this.wrong = wrong;***

***}***

***public int getUnattempted() {***

***return unattempted;***

***}***

***public void setUnattempted(int unattempted) {***

***this.unattempted = unattempted;***

***}***

***public double getPer() {***

***return per;***

***}***

***public void setPer(double per) {***

***this.per = per;***

***}***

***public Performance(String examId, String language, String userId, int right, int wrong, int unattempted, double per) {***

***this.examId = examId;***

***this.language = language;***

***this.userId = userId;***

***this.right = right;***

***this.wrong = wrong;***

***this.unattempted = unattempted;***

***this.per = per;***

***}***

***}***

**THE DAO CLASSES USED IN TakeTestFrame**

1. The **QuestionDAO**

2. The **PerformanceDAO**

***public static void addPerformance(Performance performance)throws SQLException{***

***String qry="Insert into performance values(?,?,?,?,?,?,?)";***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement(qry);***

***ps.setString(1, performance.getUserId());***

***ps.setString(2, performance.getExamId());***

***ps.setInt(3,performance.getRight());***

***ps.setInt(4,performance.getWrong());***

***ps.setInt(5,performance.getUnattempted());***

***ps.setDouble(6,performance.getPer());***

***ps.setString(7,performance.getLanguage());***

***ps.executeUpdate();***

***}***

**HOW TO LOAD AND SHOW THE QUESTIONS:**

The first step we need to do in the **constructor** of **TakeTestFrame** is to load all the questions from the database and show the first question.

To do this we will take 4 steps:

**a. Create a method called loadQuestions( ) in the TakeTestFrame and from this method we will call the method getQuestionsByExamId( ) of the QuestionDAO whose prototype is:**

***public static ArrayList <Question> getQuestionsByExamId(String examId)throws SQLException***

**b. Store all the Question objects in qstore.**

Following is the code:

***public void loadQuestions() {***

***try{***

***ArrayList<Question> questionList=QuestionDAO.getQuestionsByExamId(exam.getExamId());***

***for(Question obj:questionList){***

***qstore.addQuestion(obj);***

***}***

***}***

***catch(SQLException ex){***

***JOptionPane.showMessageDialog(null, "Error while connecting to DB!","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

