

Санкт-Петербургское Государственное
бюджетное профессиональное образовательное учреждение
«Колледж информационных технологий»

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7

МДК 01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Специальность 09.02.07 Информационные системы и программирование

Выполнил студент
гр. 393: Баланин Э.Н.

Санкт-Петербург 2021

Интерфейс

Был разработан интерфейс приложения «Database». Интерфейс главной Activity продемонстрирован на рисунке 1.

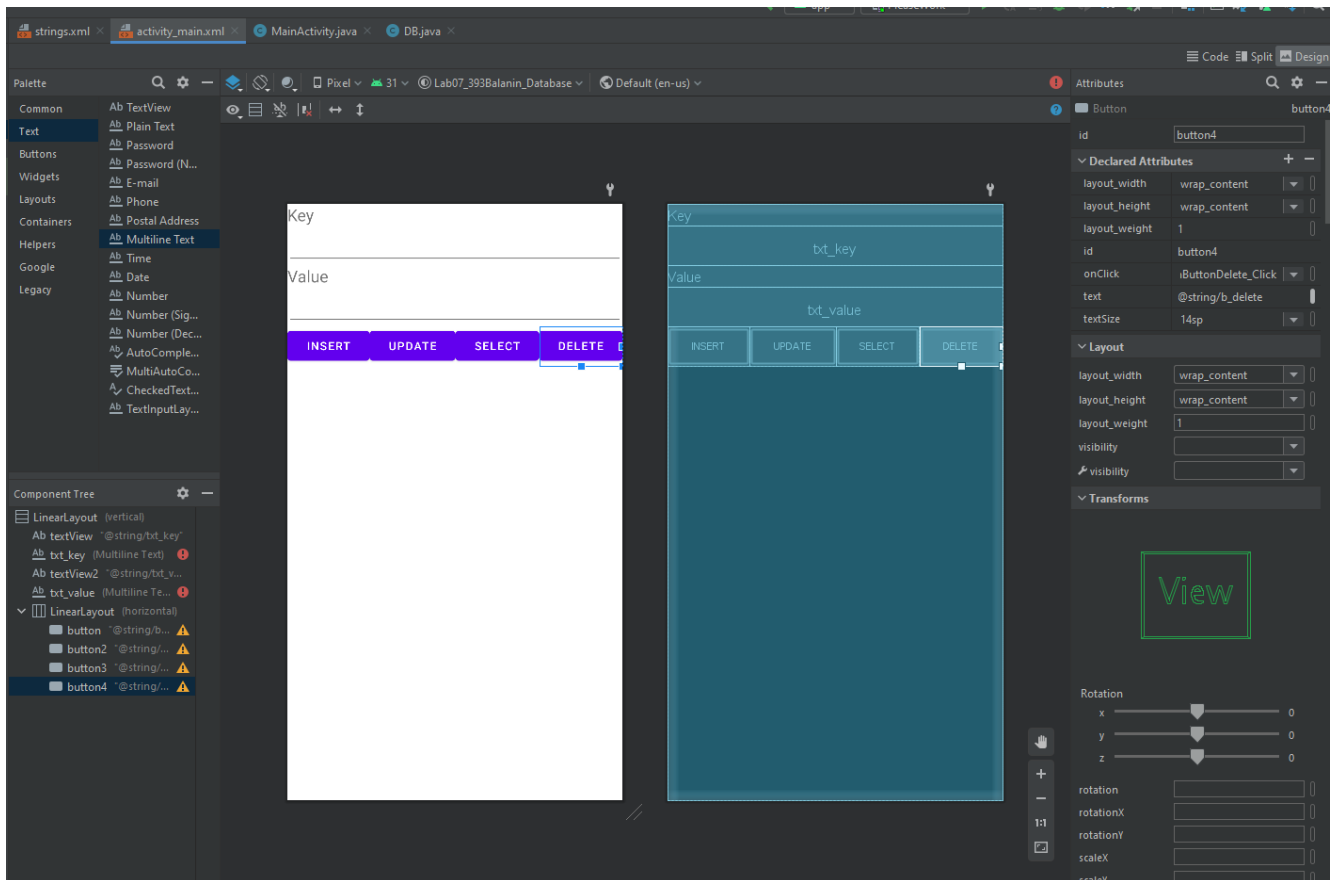


Рисунок 1 – Интерфейс Activity

Интерфейс Activity разработан следующим образом: в древе компонентов Constraint Layout был заменен на Linear Layout (Vertical), были добавлены 2 TextView и EditText компонента, еще 1 Linear Layout (Horizontal) для четырех кнопок.

