**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: **«Разработка приложений для мобильных устройств»**

на тему: **Использование шаблона проектирования *MVC* в разработке *Android* приложений**

Выполнил: студент гр. ИТП-31

Дашкевич Д.А.

Принял: ассистент

Белявский Е. В.

Гомель 2019

**Цель**: изучить архитектуру «Модель-Представление-Контроллер» в *Android*.

**Ход работы**

**Вариант 2**

# **Задание:**

Разработать *Android* приложение с использованием архитектуры *MVC*, выполняющее ввод данных, вывод и редактирование в соответствии с вариантом. Для выполнения каждого пункта задания использовать отдельную *Activity* и модель. Выполнить запуск приложения на эмуляторе.

2. *Customer*: *id*, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счёта.

Создать массив объектов. Вывести:

а) список покупателей в алфавитном порядке;

б) список покупателей, у которых номер кредитной карточки находится в заданном интервале.

**Результат работы приложения:**

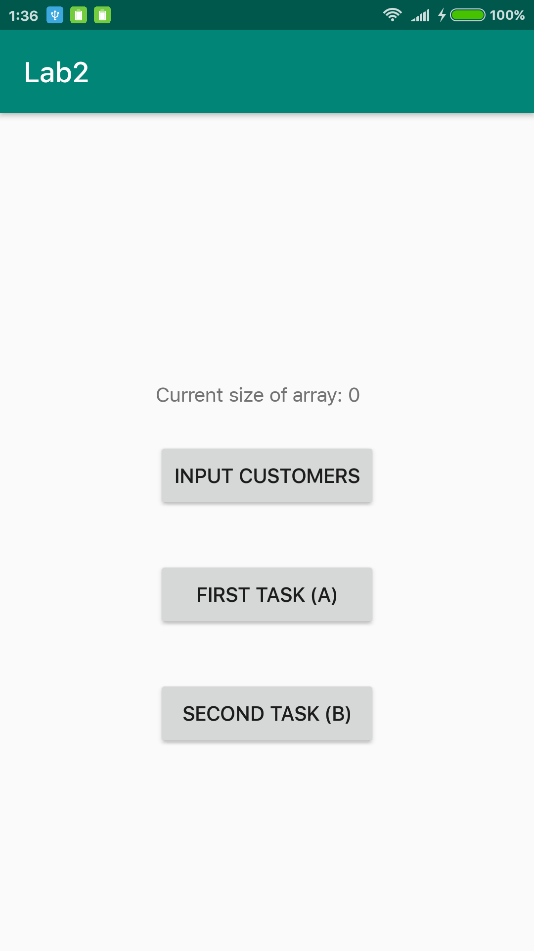


Рисунок 1 – Первоначальный экран приложения

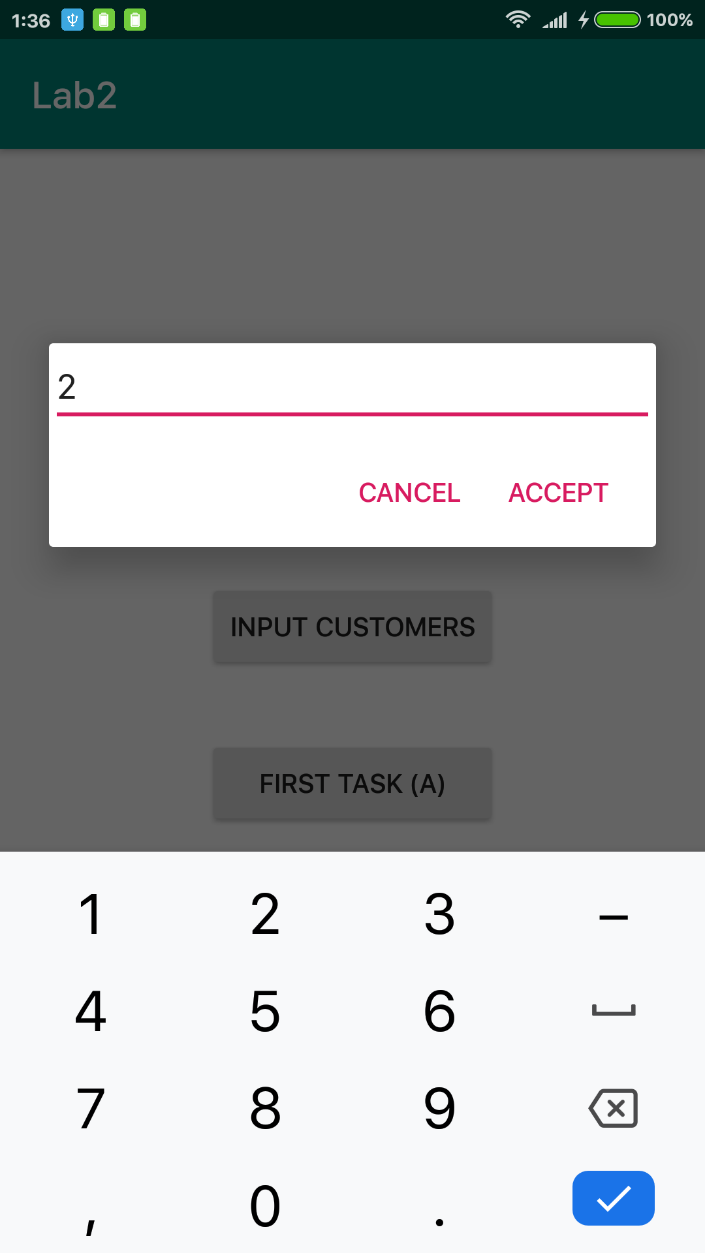


Рисунок 2 – Диалоговое окно ввода количества клиентов

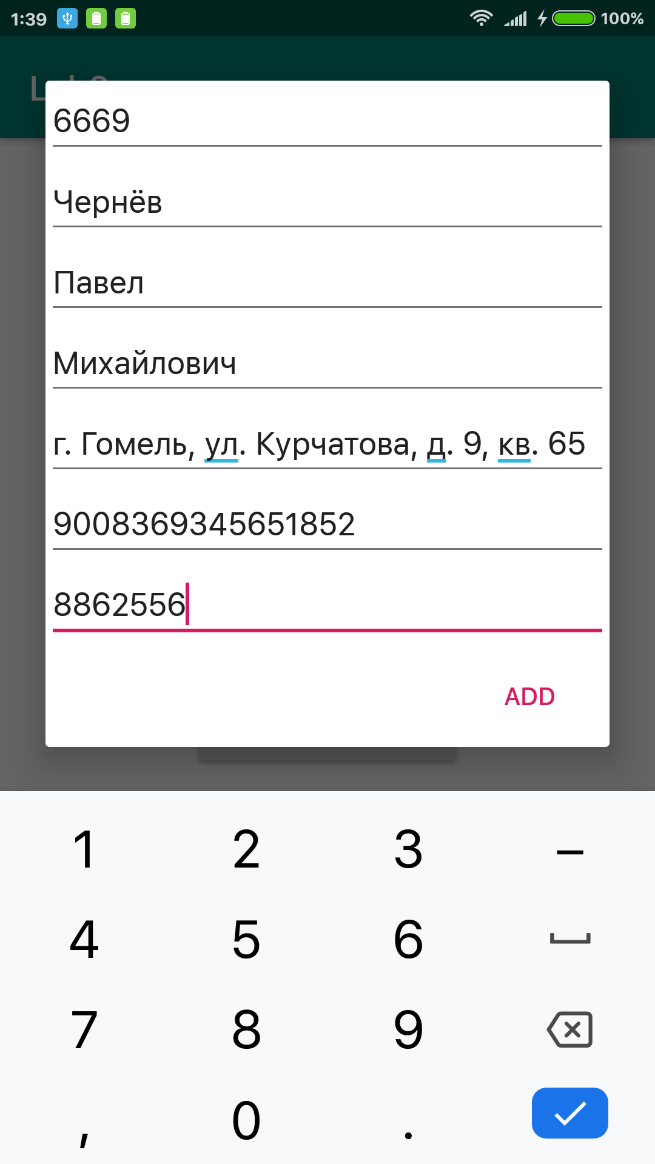


Рисунок 3 – Диалоговое окно ввода данных клиента

**Исходный код приложения**

**MainActivity.java:**

**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** TextView **mCurrentSizeTextView**;  
  
 @SuppressLint(**"SetTextI18n"**)  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 bindButtons();  
  
 **mCurrentSizeTextView** = findViewById(R.id.***currentSizeOfArrayTextView***);  
 **mCurrentSizeTextView**.setText(getString(R.string.***current\_size\_of\_array***) + **" 0"**);  
 }  
  
 **private void** bindButtons() {  
 Button inputButton = findViewById(R.id.***inputCustomersButton***);  
 Button openFirstActivityButton = findViewById(R.id.***firstTaskButton***);  
 Button openSecondActivityButton = findViewById(R.id.***secondTaskButton***);  
  
 inputButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 MainActivity.**this**.openInputArraySizeDialog();  
 }  
 });  
  
 openFirstActivityButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (CustomerList.*getInstance*() != **null**) {  
 startActivity(**new** Intent(MainActivity.**this**, FirstActivity.**class**));  
 } **else** {  
 Toast.*makeText*(v.getContext(), **"No data available!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
 });  
  
 openSecondActivityButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (CustomerList.*getInstance*() != **null**) {  
 startActivity(**new** Intent(MainActivity.**this**, SecondActivity.**class**));  
 } **else** {  
 Toast.*makeText*(v.getContext(), **"No data available!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 }  
  
