|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №10**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  студент группыИКБО-28-22 | Некрасов Г.А. |
| **Проверил:**  к.э.н доцент | Петросян Л.Э. |

Москва 2024 г.

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc167627390)

[Работа с 1 пунктом задания 3](#_Toc167627391)

[Работа с 2 пунктом задания 5](#_Toc167627392)

[Тестирование работы 16](#_Toc167627393)

# Задание

* Реализовать запись, получение, изменение и удаление имени пользователя приложения через SharedPreference.
* Создать базу данных по выбранной тематике. Реализовать методы записи, поиска, изменения и удаления данных. В базе данных должна быть минимум одна таблица с 5 полями.

# Работа с 1 пунктом задания

Листинг 1. MainActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_10pract;  import android.content.Context; import android.content.SharedPreferences; import android.os.Bundle; import android.view.View; import android.widget.TextView; import android.widget.Toast; import android.content.Intent;  import androidx.appcompat.app.AppCompatActivity;  public class MainActivity extends AppCompatActivity {   private TextView textUserInfo;  private SharedPreferences sharedPreferences;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   textUserInfo = findViewById(R.id.*textUserInfo*);  sharedPreferences = getSharedPreferences("MyPreferences", Context.*MODE\_PRIVATE*);  }   // Метод для получения имени пользователя при нажатии кнопки "Получить имя"  public void getUserName(View view) {  SharedPreferences sharedPreferences = getSharedPreferences("MyPreferences", Context.*MODE\_PRIVATE*);  SharedPreferences.Editor editor = sharedPreferences.edit();   // Получение данных из SharedPreferences  String userName = sharedPreferences.getString("UserName", "Имя пользователя не указано");  String course = sharedPreferences.getString("Course", "Курс не указан");  String group = sharedPreferences.getString("Group", "Группа не указана");   // Сохранение изначальной информации  editor.putString("UserName", "Глеб");  editor.putString("Course", "2");  editor.putString("Group", "ИКБО-28-22");  editor.apply();   // Формирование и вывод информации  String userInfo = "Имя: " + userName + "\nНомер курса: " + course + "\nГруппа: " + group;  textUserInfo.setText(userInfo);  textUserInfo.setVisibility(View.*VISIBLE*);  // Добавляем уведомление о получении данных  showToast("Данные получены");  }   // Метод для удаления имени пользователя при нажатии кнопки "Удалить"  public void removeUserName(View view) {  SharedPreferences.Editor editor = sharedPreferences.edit();  editor.remove("UserName");  editor.remove("Course");  editor.remove("Group");  editor.apply();  textUserInfo.setText(""); // Очищаем поле с информацией  // Добавляем уведомление об удалении данных  showToast("Данные удалены");  }   // Метод для отображения уведомлений  private void showToast(String message) {  Toast.*makeText*(this, message, Toast.*LENGTH\_SHORT*).show();  }   public void goToSecondActivity(View view) {  Intent intent = new Intent(this, SecondActivity.class);  startActivity(intent);  } } |

Листинг 2. Разметка activity\_main.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">   <TextView  android:id="@+id/textUserInfo"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="100dp"  android:text=""  android:textSize="18sp" />   <Button  android:id="@+id/btnGet"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/textUserInfo"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp"  android:onClick="getUserName"  android:text="Получить имя" />   <Button  android:id="@+id/btnRemove"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/btnGet"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp"  android:onClick="removeUserName"  android:text="Удалить" />   <Button  android:id="@+id/btnSecondActivity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Перейти к второй активности"  android:onClick="goToSecondActivity"  android:layout\_alignParentBottom="true"  android:layout\_alignParentEnd="true"  android:layout\_marginBottom="16dp"  android:layout\_marginEnd="16dp"/>  </RelativeLayout> |

# Работа с 2 пунктом задания

Листинг 3. Contact.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_10pract;  public class Contact {  private int id;  private String name;  private int age;  private String email;  private String phone;   // Конструктор  public Contact(int id, String name, int age, String email, String phone) {  this.id = id;  this.name = name;  this.age = age;  this.email = email;  this.phone = phone;  }   // Геттеры и сеттеры  public int getId() {  return id;  }   public void setId(int id) {  this.id = id;  }   public String getName() {  return name;  }   public void setName(String name) {  this.name = name;  }   public int getAge() {  return age;  }   public void setAge(int age) {  this.age = age;  }   public String getEmail() {  return email;  }   public void setEmail(String email) {  this.email = email;  }   public String getPhone() {  return phone;  }   public void setPhone(String phone) {  this.phone = phone;  }   @Override  public String toString() {  return "Name: " + name + "; Age: " + age + "; Email: " + email + "; Phone: " + phone;  }  } |

