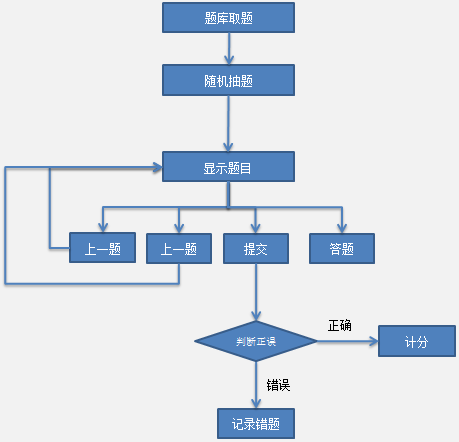
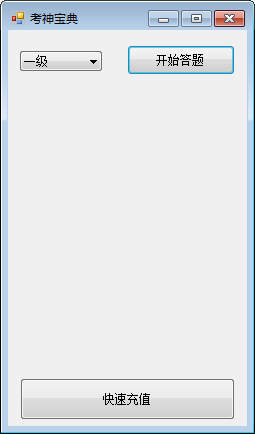
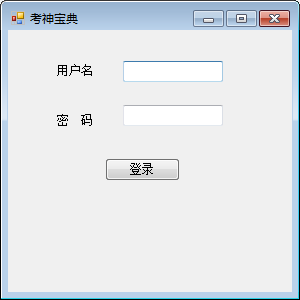
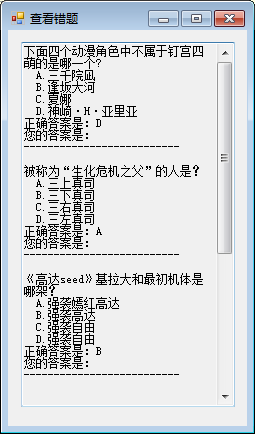
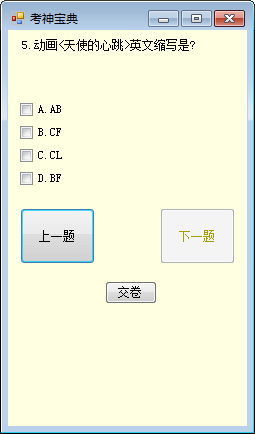
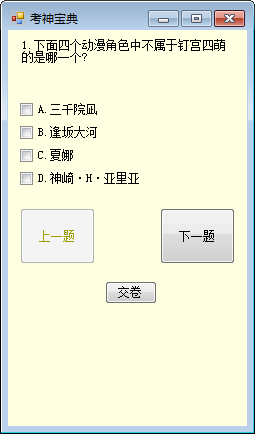
# 核心流程



# 运行截图





# 数据库

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **列** | **ID（主键）** | **题目** | **选项A** | **选项B** | **选项C** | **选项D** | **正确答案** | **备注** |
| **类型** | 自动编号 | 文本 | 文本 | 文本 | 文本 | 文本 | 文本 | 文本 |
| **内容示例** | 1 | 1+1=？ | 1 | 2 | 3 | 0 | B |  |

# 随机选题

## 数据访问层

|  |
| --- |
| package org.ah.dao;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  public class BaseDao {  // public static final String DRIVER = "com.mysql.jdbc.Driver";  // public static final String URL = "jdbc:mysql://127.0.0.1:3306/test";  public static final String URL = "jdbc:mysql://127.0.0.1/test";  public static final String USER\_NAME = "root";  public static final String USER\_PWD = "root";  public Connection conn = null;  public PreparedStatement pst = null;  public ResultSet rs = null;  public void getConn() {  try {  // Class.forName(DRIVER);// 指定连接类型  conn = DriverManager.getConnection(URL, USER\_NAME, USER\_PWD);// 获取连接  } catch (Exception e) {  e.printStackTrace();  }  }  public void closeConn() {  try {  if (rs != null) {  rs.close();  rs = null;  }  if (pst != null) {  pst.close();  pst = null;  }  if (conn != null) {  conn.close();  conn = null;  }  } catch (SQLException e) {  e.printStackTrace();  }  }  } |
| package org.ah.dao;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import org.ah.core.bean.Question;  public class QuestionDao extends BaseDao {  public static void main(String[] args) {  QuestionDao dao = new QuestionDao();  dao.getAllQuestions();  }  public List<Question> getAllQuestions() {  List<Question> lst = new ArrayList<Question>();  super.getConn();  try {  PreparedStatement pStat = super.conn  .prepareStatement("SELECT \* FROM question");  ResultSet rs = pStat.executeQuery();  while (rs.next()) {  int nId = rs.getInt(1);  String sTitle = rs.getString(2);  String sA = rs.getString(3);  String sB = rs.getString(4);  String sC = rs.getString(5);  String sD = rs.getString(6);  String sAnser = rs.getString(7);  // 使用对象封装数据  Question q = new Question(nId, sTitle, sA, sB, sC, sD, sAnser);  lst.add(q);  }  } catch (SQLException e) {  e.printStackTrace();  } finally {  super.closeConn();  }  return lst;  }  } |

