Министерство науки и высшего образования Российской Федерации

Федерального государственного бюджетное образовательное учреждение высшего образования

«Российской экономический университет имени Г. В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

ПРАКТИЧЕСКИЕ РАБОТЫ № 7

ДИСЦИПЛИНА: «РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»

Тема: Работа с датчиками и создание гироскопа.

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Листов:

|  |  |  |
| --- | --- | --- |
| Выполнил студент |  | Проверил преподаватель |
| Группа П50-3-18 |  | \_\_\_\_\_\_\_\_\_\_\_А.О.Лясников |
| Пахомов Даниил Александрович |  | «\_\_\_» \_\_\_\_\_\_\_\_2020 года |

Москва 2020

ПРАКТИЧЕСКАЯ РАБОТА №6

Работа с базой данных SQLite.

Цель работы: научится создавать приложения, которые могут взаимодействовать с базой данных, которая будет создавать, удалять обновлять, выводить данные.

1. MainActivity

|  |
| --- |
| package com.example.sqlite  import android.database.Cursor import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.widget.Toast import androidx.appcompat.app.AlertDialog import kotlinx.android.synthetic.main.activity\_main.\*  class MainActivity : AppCompatActivity() {  var peopleDB: DatabaseHelper? = null   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)  peopleDB = DatabaseHelper(this@MainActivity)  btnAdd.setOnClickListener { addUser() }  btnShow.setOnClickListener { showUsers() }   btnDelete.setOnClickListener { deleteUsers() }  btnRefresh.setOnClickListener { updateUsers() }  }   private fun updateUsers() {  peopleDB!!.update(edtName.text.toString(),edtEmail.text.toString(),ID.text.toString().toInt())  Toast.makeText(this, "Данные обновлены", Toast.LENGTH\_SHORT).show()  }   private fun deleteUsers() {  val data = peopleDB!!.poisk(ID.text.toString().toInt())  val buffer = StringBuffer()  while(data!!.moveToNext())  {  buffer.append("ID: ${data.getString(0)} \n")  buffer.append("Name: ${data.getString(1)} \n")  buffer.append("Email: ${data.getString(2)} \n")  }  if(peopleDB!!.delete(ID.text.toString().toInt()) == true)  {  display("Удален пользователь", buffer.toString())  }else  {  Toast.makeText(this, "Нет такого пользователя", Toast.LENGTH\_SHORT).show()  }  }   private fun showUsers() {  val data: Cursor? = peopleDB!!.showData()   if (data!!.getCount() == 0) {  display("Error", "нет данных")  return  }  val buffer = StringBuffer()  while(data.moveToNext())  {  buffer.append("ID: ${data.getString(0)} \n")  buffer.append("Name: ${data.getString(1)} \n")  buffer.append("Email: ${data.getString(2)} \n")  }   display("Все пользователи", buffer.toString())  }  private fun addUser() {  val name: String = edtName.text.toString()  val email: String = edtEmail.text.toString()   val insertData: Boolean = peopleDB!!.addData(name, email)   if (insertData == true) {  Toast.makeText(this, "Запись добавлена", Toast.LENGTH\_SHORT).show()  } else {  Toast.makeText(this, "Что-то пошло не так", Toast.LENGTH\_SHORT).show()  }  }   fun display(title: String, message: String) {  val bulder: AlertDialog.Builder = AlertDialog.Builder(this)  bulder.setCancelable(true)  bulder.setTitle(title)  bulder.setMessage(message)  bulder.show()  } } |

1. DatabaseHelper

|  |
| --- |
| package com.example.sqlite  import android.content.ContentValues import android.content.Context import android.database.Cursor import android.database.sqlite.SQLiteDatabase import android.database.sqlite.SQLiteOpenHelper  class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, "people.db", null, 1) {   val TABLE\_NAME = "people\_table"  val COL2 = "NAME"  val COL3 = "EMAIL"   private val DROP\_USER\_TABLE = "DROP TABLE IF EXISTS $TABLE\_NAME"   override fun onCreate(db: SQLiteDatabase?) {  val createTable =  "CREATE TABLE $TABLE\_NAME (ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, EMAIL TEXT, TVSHOW TEXT)"  db?.execSQL(createTable);  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db?.execSQL(DROP\_USER\_TABLE)  onCreate(db)  }   fun addData(name: String?, email: String?):Boolean  {  val db = this.writableDatabase  val contentValues = ContentValues()  contentValues.put(COL2,name)  contentValues.put(COL3,email)  val result = db.insert(TABLE\_NAME,null, contentValues)  return result != -1L;  }   fun showData(): Cursor?{  val db = this.writableDatabase  return db.rawQuery("SELECT \* FROM $TABLE\_NAME",null)  }   fun poisk(id: Int?):Cursor  {  val db = this.writableDatabase  return db.rawQuery("SELECT \* FROM $TABLE\_NAME WHERE ID = $id",null)  }   fun delete(id: Int?):Boolean  {  val db = this.writableDatabase  val count = db.delete(TABLE\_NAME,"ID = $id" ,null)  return count != 0  }   fun update(name: String?, Email:String?,id:Int?)  {  var cv = ContentValues()  cv.put(COL2,name)  cv.put(COL3,Email)  var db = this.writableDatabase  db.update(TABLE\_NAME,cv,"ID = $id",null)  } } |

1. activity\_main

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">  <EditText  android:id="@+id/edtName"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:hint="Имя"  app:layout\_constraintTop\_toTopOf="parent" />  <EditText  android:id="@+id/edtEmail"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:hint="E-mail"  app:layout\_constraintTop\_toTopOf="@id/edtName" />  <Button  android:id="@+id/btnAdd"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:text="Создать запись"  app:layout\_constraintTop\_toBottomOf="@id/edtEmail" />  <Button  android:id="@+id/btnShow"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:text="Посмотреть записи"  app:layout\_constraintTop\_toBottomOf="@id/btnAdd" />  <Button  android:id="@+id/btnDelete"  android:layout\_width="270dp"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:text="Удалить запись"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toStartOf="@id/ID"  app:layout\_constraintTop\_toBottomOf="@id/btnShow" />  <EditText  android:id="@+id/ID"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:hint="ID delete"  app:layout\_constraintStart\_toEndOf="@id/btnDelete"  app:layout\_constraintTop\_toBottomOf="@id/btnShow"  app:layout\_constraintBottom\_toTopOf="@id/btnRefresh"/>  <Button  android:id="@+id/btnRefresh"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:text="Обновить"  app:layout\_constraintTop\_toBottomOf="@id/btnDelete" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Вывод: Научился создавать программу которая может взаимодействовать с SQLite (добавлять, удалять, обновлять, выводить данные)