МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

Лабораторная работа №11

по дисциплине: «Современные технологии разработки программного обеспечения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Чибисов И.А.* | Преподаватель | *Эстрайх И.В.*  *ст. преподаватель* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.04.03 – Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АПМ-22* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр | *010594214* |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г | |

Новосибирск

2023

Цель работы: научиться подключать и использовать в приложениях базы данных

Задание на лабораторную работу

Задание на лабораторную работу

1. Добавьте в БД ещё несколько таблиц с разной структурой, а на главный экран приложения — соответствующие им кнопки. Доработайте приложение, чтобы при нажатии на разные кнопки открывались соответствующие им таблицы.

2. Подключите к приложению-викторине, созданному в предыдущей лабораторной работе базу данных, храните в ней вопросы и ответы, а уже из неё отображайте в приложение. Примерная структура БД в этом случае: таблица вопросов (идентификатор вопроса, текст вопроса) и таблица ответов (идентификатор ответа, текст ответа идентификатор вопроса, идентификатор правильного ответа).

3. Включите в отчет цель работы, коды программ, скриншоты с результатами и вывод.

Приложение 1 Показ таблиц

Для создания приложения первым делом я подготовил БД

Я создал три простые таблицы для работы с ними в Студии, для создания использовал DataGrip.

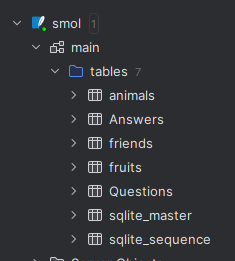


Рисунок 1 – БД smol

Нас интересуют таблицы animals, friends и fruits, струкутра таблиц представлена ниже:

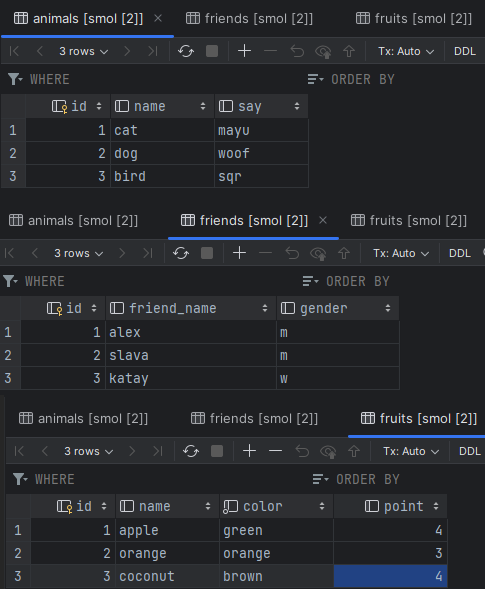


Рисунок 2 – созданные таблицы

Приступим к приложению, на главном экране у нас 3 кнопки они переводят на конкретную таблицу.

Код MainActivity ниже:

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** Button btnAnimals = findViewById(R.id.*btnAnimals*)**;** Button btnFriends = findViewById(R.id.*btnFriends*)**;** Button btnFruits = findViewById(R.id.*btnFruits*)**;** btnAnimals.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openTableActivity("animals")**;** }  
 })**;** btnFriends.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openTableActivity("friends")**;** }  
 })**;** btnFruits.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openTableActivity("fruits")**;** }  
 })**;** }  
  
 private void openTableActivity(String tableName) {  
 Intent intent = new Intent(this**,** TableActivity.class)**;** intent.putExtra("tableName"**,** tableName)**;** startActivity(intent)**;** }  
}

В нем мы считываем клик по кнопки и на новой активности открываем выбранную таблицу.