## 业务逻辑层

|  |
| --- |
| package org.ah.core;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import java.util.Random;  import java.util.Scanner;  import org.ah.core.bean.Question;  import org.ah.dao.QuestionDao;  public class Bus {  **/\*\***  **\* 随机取题**  **\***  **\* @param sourceQs题集源**  **\* @param count取多少道题**  **\* @return 新题集**  **\*/**  **public static List<Question> extractQuestions(List<Question> sourceQs,**  **int count) {**  **Random r = new Random();**  **int n随机数 = -1;**  **List<Question> newQs = new ArrayList<Question>();**  **for (int i = 1; i <= count; i++) {**  **n随机数 = r.nextInt(sourceQs.size());**  **Question q选中的题 = sourceQs.get(n随机数);**  **newQs.add(q选中的题);**  **sourceQs.remove(n随机数);// 防止重复选中同一道题**  **}**  **return newQs;**  **}**  public static void main(String[] args) throws IOException {  Scanner sc = new Scanner(System.in);  QuestionDao dao = new QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  List<Question> newQs = extractQuestions(qs全部題目, 5);  int newId = 1;  for (Question q : newQs) {  System.out.println((newId++) + "." + q.getTitle());  System.out.println("\tA." + q.getOptA());  System.out.println("\tB." + q.getOptB());  System.out.println("\tC." + q.getOptC());  System.out.println("\tD." + q.getOptD());  String \_userAnswer = sc.next();  // 判斷  if (\_userAnswer.toUpperCase().equals(q.getAnswer())) {  System.out.println("Good");  } else {  System.out.println("Stupid");  }  }  sc.close();  System.out.println("答题结束");  }  } |

# 记录错题

|  |
| --- |
| package org.ah.core;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import java.util.Random;  import java.util.Scanner;  import org.ah.core.bean.Question;  import org.ah.dao.QuestionDao;  public class Bus {  public static List<Question> extractQuestions(List<Question> sourceQs,  int count) {  Random r = new Random();  int n随机数 = -1;  List<Question> newQs = new ArrayList<Question>();  for (int i = 1; i <= count; i++) {  n随机数 = r.nextInt(sourceQs.size());  Question q选中的题 = sourceQs.get(n随机数);  newQs.add(q选中的题);  sourceQs.remove(n随机数);// 防止重复选中同一道题  }  return newQs;  }  **private static List<Question> errQs = new ArrayList<Question>();**  **private static List<Question> newQs = new ArrayList<Question>();// 提取出来**  /\*\*  \* 记录错题  \*  \* @param q  \*/  **public static void recordErr(Question q) {**  **errQs.add(q);**  **newQs.remove(q);**  **}**  /\*\*  \* 统计分数  \*/  **public static void calcScore() {**  **int score = 100 \* newQs.size() / (newQs.size() + errQs.size());**  **System.out.println("得分：" + score);**  **}**  public static void main(String[] args) throws IOException {  Scanner sc = new Scanner(System.in);  QuestionDao dao = new QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  newQs = extractQuestions(qs全部題目, 5);  int newId = 1;  **for (int i = 0; i < newQs.size(); i++) {**  **Question q = newQs.get(i);**  **// 列表在变，此语句无效 for (Question q : newQs) {**  System.out.println((newId++) + "." + q.getTitle());  System.out.println("\tA." + q.getOptA());  System.out.println("\tB." + q.getOptB());  System.out.println("\tC." + q.getOptC());  System.out.println("\tD." + q.getOptD());  String \_userAnswer = sc.next();  // 判斷  if (\_userAnswer.toUpperCase().equals(q.getAnswer())) {  } else {  **recordErr(q);**  **i--;**  }  }  sc.close();  System.out.println("答题结束");  calcScore();  }  } |