Листинг 4. DatabaseHelper.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_10pract;  import android.content.ContentValues; import android.content.Context; import android.database.Cursor; import android.database.sqlite.SQLiteDatabase; import android.database.sqlite.SQLiteOpenHelper; import java.util.ArrayList; import java.util.List;  public class DatabaseHelper extends SQLiteOpenHelper {   private static final String DATABASE\_NAME = "Contacts.db";  private static final int DATABASE\_VERSION = 1;   // Имя таблицы и её столбцы  private static final String TABLE\_NAME = "Contacts";  private static final String COLUMN\_ID = "id";  private static final String COLUMN\_NAME = "name";  private static final String COLUMN\_AGE = "age";  private static final String COLUMN\_EMAIL = "email";  private static final String COLUMN\_PHONE = "phone";   // Запрос для создания таблицы  private static final String SQL\_CREATE\_TABLE =  "CREATE TABLE " + TABLE\_NAME + " (" +  COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +  COLUMN\_NAME + " TEXT," +  COLUMN\_AGE + " INTEGER," +  COLUMN\_EMAIL + " TEXT," +  COLUMN\_PHONE + " TEXT)";   // Конструктор  public DatabaseHelper(Context context) {  super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  }   @Override  public void onCreate(SQLiteDatabase db) {  // Создание таблицы  db.execSQL(SQL\_CREATE\_TABLE);  }   @Override  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  // Обновление базы данных (в данном примере - пусто)  }   // Добавление контакта в базу данных  public long addContact(Contact contact) {  SQLiteDatabase db = this.getWritableDatabase();  ContentValues values = new ContentValues();  values.put(COLUMN\_NAME, contact.getName());  values.put(COLUMN\_AGE, contact.getAge());  values.put(COLUMN\_EMAIL, contact.getEmail());  values.put(COLUMN\_PHONE, contact.getPhone());  long result = db.insert(TABLE\_NAME, null, values);  db.close();  return result;  }   // Получение всех контактов из базы данных  public List<Contact> getAllContacts() {  List<Contact> contactList = new ArrayList<>();  String selectQuery = "SELECT \* FROM " + TABLE\_NAME;  SQLiteDatabase db = this.getWritableDatabase();  Cursor cursor = db.rawQuery(selectQuery, null);  if (cursor.moveToFirst()) {  int idIndex = cursor.getColumnIndex(COLUMN\_ID);  int nameIndex = cursor.getColumnIndex(COLUMN\_NAME);  int ageIndex = cursor.getColumnIndex(COLUMN\_AGE);  int emailIndex = cursor.getColumnIndex(COLUMN\_EMAIL);  int phoneIndex = cursor.getColumnIndex(COLUMN\_PHONE);  do {  int id = cursor.getInt(idIndex);  String name = cursor.getString(nameIndex);  int age = cursor.getInt(ageIndex);  String email = cursor.getString(emailIndex);  String phone = cursor.getString(phoneIndex);  Contact contact = new Contact(id, name, age, email, phone);  contactList.add(contact);  } while (cursor.moveToNext());  }  cursor.close();  db.close();  return contactList;  }   // Обновление контакта в базе данных  public int updateContact(Contact contact) {  SQLiteDatabase db = this.getWritableDatabase();  ContentValues values = new ContentValues();  values.put(COLUMN\_NAME, contact.getName());  values.put(COLUMN\_AGE, contact.getAge());  values.put(COLUMN\_EMAIL, contact.getEmail());  values.put(COLUMN\_PHONE, contact.getPhone());  return db.update(TABLE\_NAME, values, COLUMN\_ID + " = ?",  new String[]{String.valueOf(contact.getId())});  }   // Удаление контакта по номеру телефона  public void deleteContactByPhone(String phone) {  SQLiteDatabase db = this.getWritableDatabase();  db.delete(TABLE\_NAME, COLUMN\_PHONE + " = ?", new String[]{phone});  db.close();  }   // Поиск контактов по номеру телефона  public List<Contact> searchContactsByPhone(String phone) {  List<Contact> contactList = new ArrayList<>();  SQLiteDatabase db = this.getWritableDatabase();  String[] columns = {COLUMN\_ID, COLUMN\_NAME, COLUMN\_AGE, COLUMN\_EMAIL, COLUMN\_PHONE};  String selection = COLUMN\_PHONE + " LIKE ?";  String[] selectionArgs = {"%" + phone + "%"};  Cursor cursor = db.query(TABLE\_NAME, columns, selection, selectionArgs,  null, null, null);   int columnIndexId = cursor.getColumnIndex(COLUMN\_ID);  int columnIndexName = cursor.getColumnIndex(COLUMN\_NAME);  int columnIndexAge = cursor.getColumnIndex(COLUMN\_AGE);  int columnIndexEmail = cursor.getColumnIndex(COLUMN\_EMAIL);  int columnIndexPhone = cursor.getColumnIndex(COLUMN\_PHONE);   while (cursor.moveToNext()) {  int id = cursor.getInt(columnIndexId);  String name = cursor.getString(columnIndexName);  int age = cursor.getInt(columnIndexAge);  String email = cursor.getString(columnIndexEmail);  String phoneNum = cursor.getString(columnIndexPhone);   Contact contact = new Contact(id, name, age, email, phoneNum);  contactList.add(contact);  }   cursor.close();  db.close();  return contactList;  } } |