В классе TableActivity мы подключаемя к БД и делаем вывод нужных данных, код ниже:

public class TableActivity extends AppCompatActivity {  
  
 private ListView listView**;** private SQLiteDatabase database**;** @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_table*)**;** listView = findViewById(R.id.*listView*)**;** String tableName = getIntent().getStringExtra("tableName")**;** database = openDatabase()**;** displayTableContents(tableName)**;** }  
  
 private SQLiteDatabase openDatabase() {  
 try {  
 File dbFile = getDatabasePath("smol.db")**;** if (!dbFile.exists()) {  
 InputStream inputStream = getAssets().open("smol.db")**;** OutputStream outputStream = new FileOutputStream(dbFile)**;** byte[] buffer = new byte[1024]**;** int length**;** while ((length = inputStream.read(buffer)) > 0) {  
 outputStream.write(buffer**,** 0**,** length)**;** }  
  
 outputStream.flush()**;** outputStream.close()**;** inputStream.close()**;** }  
  
 return SQLiteDatabase.*openDatabase*(dbFile.getPath()**,** null**,** SQLiteDatabase.*OPEN\_READWRITE*)**;** } catch (IOException e) {  
 e.printStackTrace()**;** }  
  
 return null**;** }  
  
  
 private void displayTableContents(String tableName) {  
 List<String> items = new ArrayList<>()**;** Cursor cursor = database.rawQuery("SELECT \* FROM " + tableName**,** null)**;** int columnsCount = cursor.getColumnCount()**;** // Добавляем названия колонок к списку items  
 StringBuilder columnNamesBuilder = new StringBuilder()**;** for (int i = 0**;** i < columnsCount**;** i++) {  
 columnNamesBuilder.append(cursor.getColumnName(i)).append(" ")**;** }  
 items.add(columnNamesBuilder.toString())**;** while (cursor.moveToNext()) {  
 StringBuilder rowBuilder = new StringBuilder()**;** for (int i = 0**;** i < columnsCount**;** i++) {  
 rowBuilder.append(cursor.getString(i)).append(" ")**;** }  
 items.add(rowBuilder.toString())**;** }  
  
 cursor.close()**;** database.close()**;** ArrayAdapter<String> adapter = new ArrayAdapter<>(this**,** android.R.layout.*simple\_list\_item\_1***,** items)**;** listView.setAdapter(adapter)**;** }  
  
}

Работа представлена на рисунках 3-4 ниже:

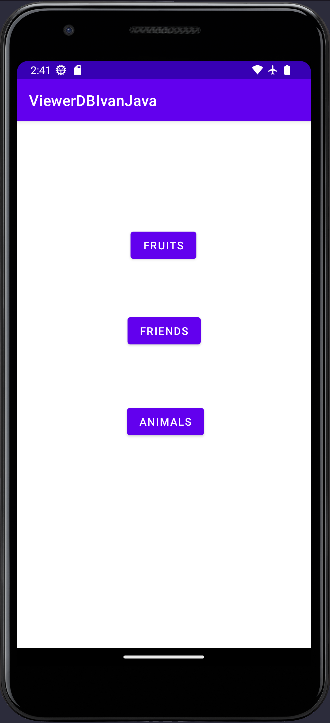


Рисунок 3– главный экран

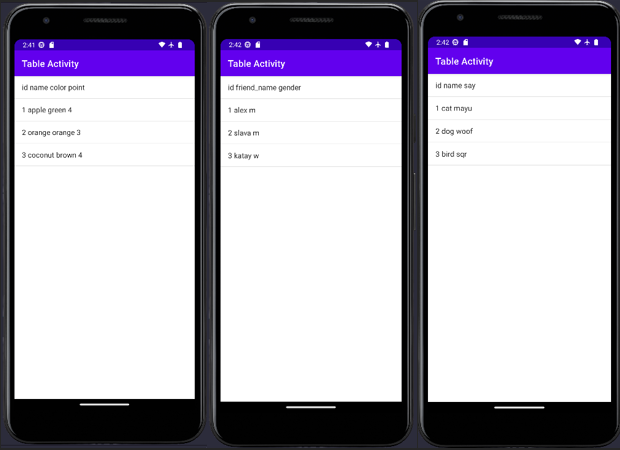


Рисунок 4 – активность для таблиц

Как мы видим активность адаптирована под разное кол-во колонок в таблице, так же может вывести больше строк, в коде это учтено.

Приложение 2 Викторина с БД

Для создания приложения первым делом я подготовил БД, точнее доработал прошлую, рисунок 5.