# 显示错题

需记录题号，记录用户输入的答案

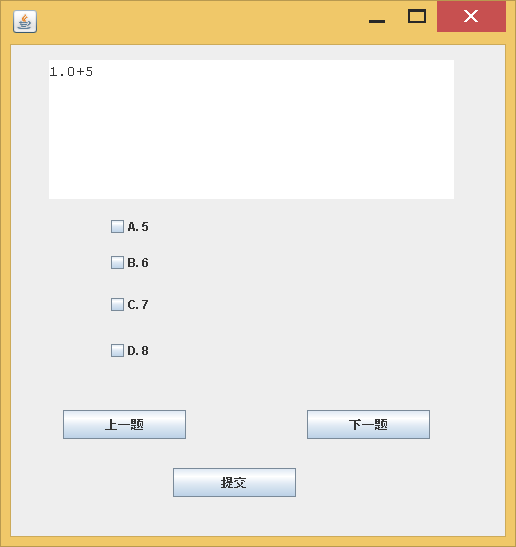
|  |
| --- |
| Question对象需要做修改，追加一个属性 |
| String userAnswer;  **public** String getUserAnswer() {  **return** userAnswer;  }  **public** **void** setUserAnswer(String userAnswer) {  **this**.userAnswer = userAnswer;  } |

|  |
| --- |
| package org.ah.core;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import java.util.Random;  import java.util.Scanner;  import org.ah.core.bean.Question;  import org.ah.dao.QuestionDao;  public class Bus {  public static List<Question> extractQuestions(List<Question> sourceQs,  int count) {  Random r = new Random();  int n随机数 = -1;  List<Question> newQs = new ArrayList<Question>();  for (int i = 1; i <= count; i++) {  n随机数 = r.nextInt(sourceQs.size());  Question q选中的题 = sourceQs.get(n随机数);  newQs.add(q选中的题);  sourceQs.remove(n随机数);// 防止重复选中同一道题  }  return newQs;  }  private static List<Question> errQs = new ArrayList<Question>();  private static List<Question> newQs = new ArrayList<Question>();// 提取出来  /\*\*  \* 记录错题  \*  \* @param q  \*/  public static void recordErr(Question q) {  errQs.add(q);  newQs.remove(q);  }  /\*\*  \* 统计分数  \*/  public static void calcScore() {  int score = 100 \* newQs.size() / (newQs.size() + errQs.size());  System.out.println("得分：" + score);  }  **public static void showErr() {**  **for (Question q : errQs) {**  **System.out.println(q.getId() + "." + q.getTitle());**  **System.out.println("\tA." + q.getOptA());**  **System.out.println("\tB." + q.getOptB());**  **System.out.println("\tC." + q.getOptC());**  **System.out.println("\tD." + q.getOptD());**  **System.out.println("正确答案：" + q.getAnswer());**  **System.out.println("您的答案：" + q.getUserAnswer());**  **}**  **}**  public static void main(String[] args) throws IOException {  Scanner sc = new Scanner(System.in);  QuestionDao dao = new QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  newQs = extractQuestions(qs全部題目, 5);  int newId = 1;  for (int i = 0; i < newQs.size(); i++) {  Question q = newQs.get(i);  // 列表在变，此语句无效 for (Question q : newQs) {  q.setId(newId++);  System.out.println(q.getId() + "." + q.getTitle());  System.out.println("\tA." + q.getOptA());  System.out.println("\tB." + q.getOptB());  System.out.println("\tC." + q.getOptC());  System.out.println("\tD." + q.getOptD());  String \_userAnswer = sc.next();  // 判斷  if (\_userAnswer.toUpperCase().equals(q.getAnswer())) {  } else {  **q.setUserAnswer(\_userAnswer.toUpperCase());**  recordErr(q);  i--;  }  }  System.out.println("答题结束");  calcScore();  **System.out.println("输入任意键查看错题");**  **System.in.read();**  **sc.close();**  showErr();  }  } |