Листинг 5. SecondActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_10pract;  import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.ListView; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity; import java.util.ArrayList; import java.util.List;  public class SecondActivity extends AppCompatActivity {   private ListView listViewContacts;  private EditText editTextName, editTextAge, editTextEmail, editTextPhone;  private Button buttonSave, buttonUpdate, buttonDelete, buttonFind;  private DatabaseHelper databaseHelper;  private List<Contact> contactList;  private ArrayAdapter<Contact> adapter;  private int selectedContactPosition = AdapterView.INVALID\_POSITION;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_second);   listViewContacts = findViewById(R.id.listViewContacts);  editTextName = findViewById(R.id.editTextName);  editTextAge = findViewById(R.id.editTextAge);  editTextEmail = findViewById(R.id.editTextEmail);  editTextPhone = findViewById(R.id.editTextPhone);  buttonSave = findViewById(R.id.buttonSave);  buttonUpdate = findViewById(R.id.buttonUpdate);  buttonDelete = findViewById(R.id.buttonDelete);  buttonFind = findViewById(R.id.buttonFind);   databaseHelper = new DatabaseHelper(this);   // Получение всех контактов из базы данных  contactList = databaseHelper.getAllContacts();   // Создание адаптера для списка контактов  adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, contactList);   // Установка адаптера для списка  listViewContacts.setAdapter(adapter);   // Обработчик нажатия на элемент списка  listViewContacts.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  // Сохраняем позицию выбранного контакта  selectedContactPosition = position;   // Получаем выбранный контакт из списка  Contact selectedContact = contactList.get(selectedContactPosition);   // Заполняем поля ввода данными выбранного контакта  editTextName.setText(selectedContact.getName());  editTextAge.setText(String.valueOf(selectedContact.getAge()));  editTextEmail.setText(selectedContact.getEmail());  editTextPhone.setText(selectedContact.getPhone());  }  });   // Обработчик нажатия на кнопку "Save"  buttonSave.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  // Получаем данные из полей ввода  String name = editTextName.getText().toString();  int age = Integer.parseInt(editTextAge.getText().toString());  String email = editTextEmail.getText().toString();  String phone = editTextPhone.getText().toString();   // Создаем новый контакт с автоматически сгенерированным id  // В данном случае, так как мы добавляем новый контакт, id не имеет значения,  // поэтому мы можем передать любое значение, например, -1  Contact newContact = new Contact(-1, name, age, email, phone);   // Добавляем контакт в базу данных  long result = databaseHelper.addContact(newContact);   if (result != -1) {  // Обновляем список контактов  contactList.clear();  contactList.addAll(databaseHelper.getAllContacts());  adapter.notifyDataSetChanged();   // Выводим Toast сообщение  Toast.makeText(SecondActivity.this, "Contact saved successfully", Toast.LENGTH\_SHORT).show();  }  }  });   // Обработчик нажатия на кнопку "Update Contact"  buttonUpdate.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  // Проверяем, что позиция выбрана  if (selectedContactPosition != AdapterView.INVALID\_POSITION) {  // Получаем данные из полей ввода  int id = contactList.get(selectedContactPosition).getId();  String name = editTextName.getText().toString();  int age = Integer.parseInt(editTextAge.getText().toString());  String email = editTextEmail.getText().toString();  String phone = editTextPhone.getText().toString();   // Создаем объект Contact с обновленными данными  Contact updatedContact = new Contact(id, name, age, email, phone);   // Обновляем контакт в базе данных  int rowsAffected = databaseHelper.updateContact(updatedContact);   if (rowsAffected > 0) {  // Обновляем список контактов  contactList.clear();  contactList.addAll(databaseHelper.getAllContacts());  adapter.notifyDataSetChanged();   // Выводим Toast сообщение  Toast.makeText(SecondActivity.this, "Contact updated successfully", Toast.LENGTH\_SHORT).show();  }  }  }  });   // Обработчик нажатия на кнопку "Delete"  buttonDelete.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  // Получаем номер телефона из поля ввода  String phone = editTextPhone.getText().toString();   // Удаляем контакт из базы данных по номеру телефона  databaseHelper.deleteContactByPhone(phone);   // Обновляем список контактов  contactList.clear();  contactList.addAll(databaseHelper.getAllContacts());  adapter.notifyDataSetChanged();   // Выводим Toast сообщение  Toast.makeText(SecondActivity.this, "Contact deleted successfully", Toast.LENGTH\_SHORT).show();  }  });   // Обработчик нажатия на кнопку "Find"  buttonFind.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  // Получаем номер телефона из поля ввода  String phone = editTextPhone.getText().toString();   // Поиск контактов по номеру телефона  contactList.clear();  contactList.addAll(databaseHelper.searchContactsByPhone(phone));  adapter.notifyDataSetChanged();   // Выводим Toast сообщение  Toast.makeText(SecondActivity.this, "Contacts found successfully", Toast.LENGTH\_SHORT).show();  }  });  } } |