Каждому добавленному компоненту были созданы собственные строковые ресурсы для надписей кнопок и текстовых полей. Список созданных строковых ресурсов продемонстрирован на рисунке 2.

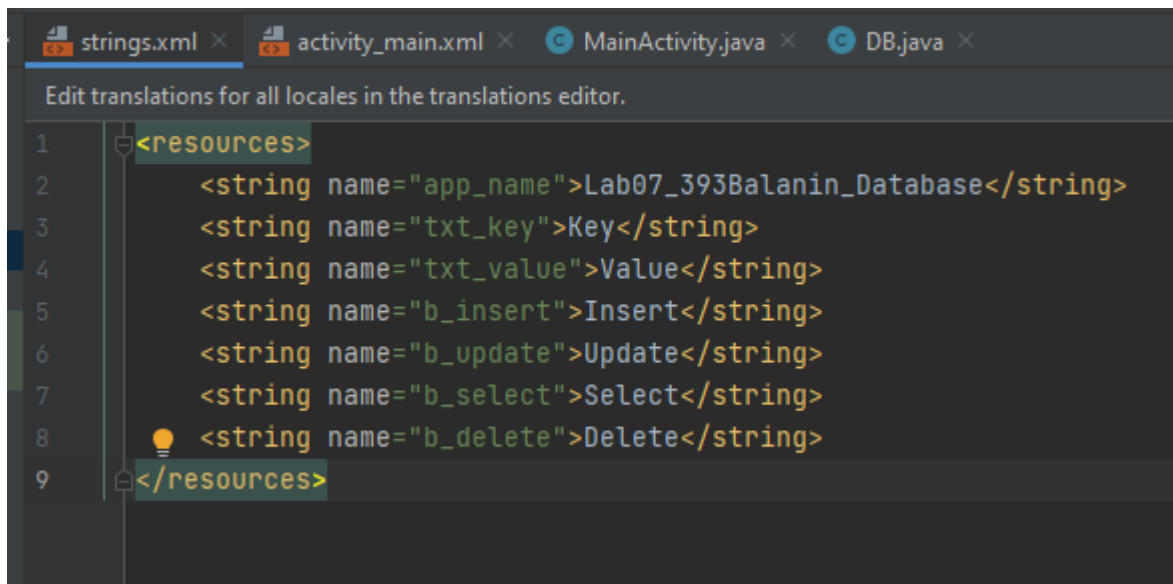
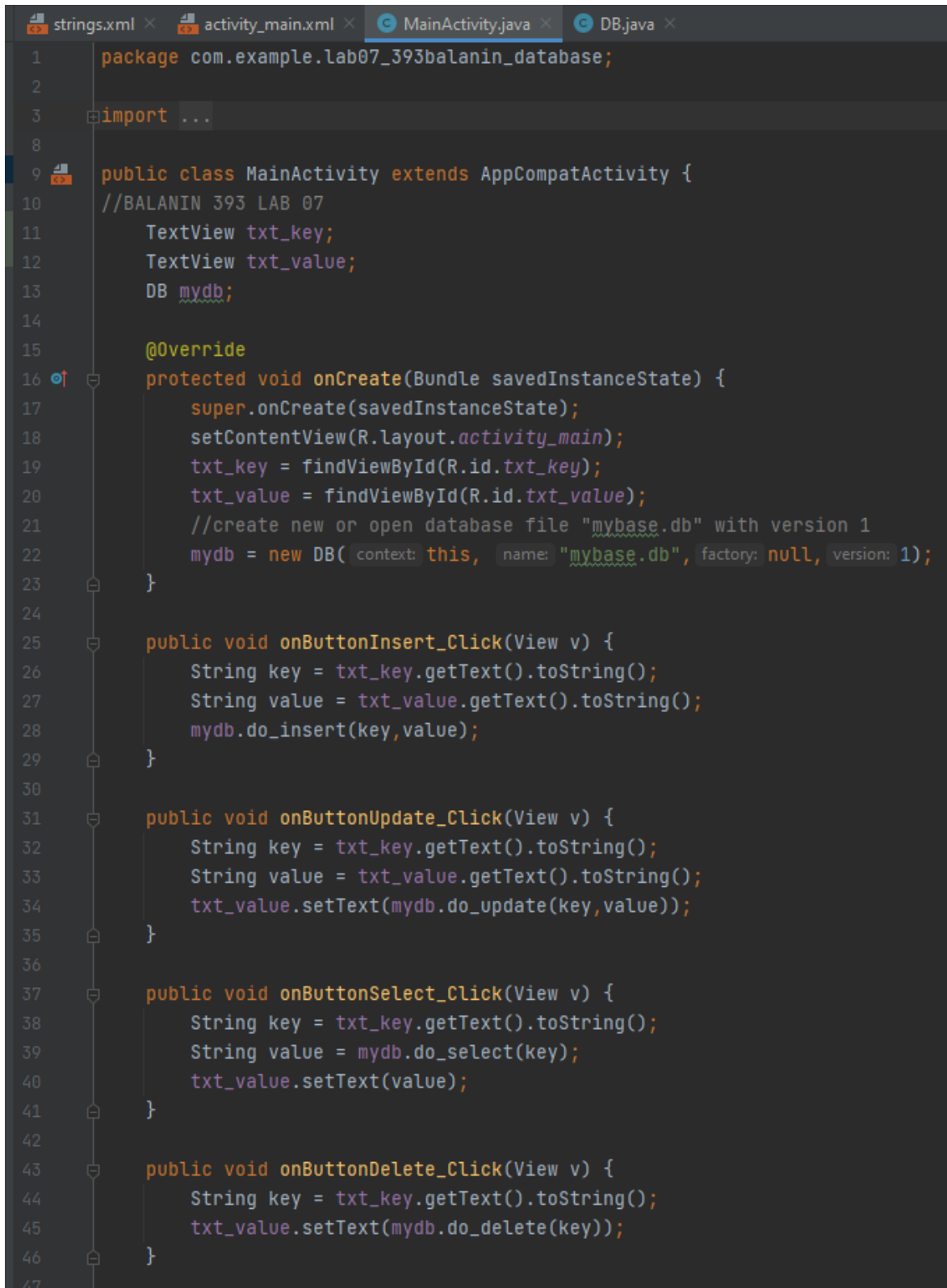