**c. Create a method called showQuestion( ) and show the first question**

***public void showQuestion(){***

***Question question=qstore.getQuestion(pos);***

***buttonGroup1.clearSelection();***

***txtQuestion.setText(question.getQuestion());***

***jrOption1.setText(question.getAnswer1());***

***jrOption2.setText(question.getAnswer2());***

***jrOption3.setText(question.getAnswer3());***

***jrOption4.setText(question.getAnswer4());***

***Answer answer=astore.getAnswerByQno(qno);***

***if(answer==null)***

***return;***

***String chosenAnswer=answer.getChosenAnswer();***

***switch(chosenAnswer){***

***case "Answer1":***

***jrOption1.setSelected(true);***

***break;***

***case "Answer2":***

***jrOption2.setSelected(true);***

***break;***

***case "Answer3":***

***jrOption3.setSelected(true);***

***break;***

***case "Answer4":***

***jrOption4.setSelected(true);***

***break;***

***}***

***}***

**d. Call both the methods from the constructor.**

***public class TakeTestFrame extends javax.swing.JFrame {***

***private Exam exam;***

***private QuestionStore qstore;***

***private AnswerStore astore;***

***private int qno,pos=0;***

***public TakeTestFrame() {***

***initComponents();***

***this.setLocationRelativeTo(null);***

***lblUsername.setText("Hello "+UserProfile.getUSername());***

***qstore=new QuestionStore();***

***astore=new AnswerStore();***

***qno=1;***

***lblQno.setText(lblQno.getText()+qno);***

***}***

***public TakeTestFrame(Exam exam){***

***this();***

***this.exam=exam;***

***lblTitle.setText(exam.getLanguage().toUpperCase()+" PAPER");***

***loadQuestions();***

***showQuestion();***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN TakeTestFrame**

***private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {***

***String chosenAnswer=getUserAnswer();***

***if(chosenAnswer!=null){***

***Question question=qstore.getQuestion(pos);***

***String correctAnswer=question.getCorrectAnswer();***

***Answer newanswer=new Answer(exam.getExamId(),exam.getLanguage(),qno,***

***chosenAnswer,correctAnswer);***

***System.out.println(newanswer);***

***Answer answer=astore.getAnswerByQno(qno);***

***if(answer==null)***

***astore.addAnswer(newanswer);***

***else{***

***if(newanswer.getChosenAnswer().equals(***

***answer.getChosenAnswer())==false){***

***int apos=astore.removeAnswer(answer);***

***astore.setAnswerAt(apos, newanswer);***

***}***

***}***

***}***

***pos++;***

***if(pos>=qstore.getCount())***

***{***

***pos=0;***

***}***

***qno++;***

***if(qno>qstore.getCount())***

***qno=1;***

***lblQno.setText("Question no:"+qno);***

***showQuestion();***

***}***

***public String getUserAnswer(){***

***if(jrOption1.isSelected())***

***return "Answer1";***

***else if(jrOption2.isSelected())***

***return "Answer2";***

***else if(jrOption3.isSelected())***

***return "Answer3";***

***else if(jrOption4.isSelected())***

***return "Answer4";***

***else***

***return null;***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN TakeTestFrame**

***private void btnPreviousActionPerformed(java.awt.event.ActionEvent evt) {***

***String chosenAnswer=getUserAnswer();***

***if(chosenAnswer!=null){***

***Question question=qstore.getQuestion(pos);***

***String correctAnswer=question.getCorrectAnswer();***

***Answer newanswer=new Answer(exam.getExamId(),exam.getLanguage(),qno,chosenAnswer,correctAnswer);***

***System.out.println(newanswer);***

***Answer answer=astore.getAnswerByQno(qno);***

***if(answer==null)***

***astore.addAnswer(newanswer);***

***else{***

***if(newanswer.getChosenAnswer().equals(answer.getChosenAnswer())==false){***

***int apos=astore.removeAnswer(answer);***

***astore.setAnswerAt(apos, newanswer);***

***}***

***}***

***}***

***pos--;***

***if(pos<0)***

***{***

***pos=qstore.getCount()-1;***

***}***

***qno--;***

***if(qno<=0)***

***qno=qstore.getCount();***

***lblQno.setText("Question no:"+qno);***

***showQuestion();***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN TakeTestFrame**

***private void btnDoneActionPerformed(java.awt.event.ActionEvent evt) {***

***String userAnswer=getUserAnswer();***

***if(userAnswer!=null){***

***Question question=qstore.getQuestion(pos);***

***String correctAnswer=question.getCorrectAnswer();***

***Answer newanswer=new Answer(exam.getExamId(),exam.getLanguage(),qno,userAnswer,correctAnswer);***

***System.out.println(newanswer);***

***Answer answer=astore.getAnswerByQno(qno);***

***if(answer==null)***

***astore.addAnswer(newanswer);***

***else{***

***if(newanswer.getChosenAnswer().equals(answer.getChosenAnswer())==false){***

***int apos=astore.removeAnswer(answer);***

***astore.setAnswerAt(apos, newanswer);***

***}***

***}***

***}***

***int right=0,wrong=0;***

***for(Question question:qstore.getAllQuestions()){***

***int qno=question.getQno();***

***Answer answer=astore.getAnswerByQno(qno);***

***if(answer==null)***

***continue;***

***else***

***{***

***String chosenAnswer1=answer.getChosenAnswer();***

***String correctAnswer=answer.getCorrectAnswer();***

***if(chosenAnswer1.equals(correctAnswer))***

***++right;***

***else***

***++wrong;***

***}***

***}***

***String reportCard="Total Questions:"+qstore.getCount();***

***reportCard+="\nRight Answers:"+right;***

***reportCard+="\nWrong Answers:"+wrong;***

***reportCard+="\nUnAttempted :"+(qstore.getCount()-(right+wrong));***

***JOptionPane.showMessageDialog(null, reportCard,"Your Result!",JOptionPane.INFORMATION\_MESSAGE);***

***try{***

***Performance performance=new Performance(exam.getExamId(),exam.getLanguage(),***

***UserProfile.getUSername(),right,wrong,***

***(qstore.getCount()-(right+wrong)),***

***(double)right/qstore.getCount()\*100);***

***PerformanceDAO.addPerformance(performance);***

***JOptionPane.showMessageDialog(null, "Your performance has been successfully added to the database"," Performance Added!",JOptionPane.INFORMATION\_MESSAGE);***

***ChoosePaperFrame paperFrame=new ChoosePaperFrame();***

***paperFrame.setVisible(true);***

***this.dispose();***

***}***

***catch(SQLException ex){***

***JOptionPane.showMessageDialog(null, "Error while connecting to DB!","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN TakeTestFrame**

***private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {***

***ChoosePaperFrame paperFrame=new ChoosePaperFrame();***

***paperFrame.setVisible(true);***

***this.dispose();***

***}***