Листинг 6. Разметка activity\_second.java

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <EditText  android:id="@+id/editTextName"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Name"  android:layout\_marginStart="16dp"  android:layout\_marginEnd="16dp"  android:layout\_marginTop="16dp" />   <EditText  android:id="@+id/editTextAge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Age"  android:layout\_below="@id/editTextName"  android:layout\_marginStart="16dp"  android:layout\_marginEnd="16dp"  android:layout\_marginTop="-5dp" />   <EditText  android:id="@+id/editTextEmail"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Email"  android:layout\_below="@id/editTextAge"  android:layout\_marginStart="16dp"  android:layout\_marginEnd="16dp"  android:layout\_marginTop="-5dp" />   <EditText  android:id="@+id/editTextPhone"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Phone"  android:layout\_below="@id/editTextEmail"  android:layout\_marginStart="16dp"  android:layout\_marginEnd="16dp"  android:layout\_marginTop="-5dp" />   <Button  android:id="@+id/buttonSave"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Save"  android:layout\_below="@id/editTextPhone"  android:layout\_marginStart="16dp"  android:layout\_marginTop="-5dp" />   <Button  android:id="@+id/buttonUpdate"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Update Contact"  android:layout\_below="@id/buttonSave"  android:layout\_marginStart="16dp"  android:layout\_marginTop="-5dp" />   <Button  android:id="@+id/buttonDelete"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/buttonUpdate"  android:layout\_marginStart="16dp"  android:layout\_marginTop="-5dp"  android:layout\_marginEnd="16dp"  android:layout\_marginBottom="16dp"  android:text="Delete by phone" />   <Button  android:id="@+id/buttonFind"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/buttonDelete"  android:layout\_marginStart="16dp"  android:layout\_marginTop="-20dp"  android:layout\_marginEnd="16dp"  android:layout\_marginBottom="16dp"  android:text="Find by phone" />   <ListView  android:id="@+id/listViewContacts"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_below="@id/buttonFind"  android:layout\_marginStart="16dp"  android:layout\_marginEnd="16dp"  android:layout\_marginTop="16dp"  android:layout\_marginBottom="-5dp" />  </RelativeLayout> |

