

Отчёт по индивидуальному практическому заданию №2

Цель контрольной работы.

Целью выполнения индивидуальной практической работы является закрепление материала теоретического курса. Работа со встроенной библиотекой управления базой данных SQLite.

Задание

В соответствии с индивидуальным заданием создать приложение, которое позволяет просматривать и редактировать списки различных объектов (товаров, сотрудников и т.п.). Данные для списков хранить в базе данных SQLite

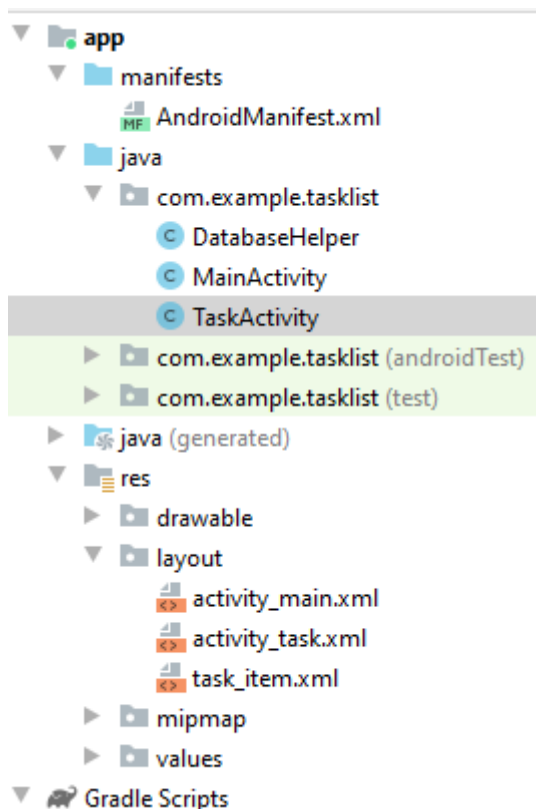
Вариант 4.

Список заказов (id, клиент, дата исполнения, стоимость). Создание, редактирование, удаление.

Исходный код и видео работы программы есть на гит-хабе:

<https://github.com/ElephantT/Android>

Структура приложения:



DatabaseHelper - описание нашей базы данных

MainActivity - работа с главной страницей, где показаны все клиенты, и можно открыть каждый заказ и посмотреть его, а также можно сохранить новый

TaskActivity - работа с заказами, можно изменить и сохранить

MainActivity

```
public class MainActivity extends AppCompatActivity {

    ListView taskList;
    DatabaseHelper databaseHelper;
    SQLiteDatabase db;
    Cursor taskCursor;
    SimpleCursorAdapter taskAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        taskList = (ListView)findViewById(R.id.list);
        taskList.setOnItemClickListener((parent, view, position, id) → {
            Intent intent = new Intent(getApplicationContext(), TaskActivity.class);
            intent.putExtra( name: "id", id);
            startActivity(intent);
        });

        databaseHelper = new DatabaseHelper(getApplicationContext());
    }
}
```

Создание базы данных и ListView

```
@Override
public void onResume() {
    super.onResume();
    db = databaseHelper.getReadableDatabase();

    taskCursor = db.rawQuery( sql: "select * from "+ DatabaseHelper.TABLE, selectionArgs: null);
    String[] headers = new String[] {DatabaseHelper.COLUMN_NAME, DatabaseHelper.COLUMN_YEAR,
        DatabaseHelper.COLUMN_COST};
    taskAdapter = new SimpleCursorAdapter( context: this, R.layout.task_item,
        taskCursor, headers, new int[]{R.id.name_text, R.id.year_text, R.id.cost_text}, flags: 0);
    taskList.setAdapter(taskAdapter);
}
```

Подключаемся к базе данных и работаем с ней через курсор.

```

public void add(View view){
    Intent intent = new Intent( packageContext: this, TaskActivity.class);
    startActivity(intent);
}

@Override
public void onDestroy(){
    super.onDestroy();
    db.close();
    taskCursor.close();
}

```

Нажатие на кнопку “Add” и закрытие приложения (с закрытием подключения и курсора)

Xml-код

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add"
        android:onClick="add"
        android:textSize="18dp"/>
    <ListView
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

```

Ячейка в списке - xml-код:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/name_text"
        android:textSize="26sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/year_text"
        android:textSize="26sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cost_text"
        android:textSize="26sp" />

</LinearLayout>

```

DatabaseHelper

```

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "tasks.db";
    private static final int VERSION = 1;
    static final String TABLE = "tasks";

    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_YEAR = "year";
    public static final String COLUMN_COST = "cost";

    public DatabaseHelper(Context context) { super(context, DATABASE_NAME, factory: null, VERSION); }
}