Рисунок 2 – Созданные строковые ресурсы

Каждый строковый ресурс был привязан к соответствующему компоненту интерфейса (как можно заметить на рисунке 1).

Программный код

Для работы приложения был разработан следующий программный код. Программный код главной Activity – Main Activity представлен на рисунке 3.



```

1 package com.example.lab07_393balanin_database;
2
3 import ...
4
5
6
7
8
9 public class MainActivity extends AppCompatActivity {
10     //BALANIN 393 LAB 07
11     TextView txt_key;
12     TextView txt_value;
13     DB mydb;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         txt_key = findViewById(R.id.txt_key);
20         txt_value = findViewById(R.id.txt_value);
21         //create new or open database file "mybase.db" with version 1
22         mydb = new DB(context: this, name: "mybase.db", factory: null, version: 1);
23     }
24
25     public void onButtonInsert_Click(View v) {
26         String key = txt_key.getText().toString();
27         String value = txt_value.getText().toString();
28         mydb.do_insert(key, value);
29     }
30
31     public void onButtonUpdate_Click(View v) {
32         String key = txt_key.getText().toString();
33         String value = txt_value.getText().toString();
34         txt_value.setText(mydb.do_update(key, value));
35     }
36
37     public void onButtonSelect_Click(View v) {
38         String key = txt_key.getText().toString();
39         String value = mydb.do_select(key);
40         txt_value.setText(value);
41     }
42
43     public void onButtonDelete_Click(View v) {
44         String key = txt_key.getText().toString();
45         txt_value.setText(mydb.do_delete(key));
46     }
47

```

Рисунок 3 – Программный код Main Activity

Данный код состоит из нескольких методов:

- `OnCreate` – в данном методе инициализируются объекты классов `TextView` и также создается экземпляр класса `DB` (код данного класса продемонстрирован на рисунке 4);
- `OnButtonInsert_Click` – вводит новую запись в базу данных `mydb`;
- `OnButtonUpdate_Click` – вызывает метод класса `DB` для замены значения в столбце таблицы;
- `OnButtonSelect_Click` – вызывает метод класса `DB` для вывода значения определенного ключа;
- `OnButtonDelete_Click` – вызывает метод класса `DB` для удаления определенной строки.