# Тестирование работы

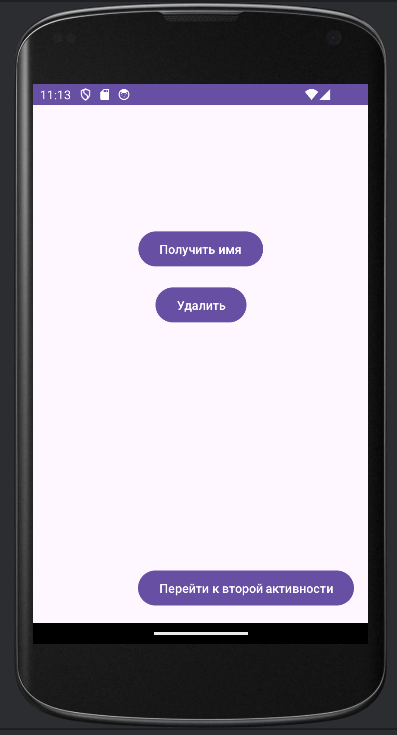


Рисунок 1. Запуск первой активности

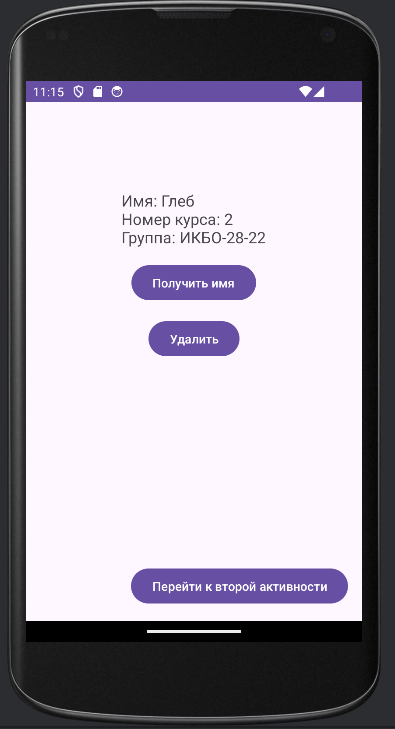


Рисунок 2. Получение имени

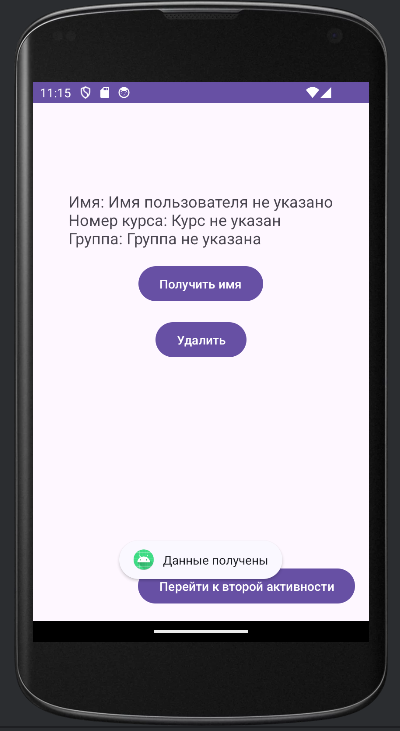


Рисунок 3. Удаление и последующие получение имени

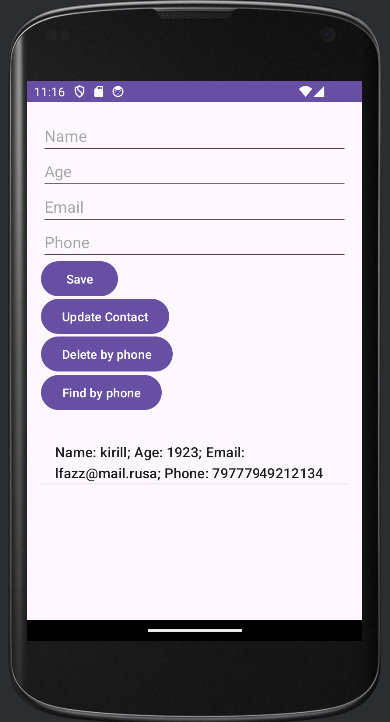


Рисунок 4. Переход на вторую активность

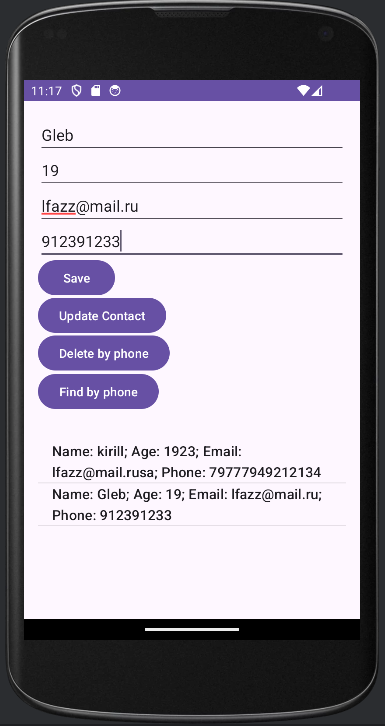


Рисунок 5. Добавление новой записи с помощью кнопки Save