# 引入窗体

## 显示

|  |
| --- |
| F3QA.java |
| **package** org.ah.form;  **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **import** javax.swing.border.EmptyBorder;  **import** javax.swing.JCheckBox;  **import** javax.swing.JTextArea;  **import** java.awt.Font;  **import** java.util.List;  **import** javax.swing.JButton;  **import** org.ah.core.Bus;  **import** org.ah.core.bean.Question;  **import** org.ah.dao.QuestionDao;  **public** **class** F3QA **extends** JFrame {  **private** JPanel contentPane;  **private** JCheckBox chckbxA;  **private** JCheckBox chckbxB;  **private** JCheckBox chckbxC;  **private** JCheckBox chckbxD;  **private** JTextArea textArea;  /\*\*  \* Launch the application.  \*/  **public** **static** **void** main(String[] args) {  F3QA frame = **new** F3QA();  frame.setVisible(**true**);  }  **private void intiForm() {**  **QuestionDao dao = new QuestionDao();**  **List<Question> qs全部題目 = dao.getAllQuestions();**  **// 随机取题**  **Bus.*newQs* = Bus.*extractQuestions*(qs全部題目, 5);**  **int newId = 1;**  **for (int i = 0; i < Bus.*newQs*.size(); i++) {**  **Question q = Bus.*newQs*.get(i);**  **// 列表在变，此语句无效 for (Question q : newQs) {**  **q.setId(newId++);**  **this.textArea.setText(q.getId() + "." + q.getTitle());**  **this.chckbxA.setText("A. " + q.getOptA());**  **this.chckbxB.setText("B. " + q.getOptB());**  **this.chckbxC.setText("C. " + q.getOptC());**  **this.chckbxD.setText("D. " + q.getOptD());**  **}**  **}**  /\*\*  \* Create the frame.  \*/  **public** F3QA() {  setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  setBounds(100, 100, 516, 547);  contentPane = **new** JPanel();  contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(**null**);  chckbxA = **new** JCheckBox("A");  chckbxA.setBounds(96, 167, 149, 29);  contentPane.add(chckbxA);  chckbxB = **new** JCheckBox("B");  chckbxB.setBounds(96, 203, 149, 29);  contentPane.add(chckbxB);  chckbxC = **new** JCheckBox("C");  chckbxC.setBounds(96, 245, 149, 29);  contentPane.add(chckbxC);  chckbxD = **new** JCheckBox("D");  chckbxD.setBounds(96, 291, 149, 29);  contentPane.add(chckbxD);  textArea = **new** JTextArea();  textArea.setEditable(**false**);  textArea.setFont(**new** Font("Monospaced", Font.***PLAIN***, 15));  textArea.setLineWrap(**true**);  textArea.setText("");  textArea.setBounds(38, 15, 405, 139);  contentPane.add(textArea);  JButton btnSubmit = **new** JButton("提交");  btnSubmit.setBounds(162, 423, 123, 29);  contentPane.add(btnSubmit);  JButton btnPrevious = **new** JButton("上一题");  btnPrevious.setBounds(52, 365, 123, 29);  contentPane.add(btnPrevious);  JButton button = **new** JButton("下一题");  button.setBounds(296, 365, 123, 29);  contentPane.add(button);  **intiForm();**  }  } |