Программный код класса `DB` продемонстрирован на рисунке 4.

```

strings.xml × activity_main.xml × MainActivity.java × DB.java ×
1 package com.example.lab07_393balanin_database;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.database.sqlite.SQLiteOpenHelper;
7
8 import androidx.annotation.Nullable;
9
10 public class DB extends SQLiteOpenHelper {
11     public DB(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
12         super(context, name, factory, version);
13     }
14
15     @Override
16     public void onCreate(SQLiteDatabase db) {
17         String sql = "CREATE TABLE my_test (my_key TEXT, my_value TEXT);"; //creating table with 2 columns
18         db.execSQL(sql);
19     }
20
21     public void do_insert(String key, String value) {
22
23         String sql = "INSERT INTO my_test VALUES (" + key + ", " + value + ")"; //insert new row into table
24         SQLiteDatabase db = getWritableDatabase();
25         db.execSQL(sql);
26     }
27
28     public String do_select(String key) {
29
30         String sql = "SELECT my_value FROM my_test WHERE my_key = '" + key + "'"; //finds value inside table by given key
31         SQLiteDatabase db = getWritableDatabase(); //get ready to read from database
32         Cursor cur = db.rawQuery(sql, selectionArgs: null); //run query and acquire results in a new "table" with only 1 column
33
34         if (cur.moveToFirst() == true) //jump to the first (and the only one) matching record, if possible
35             return cur.getString(columnIndex: 0); // return value from the first column (my_value)
36
37         return "(!) not found";
38     }
39
40     public String do_update(String key, String value) {
41
42         String sql = "UPDATE my_test SET my_value = '" + value + "' WHERE my_key = '" + key + "'"; //update matching records
43         SQLiteDatabase db = getWritableDatabase(); //get ready to read from database
44         db.execSQL(sql);
45         return "record updated, check it out";
46     }
47
48     public String do_delete(String key) {
49
50         String sql = "DELETE FROM my_test WHERE my_key = '" + key + "'"; //delete matching records
51         SQLiteDatabase db = getWritableDatabase(); //get ready to read from database
52         db.execSQL(sql);
53         return "this record is gone";
54     }
55
56     @Override
57     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
58     }
59 }

```

Рисунок 4 - Программный код второй Activity

Данный программный код состоит из следующих функций:

- onCreate – создание самой базы данных для хранения информации;
- do_insert – происходит запись новой строки в базу данных;

- do_update – происходит замена значения столбца value в строках с схожим значением key;
- do_select – вывод значения value определенной строки;
- do_delete – удаление строк с указанным значением key.

Интерфейс приложения продемонстрирован на рисунке 5.

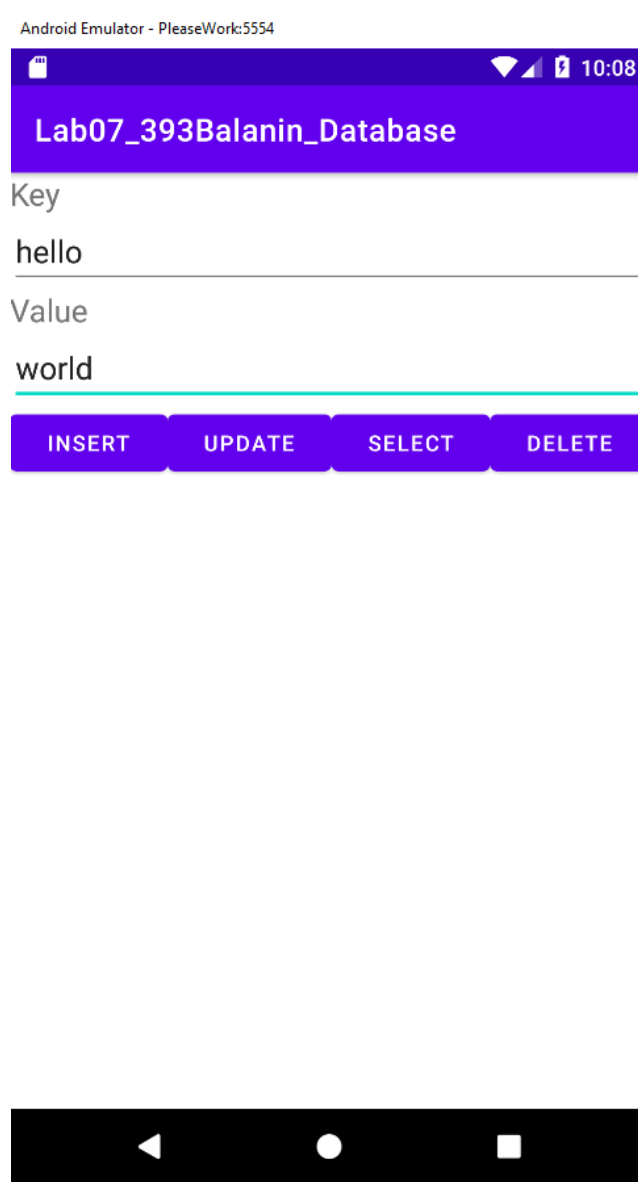


Рисунок 5 – Интерфейс приложения

Проблематично продемонстрировать работу данного приложения на картинках, поэтому прилагается репозиторий github с разработанной программой: https://github.com/b11st/Lab07_393Balanin_DB