```

```

public void onCreate(SQLiteDatabase db) {

    db.execSQL("CREATE TABLE tasks (" + COLUMN_ID
        + " INTEGER PRIMARY KEY AUTOINCREMENT," + COLUMN_NAME
        + " TEXT, " + COLUMN_YEAR + " INTEGER, " + COLUMN_COST + " INTEGER);");

    db.execSQL("INSERT INTO " + TABLE + " (" + COLUMN_NAME
        + ", " + COLUMN_YEAR + ", " + COLUMN_COST + ") VALUES ('Иван Иванович', 1945, 3);");
    db.execSQL("INSERT INTO " + TABLE + " (" + COLUMN_NAME
        + ", " + COLUMN_YEAR + ", " + COLUMN_COST + ") VALUES ('Александр', 1401, 100);");

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE);
    onCreate(db);
}

```

Создание бд с начальными значениями и её обновление

TaskActivity

Xml-код обработки ячейки

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"/>
    <EditText
        android:id="@+id/year"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Data"/>
    <EditText
        android:id="@+id/cost"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Cost"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

```

```

        <Button
            android:id="@+id/saveButton"
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:text="Save"
            android:onClick="save"/>
        <Button
            android:id="@+id/deleteButton"
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:text="Delete"
            android:onClick="delete"/>
    </LinearLayout>
</LinearLayout>

public class TaskActivity extends AppCompatActivity {

    EditText nameBox;
    EditText yearBox;
    EditText costBox;
    Button delButton;
    Button saveButton;

    DatabaseHelper sqlHelper;
    SQLiteDatabase db;
    Cursor userCursor;
    long userId = 0;

```

Переменный для работы с ячейкой

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_task);

    nameBox = (EditText) findViewById(R.id.name);
    yearBox = (EditText) findViewById(R.id.year);
    costBox = (EditText) findViewById(R.id.cost);
    delButton = (Button) findViewById(R.id.deleteButton);
    saveButton = (Button) findViewById(R.id.saveButton);

    sqlHelper = new DatabaseHelper( context: this);
    db = sqlHelper.getWritableDatabase();

    Bundle extras = getIntent().getExtras();
    if (extras != null) {
        userId = extras.getLong( key: "id");
    }

    if (userId > 0) {
        userCursor = db.rawQuery( sql: "select * from " + DatabaseHelper.TABLE + " where " +
            DatabaseHelper.COLUMN_ID + "=?", new String[]{String.valueOf(userId)});
        userCursor.moveToFirst();
        nameBox.setText(userCursor.getString( i: 1));
        yearBox.setText(String.valueOf(userCursor.getInt( i: 2)));
        costBox.setText(String.valueOf(userCursor.getInt( i: 3)));
        userCursor.close();
    } else {
        delButton.setVisibility(View.GONE);
    }
}