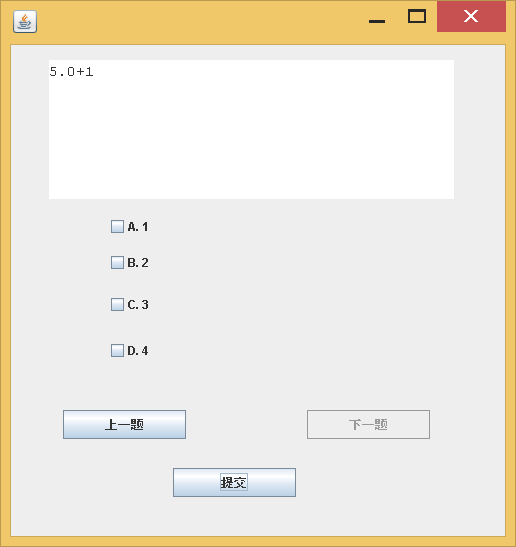


## 下一题

|  |
| --- |
| **package** org.ah.form;  **import** javax.swing.JFrame;  **public** **class** F3QA **extends** JFrame {  **private** JPanel contentPane;  **private** JCheckBox chckbxA;  **private** JCheckBox chckbxB;  **private** JCheckBox chckbxC;  **private** JCheckBox chckbxD;  **private** JTextArea textArea;  /\*\*  \* Launch the application.  \*/  **public** **static** **void** main(String[] args) {  F3QA frame = **new** F3QA();  frame.setVisible(**true**);  }  **private** **void** ShowQuestion(Question q) {  **this**.textArea.setText(q.getId() + "." + q.getTitle());  **this**.chckbxA.setText("A. " + q.getOptA());  **this**.chckbxB.setText("B. " + q.getOptB());  **this**.chckbxC.setText("C. " + q.getOptC());  **this**.chckbxD.setText("D. " + q.getOptD());  }  **int** qId = 0;  **private** **void** intiForm() {  QuestionDao dao = **new** QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  Bus.*newQs* = Bus.*extractQuestions*(qs全部題目, 5);  **for** (**int** i = 0; i < Bus.*newQs*.size(); i++) {  Question q = Bus.*newQs*.get(i);  // 列表在变，此语句无效 for (Question q : newQs) {  q.setId(i+1);  }  ShowQuestion(Bus.*newQs*.get(qId));  }  /\*\*  \* Create the frame.  \*/  **public** F3QA() {  setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  setBounds(100, 100, 516, 547);  contentPane = **new** JPanel();  contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(**null**);  chckbxA = **new** JCheckBox("A");  chckbxA.setBounds(96, 167, 149, 29);  contentPane.add(chckbxA);  chckbxB = **new** JCheckBox("B");  chckbxB.setBounds(96, 203, 149, 29);  contentPane.add(chckbxB);  chckbxC = **new** JCheckBox("C");  chckbxC.setBounds(96, 245, 149, 29);  contentPane.add(chckbxC);  chckbxD = **new** JCheckBox("D");  chckbxD.setBounds(96, 291, 149, 29);  contentPane.add(chckbxD);  textArea = **new** JTextArea();  textArea.setEditable(**false**);  textArea.setFont(**new** Font("Monospaced", Font.***PLAIN***, 15));  textArea.setLineWrap(**true**);  textArea.setText("");  textArea.setBounds(38, 15, 405, 139);  contentPane.add(textArea);  JButton btnSubmit = **new** JButton("提交");  btnSubmit.setBounds(162, 423, 123, 29);  contentPane.add(btnSubmit);  JButton btnPrevious = **new** JButton("上一题");  btnPrevious.setBounds(52, 365, 123, 29);  contentPane.add(btnPrevious);  JButton button = **new** JButton("下一题");  button.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  ShowQuestion(Bus.*newQs*.get(++qId));  }  });  button.setBounds(296, 365, 123, 29);  contentPane.add(button);  intiForm();  }  } |

## 下一题限制

|  |
| --- |