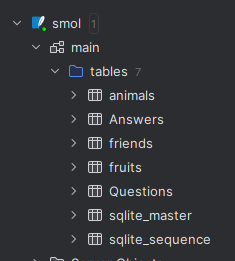


Рисунок 5 – БД smol

Я создал две таблицы для ответов и вопросов, рисунок 6-7.

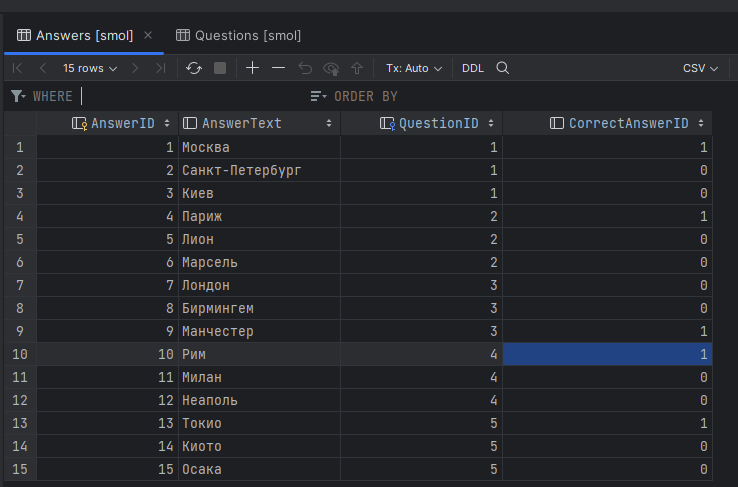


Рисунок 6 – Ответы

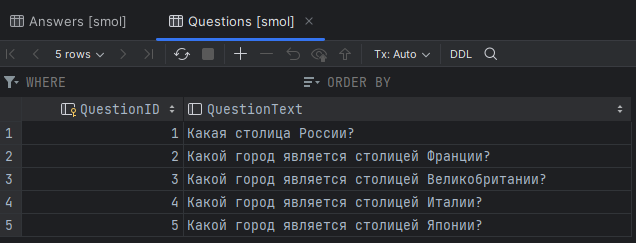


Рисунок 7 – Вопросы

Создавал и заполонял скрипом ниже:

-- Создание таблицы вопросов  
CREATE TABLE Questions (  
 QuestionID INTEGER PRIMARY KEY,  
 QuestionText TEXT  
);  
  
-- Создание таблицы ответов  
CREATE TABLE Answers (  
 AnswerID INTEGER PRIMARY KEY,  
 AnswerText TEXT,  
 QuestionID INTEGER,  
 CorrectAnswerID INTEGER,  
 FOREIGN KEY (QuestionID) REFERENCES Questions(QuestionID)  
);  
  
-- Вставка вопросов  
INSERT INTO Questions (QuestionID, QuestionText) VALUES  
 (1, 'Какая столица России?'),  
 (2, 'Какой город является столицей Франции?'),  
 (3, 'Какой город является столицей Великобритании?'),  
 (4, 'Какой город является столицей Италии?'),  
 (5, 'Какой город является столицей Японии?');  
  
-- Вставка ответов  
INSERT INTO Answers (AnswerID, AnswerText, QuestionID, CorrectAnswerID) VALUES  
 (1, 'Москва', 1, 1),  
 (2, 'Санкт-Петербург', 1, 0),  
 (3, 'Киев', 1, 0),  
 (4, 'Париж', 2, 1),  
 (5, 'Лион', 2, 0),  
 (6, 'Марсель', 2, 0),  
 (7, 'Лондон', 3, 0),  
 (8, 'Бирмингем', 3, 0),  
 (9, 'Манчестер', 3, 1),  
 (10, 'Рим', 4, 1),  
 (11, 'Милан', 4, 0),  
 (12, 'Неаполь', 4, 0),  
 (13, 'Токио', 5, 1),  
 (14, 'Киото', 5, 0),  
 (15, 'Осака', 5, 0);

Перейдем к приложению, главный экран — это выпадающий список из 5 стран и кнопкой, для перехода на викторину.

Активность с викториной, это вопрос из БД и три ответа, так же из БД.