```

Создание новой ячейки (тогда убираем видимость кнопки удаления), или открытие старой

```

public void save(View view){
    ContentValues cv = new ContentValues();
    cv.put(DatabaseHelper.COLUMN_NAME, nameBox.getText().toString());
    cv.put(DatabaseHelper.COLUMN_YEAR, Integer.parseInt(yearBox.getText().toString()));
    cv.put(DatabaseHelper.COLUMN_COST, Integer.parseInt(costBox.getText().toString()));

    if (userId > 0) {
        db.update(DatabaseHelper.TABLE, cv, whereClause: DatabaseHelper.COLUMN_ID + "=",
            + String.valueOf(userId), whereArgs: null);
    } else {
        db.insert(DatabaseHelper.TABLE, nullColumnHack: null, cv);
    }
    toMainPage();
}

public void delete(View view){
    db.delete(DatabaseHelper.TABLE, whereClause: "_id = ?", new String[]{String.valueOf(userId)});
    toMainPage();
}

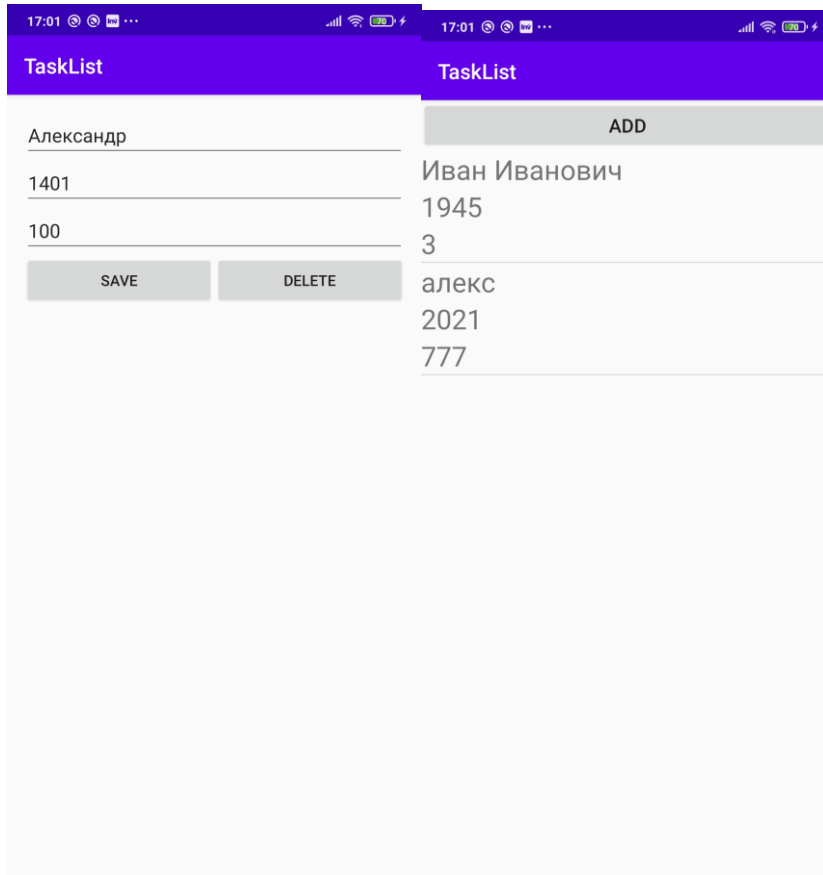
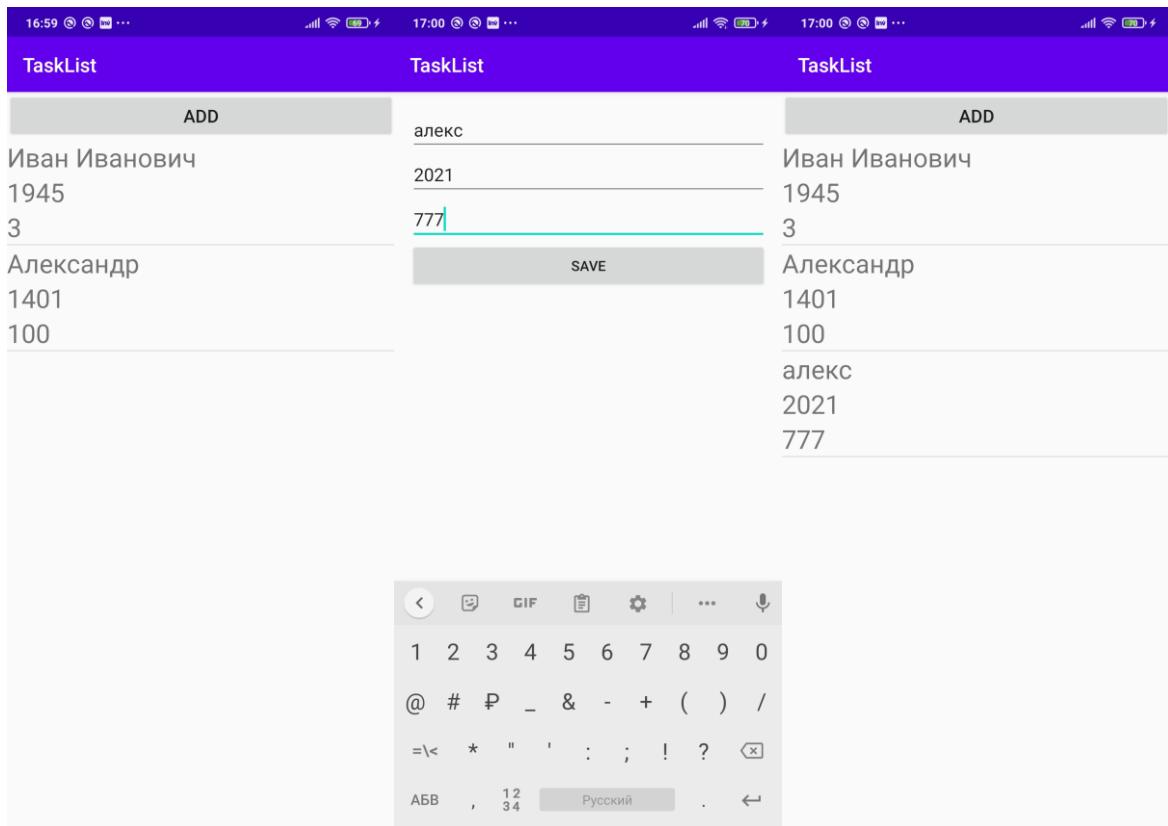
private void toMainPage(){
    db.close();
    Intent intent = new Intent( packageContext: this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
    startActivity(intent);
}
}

```

Активация Windows

Сохранение новой или изменений старой ячейки, удаление старой и возврат на главную страницу.

Скриншоты работы приложения (можете так же посмотреть видео на гитхабе, перейдя по ссылке в отчёте в начале или в комментарии к ипр2):



Александр Дубейковский, студент 893551