| **package** org.ah.form;  **import** javax.swing.JFrame;  **public** **class** F3QA **extends** JFrame {  **private** JPanel contentPane;  **private** JCheckBox chckbxA;  **private** JCheckBox chckbxB;  **private** JCheckBox chckbxC;  **private** JCheckBox chckbxD;  **private** JTextArea textArea;  /\*\*  \* Launch the application.  \*/  **public** **static** **void** main(String[] args) {  F3QA frame = **new** F3QA();  frame.setVisible(**true**);  }  **private** **void** ShowQuestion(Question q) {  **this**.textArea.setText(q.getId() + "." + q.getTitle());  **this**.chckbxA.setText("A. " + q.getOptA());  **this**.chckbxB.setText("B. " + q.getOptB());  **this**.chckbxC.setText("C. " + q.getOptC());  **this**.chckbxD.setText("D. " + q.getOptD());  **if** (q.getId() >= 5) {  **this**.btnNext.setEnabled(**false**);  } **else** {  **this**.btnNext.setEnabled(**true**);  }  }  **int** qId = 0;  **private** JButton btnSubmit;  **private** JButton btnPrevious;  **private** JButton btnNext;  **private** **void** intiForm() {  QuestionDao dao = **new** QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  Bus.*newQs* = Bus.*extractQuestions*(qs全部題目, 5);    **for** (**int** i = 0; i < Bus.*newQs*.size(); i++) {  Question q = Bus.*newQs*.get(i);  // 列表在变，此语句无效 for (Question q : newQs) {  q.setId(i+1);  }  ShowQuestion(Bus.*newQs*.get(qId));  }  /\*\*  \* Create the frame.  \*/  **public** F3QA() {  setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  setBounds(100, 100, 516, 547);  contentPane = **new** JPanel();  contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(**null**);  chckbxA = **new** JCheckBox("A");  chckbxA.setBounds(96, 167, 149, 29);  contentPane.add(chckbxA);  chckbxB = **new** JCheckBox("B");  chckbxB.setBounds(96, 203, 149, 29);  contentPane.add(chckbxB);  chckbxC = **new** JCheckBox("C");  chckbxC.setBounds(96, 245, 149, 29);  contentPane.add(chckbxC);  chckbxD = **new** JCheckBox("D");  chckbxD.setBounds(96, 291, 149, 29);  contentPane.add(chckbxD);  textArea = **new** JTextArea();  textArea.setEditable(**false**);  textArea.setFont(**new** Font("Monospaced", Font.***PLAIN***, 15));  textArea.setLineWrap(**true**);  textArea.setText("《西游记》是中国古代第一部浪漫主义章回体长篇神魔小说。该书以“唐僧取经”这一历史事件为蓝本，通过作者的艺术加工，深刻地描绘了当时的社会现实。主要描写了孙悟空出世，后遇见了唐僧、猪八戒和沙和尚三人，一路降妖伏魔，保护唐僧西行取经，经历了九九八十一难，终于到达西天见到如来佛祖，最终五圣成真的故事。");  textArea.setBounds(38, 15, 405, 139);  contentPane.add(textArea);  btnSubmit = **new** JButton("提交");  btnSubmit.setBounds(162, 423, 123, 29);  contentPane.add(btnSubmit);  btnPrevious = **new** JButton("上一题");  btnPrevious.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  }  });  btnPrevious.setBounds(52, 365, 123, 29);  contentPane.add(btnPrevious);  btnNext = **new** JButton("下一题");  btnNext.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {    ShowQuestion(Bus.*newQs*.get(++qId));  }  });  btnNext.setBounds(296, 365, 123, 29);  contentPane.add(btnNext);  intiForm();  }  } |