Код MainActivity

public class MainActivity extends AppCompatActivity {  
  
 private Spinner countrySpinner**;** @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** countrySpinner = findViewById(R.id.*countrySpinner*)**;** Button startQuizButton = findViewById(R.id.*startQuizButton*)**;** startQuizButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String selectedCountry = countrySpinner.getSelectedItem().toString()**;** Intent intent = new Intent(MainActivity.this**,** QuizActivity.class)**;** intent.putExtra("country"**,** selectedCountry)**;** startActivity(intent)**;** }  
 })**;** }  
}

С него мы просто вызываем викторину для выбранной страны.

Код QuizActivity:

Данный класс уже сложнее и больше, поэтому представлен в приложение А.

Данный класс представляет собой сразу несколько функций, эта работа с БД и вызов пары дополнительных методов для работы с визуалом, покраской кнопок и вызовом тостов.

Перейдем к обзору приложения.

На рисунке 8 представлен главный экран:

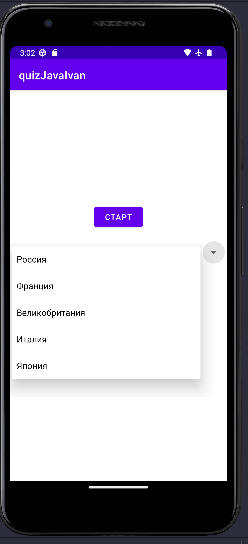
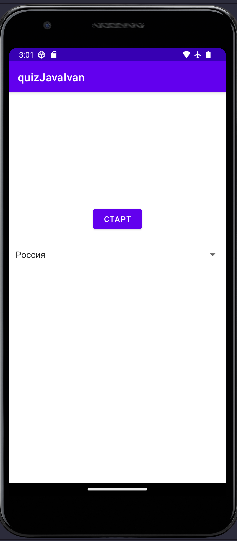


Рисунок 8 – Главный экран

На рисунке 9-10 представлена Quiz активность, с выбором правильных и неправильных ответов.

Как мы видим она отлично работает.

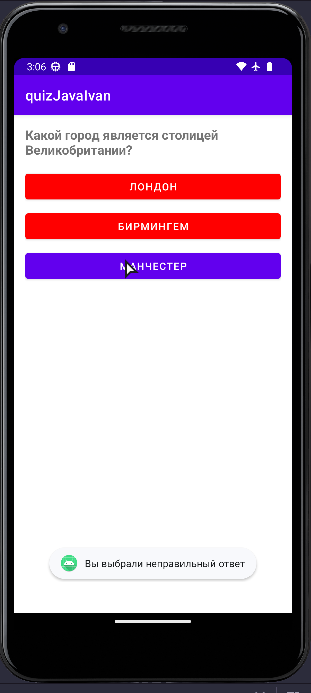
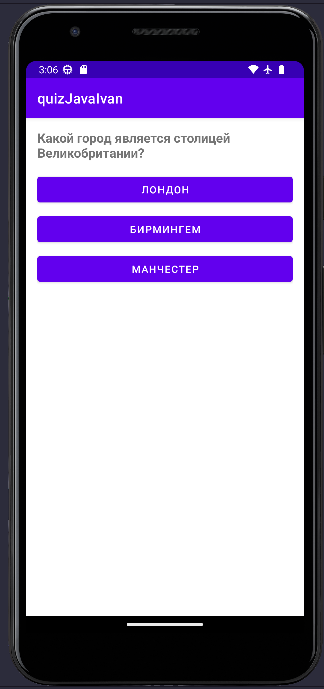