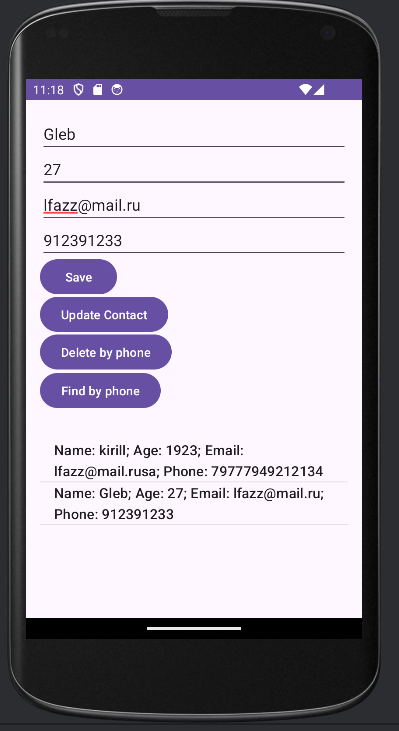


Рисунок 6. Изменение, например возраста с помощью кнопки Update Contact

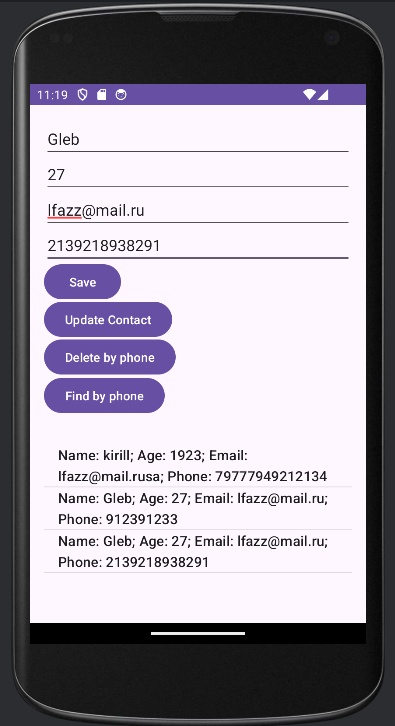


Рисунок 7. Пример создания одинаковых записей с разными телефонами

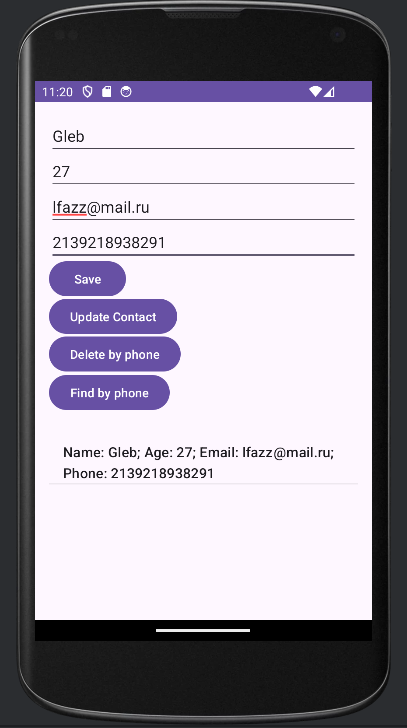


Рисунок 8. Проверка работы кнопки Find by phone (поиск по номеру телефона)

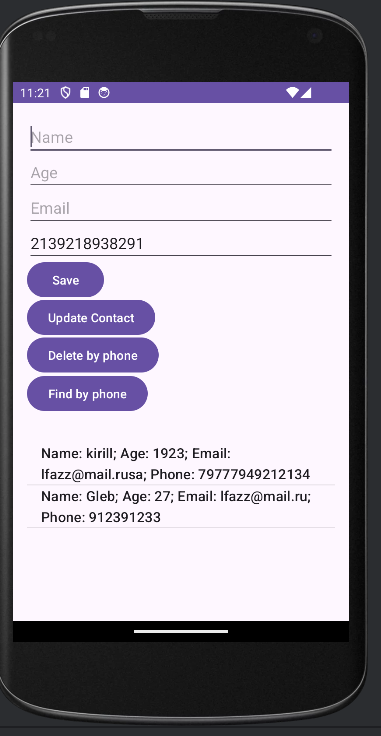


Рисунок 9. Проверка работы кнопки Delete by phone (удаление по номеру телефона)