## 提交

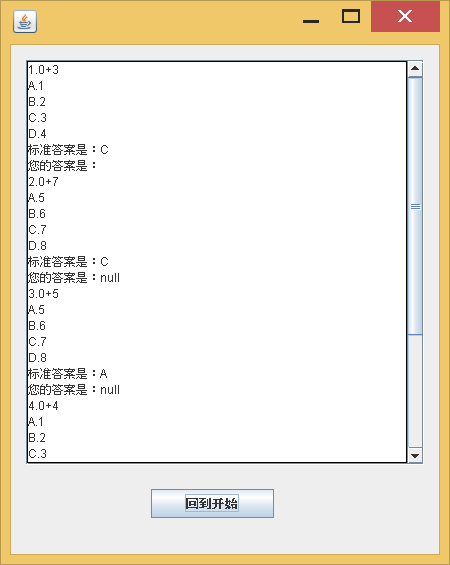
需要记录答案。请问：哪些事件触发时，需要记录答案？

|  |
| --- |
| package org.ah.form;  import javax.swing.JFrame;  import javax.swing.JPanel;  import javax.swing.border.EmptyBorder;  import javax.swing.JCheckBox;  import javax.swing.JTextArea;  import java.awt.Font;  import java.util.List;  import javax.swing.JButton;  import org.ah.core.Bus;  import org.ah.core.bean.Question;  import org.ah.dao.QuestionDao;  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  public class F3QA extends JFrame {  private JPanel contentPane;  private JCheckBox chckbxA;  private JCheckBox chckbxB;  private JCheckBox chckbxC;  private JCheckBox chckbxD;  private JTextArea textArea;  /\*\*  \* Launch the application.  \*/  public static void main(String[] args) {  F3QA frame = new F3QA();  frame.setVisible(true);  }  private void ShowQuestion(Question q) {  this.textArea.setText(q.getId() + "." + q.getTitle());  this.chckbxA.setText("A. " + q.getOptA());  this.chckbxB.setText("B. " + q.getOptB());  this.chckbxC.setText("C. " + q.getOptC());  this.chckbxD.setText("D. " + q.getOptD());  if (q.getId() >= 5) {  this.btnNext.setEnabled(false);  } else {  this.btnNext.setEnabled(true);  }  if (q.getId() <= 1) {  this.btnPrevious.setEnabled(false);  } else {  this.btnPrevious.setEnabled(true);  }  }  int qId = 0;  private JButton btnSubmit;  private JButton btnPrevious;  private JButton btnNext;  private void intiForm() {  QuestionDao dao = new QuestionDao();  List<Question> qs全部題目 = dao.getAllQuestions();  // 随机取题  Bus.newQs = Bus.extractQuestions(qs全部題目, 5);  for (int i = 0; i < Bus.newQs.size(); i++) {  Question q = Bus.newQs.get(i);  // 列表在变，此语句无效 for (Question q : newQs) {  q.setId(i + 1);  }  ShowQuestion(Bus.newQs.get(qId));  }  private void recordAnswer(int i) {  Question q = Bus.newQs.get(i);  String sUserAnswer = "";  if (this.chckbxA.isSelected()) {  sUserAnswer += "A";  }  if (this.chckbxB.isSelected()) {  sUserAnswer += "B";  }  if (this.chckbxC.isSelected()) {  sUserAnswer += "C";  }  if (this.chckbxD.isSelected()) {  sUserAnswer += "D";  }  q.setUserAnswer(sUserAnswer);  }  private void submit() {  for (int i = 0; i < Bus.newQs.size(); i++) {  Question q = Bus.newQs.get(i);  if (q.getAnswer().equals(q.getUserAnswer())) {  } else {  Bus.recordErr(q);  i--;  }  }  int score = 100 \* Bus.newQs.size() / (Bus.newQs.size() + Bus.errQs.size());  JOptionPane.showMessageDialog(null,  score + "", "分数", JOptionPane.INFORMATION\_MESSAGE);  Bus.*showErr*();  }  /\*\*  \* Create the frame.  \*/  public F3QA() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 516, 547);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  chckbxA = new JCheckBox("A");  chckbxA.setBounds(96, 167, 149, 29);  contentPane.add(chckbxA);  chckbxB = new JCheckBox("B");  chckbxB.setBounds(96, 203, 149, 29);  contentPane.add(chckbxB);  chckbxC = new JCheckBox("C");  chckbxC.setBounds(96, 245, 149, 29);  contentPane.add(chckbxC);  chckbxD = new JCheckBox("D");  chckbxD.setBounds(96, 291, 149, 29);  contentPane.add(chckbxD);  textArea = new JTextArea();  textArea.setEditable(false);  textArea.setFont(new Font("Monospaced", Font.PLAIN, 15));  textArea.setLineWrap(true);  textArea.setText("");  textArea.setBounds(38, 15, 405, 139);  contentPane.add(textArea);  btnSubmit = new JButton("提交");  btnSubmit.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  recordAnswer(qId);  **submit();**  }  });  btnSubmit.setBounds(162, 423, 123, 29);  contentPane.add(btnSubmit);  btnPrevious = new JButton("上一题");  btnPrevious.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  }  });  btnPrevious.setBounds(52, 365, 123, 29);  contentPane.add(btnPrevious);  btnNext = new JButton("下一题");  btnNext.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  **recordAnswer(qId);**  ShowQuestion(Bus.newQs.get(++qId));  }  });  btnNext.setBounds(296, 365, 123, 29);  contentPane.add(btnNext);  intiForm();  }  } |

### 清空上一题选项

|  |
| --- |
| **private** **void** clearOption(Question q) {  **this**.chckbxA.setSelected(**false**);  **this**.chckbxB.setSelected(**false**);  **this**.chckbxC.setSelected(**false**);  **this**.chckbxD.setSelected(**false**);  String sUserAnswer = q.getUserAnswer();  **if** (sUserAnswer != **null**) {  **if** (sUserAnswer.contains("A")) {  **this**.chckbxA.setSelected(**true**);  }  **if** (sUserAnswer.contains("B")) {  **this**.chckbxB.setSelected(**true**);  }  **if** (sUserAnswer.contains("C")) {  **this**.chckbxC.setSelected(**true**);  }  **if** (sUserAnswer.contains("D")) {  **this**.chckbxD.setSelected(**true**);  }  }  }  **private** **void** ShowQuestion(Question q) {  **this**.textArea.setText(q.getId() + "." + q.getTitle());  **this**.chckbxA.setText("A. " + q.getOptA());  **this**.chckbxB.setText("B. " + q.getOptB());  **this**.chckbxC.setText("C. " + q.getOptC());  **this**.chckbxD.setText("D. " + q.getOptD());  **this**.btnNext.setEnabled(**false**);  } **else** {  **this**.btnNext.setEnabled(**true**);  }  **if** (q.getId() <= 1) {  **this**.btnPrevious.setEnabled(**false**);  } **else** {  **this**.btnPrevious.setEnabled(**true**);  }  clearOption(q);  } |