Рисунок 9 – Пример викторины

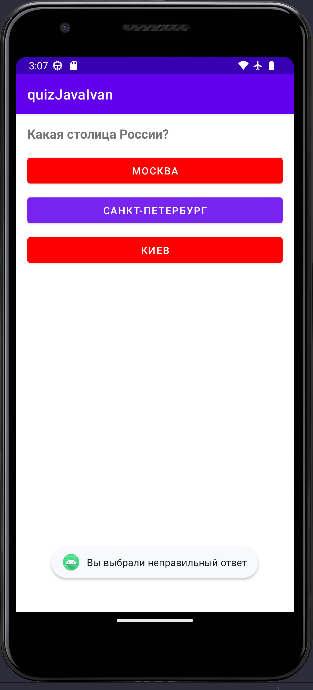
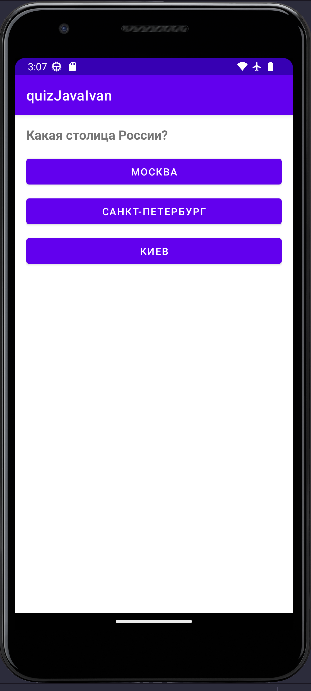
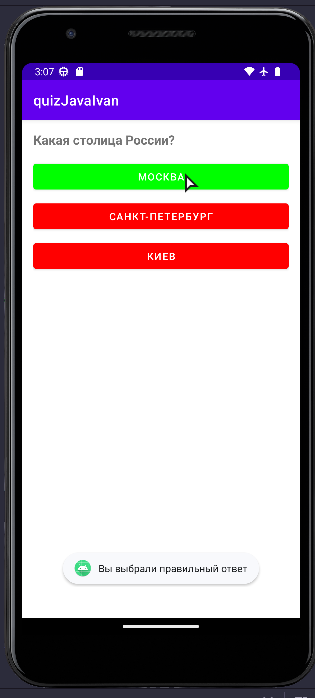
 

Рисунок 10– Пример викторины

Вывод: мы научились подключать и использовать в приложениях базы данных

ПРИЛОЖЕНИЕ А

public class QuizActivity extends AppCompatActivity {  
  
 private Button answer1Button**;** private Button answer2Button**;** private Button answer3Button**;** private Button answer4Button**;** private String selectedCountry**;** private SQLiteDatabase database**;** @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_quiz*)**;** // Инициализация базы данных  
 copyDatabaseFromAssets()**;** database = openOrCreateDatabase("smol.db"**,** *MODE\_PRIVATE***,** null)**;** TextView questionTextView = findViewById(R.id.*questionTextView*)**;** answer1Button = findViewById(R.id.*answer1Button*)**;** answer2Button = findViewById(R.id.*answer2Button*)**;** answer3Button = findViewById(R.id.*answer3Button*)**;** selectedCountry = getIntent().getStringExtra("country")**;** // Инициализация базы данных  
 database = openOrCreateDatabase("smol.db"**,** *MODE\_PRIVATE***,** null)**;** // Получение вопроса и ответов из базы данных  
 String question = getQuestionFromDB()**;** String[] answers = getAnswersFromDB()**;** // Установка вопроса и ответов на экране  
 questionTextView.setText(question)**;** answer1Button.setText(answers[0])**;** answer2Button.setText(answers[1])**;** answer3Button.setText(answers[2])**;** // Назначение слушателей для кнопок  
 answer1Button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 checkAnswer(answer1Button.getText().toString())**;** }  
 })**;** answer2Button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 checkAnswer(answer2Button.getText().toString())**;** }  
 })**;** answer3Button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 checkAnswer(answer3Button.getText().toString())**;** }  
 })**;** }  
  
 private String getQuestionFromDB() {  
 String question = ""**;** String query = "SELECT QuestionText FROM Questions WHERE QuestionID = " + getQuestionID()**;** Cursor cursor = database.rawQuery(query**,** null)**;** if (cursor.moveToFirst()) {  
 question = cursor.getString(0)**;** }  
 cursor.close()**;** return question**;** }  
  
 private String[] getAnswersFromDB() {  
 String[] answers = new String[4]**;** String query = "SELECT AnswerText FROM Answers WHERE QuestionID = " + getQuestionID()**;** Cursor cursor = database.rawQuery(query**,** null)**;** int i = 0**;** while (cursor.moveToNext()) {  
 answers[i] = cursor.getString(0)**;** i++**;** }  
 cursor.close()**;** return answers**;** }  
  
 private int getQuestionID() {  
 int questionID = 0**;** if (selectedCountry.equals("Россия")) {  
 questionID = 1**;** } else if (selectedCountry.equals("Франция")) {  
 questionID = 2**;** } else if (selectedCountry.equals("Великобритания")) {  
 questionID = 3**;** } else if (selectedCountry.equals("Италия")) {  
 questionID = 4**;** } else if (selectedCountry.equals("Япония")) {  
 questionID = 5**;** }  
 return questionID**;** }  
  
 private void checkAnswer(String selectedAnswer) {  
 String query = "SELECT CorrectAnswerID FROM Answers WHERE QuestionID = " + getQuestionID()**;** Cursor cursor = database.rawQuery(query**,** null)**;** int correctAnswerID = 0**;** if (cursor.moveToFirst()) {  
 correctAnswerID = cursor.getInt(0)**;** }  
 cursor.close()**;** // Получение текста правильного ответа на основе correctAnswerID  
 String correctAnswer = getAnswerText(correctAnswerID)**;** if (selectedAnswer.equals(correctAnswer)) {  
 Toast.*makeText*(this**,** "Вы выбрали правильный ответ"**,** Toast.*LENGTH\_SHORT*).show()**;** // Изменить состояние кнопок  
 answer1Button.setBackgroundColor(Color.*GREEN*)**;** answer2Button.setBackgroundColor(Color.*RED*)**;** answer3Button.setBackgroundColor(Color.*RED*)**;** } else {  
 Toast.*makeText*(this**,** "Вы выбрали неправильный ответ"**,** Toast.*LENGTH\_SHORT*).show()**;** // Изменить состояние кнопки с правильным ответом  
 if (correctAnswerID == 1) {  
 answer1Button.setBackgroundColor(Color.*GREEN*)**;** } else if (correctAnswerID == 2) {  
 answer2Button.setBackgroundColor(Color.*GREEN*)**;** } else if (correctAnswerID == 3) {  
 answer3Button.setBackgroundColor(Color.*GREEN*)**;** }  
 // Изменить состояние остальных кнопок  
 if (!answer1Button.getText().toString().equals(selectedAnswer)) {  
 answer1Button.setBackgroundColor(Color.*RED*)**;** }  
 if (!answer2Button.getText().toString().equals(selectedAnswer)) {  
 answer2Button.setBackgroundColor(Color.*RED*)**;** }  
 if (!answer3Button.getText().toString().equals(selectedAnswer)) {  
 answer3Button.setBackgroundColor(Color.*RED*)**;** }  
 }  
 }  
  
 private String getAnswerText(int answerID) {  
 String answerText = ""**;** String query = "SELECT AnswerText FROM Answers WHERE QuestionID = " + getQuestionID() + " AND AnswerID = " + answerID**;** Cursor cursor = database.rawQuery(query**,** null)**;** if (cursor.moveToFirst()) {  
 answerText = cursor.getString(0)**;** }  
 cursor.close()**;** return answerText**;** }  
 private String getCorrectAnswer() {  
 String correctAnswer = ""**;** String query = "SELECT AnswerText FROM Answers WHERE QuestionID = " + getQuestionID() + " AND AnswerID = (SELECT CorrectAnswerID FROM Answers WHERE QuestionID = " + getQuestionID() + ")"**;** Cursor cursor = database.rawQuery(query**,** null)**;** if (cursor.moveToFirst()) {  
 correctAnswer = cursor.getString(0)**;** }  
 cursor.close()**;** return correctAnswer**;** }  
 private void copyDatabaseFromAssets() {  
 try {  
 InputStream inputStream = getAssets().open("smol.db")**;** String outFileName = getDatabasePath("smol.db").getPath()**;** OutputStream outputStream = new FileOutputStream(outFileName)**;** byte[] buffer = new byte[1024]**;** int length**;** while ((length = inputStream.read(buffer)) > 0) {  
 outputStream.write(buffer**,** 0**,** length)**;** }  
 outputStream.flush()**;** outputStream.close()**;** inputStream.close()**;** } catch (IOException e) {  
 e.printStackTrace()**;** }  
 }  
}