## 多窗口：显示错题



|  |
| --- |
| **F3QA.java** |
| **private** **void** submit() {  **for** (**int** i = 0; i < Bus.*newQs*.size(); i++) {  Question q = Bus.*newQs*.get(i);  **if** (q.getAnswer().equals(q.getUserAnswer())) {  } **else** {  Bus.*recordErr*(q);  i--;  }  }  // 计分  **int** score = 100 \* Bus.*newQs*.size()  / (Bus.*newQs*.size() + Bus.*errQs*.size());  JOptionPane.*showMessageDialog*(**null**, score + "", "分数",  JOptionPane.***INFORMATION\_MESSAGE***);  Bus.*showErr*();  // 显示错题  F4ShowErr f4 = **new** F4ShowErr();  f4.setVisible(**true**);  **this**.setEnabled(**false**);  } |
| **F4ShowErr.java** |
| package org.ah.form;  import java.awt.BorderLayout;  import java.awt.EventQueue;  import javax.swing.JFrame;  import javax.swing.JPanel;  import javax.swing.JScrollBar;  import javax.swing.border.EmptyBorder;  import javax.swing.JTextArea;  import javax.swing.JScrollPane;  import javax.swing.border.LineBorder;  import java.awt.Color;  import javax.swing.JButton;  import org.ah.core.Bus;  import org.ah.core.bean.Question;  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  public class F4ShowErr extends JFrame {  private JPanel contentPane;  private JScrollBar scrollBar;  private JScrollPane scrollPane;  /\*\*  \* Launch the application.  \*/  public static void main(String[] args) {  F4ShowErr frame = new F4ShowErr();  frame.setVisible(true);  }  /\*\*  \* Create the frame.  \*/  public F4ShowErr() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 565);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  JTextArea txtaShow = new JTextArea();  txtaShow.setEditable(false);  txtaShow.setLineWrap(true);  txtaShow.setText("");  txtaShow.setBounds(15, 15, 343, 405);  contentPane.add(txtaShow);  scrollPane = new JScrollPane(txtaShow);  scrollPane.setViewportBorder(new LineBorder(new Color(0, 0, 0)));  scrollPane.setBounds(15, 15, 398, 405);  contentPane.add(scrollPane);  **JButton btnNewButton = new JButton("回到开始");**  **btnNewButton.addActionListener(new ActionListener() {**  **public void actionPerformed(ActionEvent e) {**  **scrollBar = scrollPane.getVerticalScrollBar();**  **scrollBar.setValue(1);**  **}**  **});**  **btnNewButton.setBounds(140, 444, 123, 29);**  **contentPane.add(btnNewButton);**  // 显示内容  if (Bus.errQs.size() > 0) {  StringBuffer sb = new StringBuffer();  for (Question q : Bus.errQs) {  sb.append(q.getId() + "." + q.getTitle() + "\n");  sb.append("A." + q.getOptA() + "\n");  sb.append("B." + q.getOptB() + "\n");  sb.append("C." + q.getOptC() + "\n");  sb.append("D." + q.getOptD() + "\n");  sb.append("标准答案是：" + q.getAnswer() + "\n");  sb.append("您的答案是：" + q.getUserAnswer()+ "\n");  }  txtaShow.setText(sb.toString());  } else {  txtaShow.setText("学霸V5");  }  }  } |

## 窗口回

|  |
| --- |
| F4ShowErr |
| **public** F4ShowErr(**final** F3QA f3) {  **this**();  addWindowListener(**new** WindowAdapter() {  @Override  **public** **void** windowClosed(WindowEvent e) {  f3.setEnabled(**true**);  f3.setVisible(**true**);  }  });  }  **public** F4ShowErr() {  setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);  ……  } |
| F3QA |
| **private** **void** submit() {  **for** (**int** i = 0; i < Bus.*newQs*.size(); i++) {  Question q = Bus.*newQs*.get(i);  **if** (q.getAnswer().equals(q.getUserAnswer())) {  } **else** {  Bus.*recordErr*(q);  i--;  }  }  // 计分  **int** score = 100 \* Bus.*newQs*.size()  / (Bus.*newQs*.size() + Bus.*errQs*.size());  JOptionPane.*showMessageDialog*(**null**, score + "", "分数",  JOptionPane.***INFORMATION\_MESSAGE***);  Bus.*showErr*();  // 显示错题  F4ShowErr f4 = **new** F4ShowErr(**this**);  f4.setVisible(**true**);  **this**.setEnabled(**false**);  } |

# End