  
 **private void** openInputArraySizeDialog() {  
 createInputArraySizeDialog().show();  
 }  
  
 @SuppressLint(**"InflateParams"**)  
 **private** AlertDialog createInputArraySizeDialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 LayoutInflater inflater = **this**.getLayoutInflater();  
 **final** View view = inflater.inflate(R.layout.***input\_array\_size\_dialog***, **null**);  
 builder.setView(view)  
 .setPositiveButton(**"Accept"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 acceptArraySizeDialog(view);  
 }  
 }).setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 }).setCancelable(**false**);  
 **return** builder.create();  
 }  
  
 @SuppressLint(**"SetTextI18n"**)  
 **private void** acceptArraySizeDialog(View view) {  
 EditText sizeOfArrayEditText = view.findViewById(R.id.***sizeOfArrayEditText***);  
 String s = sizeOfArrayEditText.getText().toString();  
 **if** (!s.isEmpty()) {  
 **int** size = Integer.*valueOf*(s);  
 **if** (size > 0) {  
 CustomerList.*createInstance*(size);  
 **mCurrentSizeTextView**.setText(getString(R.string.***current\_size\_of\_array***) + **" "** + size);  
 openInputCustomerDialog();  
 } **else** {  
 Toast.*makeText*(getApplicationContext(), **"Size must be positive!"**,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 } **else** {  
 Toast.*makeText*(getApplicationContext(), **"The input field must be filled in!"**,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
  
 **private void** openInputCustomerDialog() {  
 AlertDialog dialog = createInputCustomerDialog();  
 dialog.show();  
 }  
  
 @SuppressLint(**"InflateParams"**)  
 **private** AlertDialog createInputCustomerDialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 LayoutInflater inflater = **this**.getLayoutInflater();  
 **final** View view = inflater.inflate(R.layout.***input\_customer\_dialog***, **null**);  
 builder.setView(view)  
 .setPositiveButton(**"Add"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 addCustomer(view);  
 **if** (CustomerList.*getInstance*().getSize() != CustomerList.*getInstance*().getRecentlyAddedIndex()) {  
 openInputCustomerDialog();  
 }  
 }  
 }).setCancelable(**false**);  
 **return** builder.create();  
 }  
  
 **private void** addCustomer(View view) {  
 String customerSurname = ((EditText)  
 view.findViewById(R.id.***customerSurnameEditText***)).getText().toString();  
 String customerName = ((EditText)  
 view.findViewById(R.id.***customerNameEditText***)).getText().toString();  
 String customerMiddlename = ((EditText)  
 view.findViewById(R.id.***customerMiddlenameEditText***)).getText().toString();  
 String customerAddress = ((EditText)  
 view.findViewById(R.id.***customerAddressEditText***)).getText().toString();  
 **try** {  
 **int** customerId = Integer.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerIdEditText***)).getText().toString());  
 **long** customerCreditCardNumber = Long.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerCreditCardNumberEditText***)).getText().toString());  
 **long** customerBankAccountNumber = Long.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerBankAccountNumberEditText***)).getText().toString());  
 CustomerList.*getInstance*().addCustomer(**new** Customer(customerId, customerSurname,  
 customerName, customerMiddlename, customerAddress, customerCreditCardNumber,  
 customerBankAccountNumber));  
 } **catch** (NumberFormatException ex) {  
 Toast.*makeText*(**this**, **"Input fields must be filled in!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
}

**Customer.java:**

**public class** Customer {  
  
 **private int mId**;  
 **private** String **mSurname**;  
 **private** String **mName**;  
 **private** String **middlename**;  
 **private** String **mAddress**;  
 **private long mCreditCardNumber**;  
 **private long mBankAccountNumber**;  
  
 **public** Customer(**int** id, String surname, String name, String middlename, String address,  
 **long** creditCardNumber, **long** bankAccountNumber) {  
 **this**.**mId** = id;  
 **this**.**mSurname** = surname;  
 **this**.**mName** = name;  
 **this**.**middlename** = middlename;  
 **this**.**mAddress** = address;  
 **this**.**mCreditCardNumber** = creditCardNumber;  
 **this**.**mBankAccountNumber** = bankAccountNumber;  
 }  
  
 **public int** getId() {  
 **return mId**;  
 }  
  
 **public** String getSurname() {  
 **return mSurname**;  
 }  
  
 **public** String getName() {  
 **return mName**;  
 }  
  
 **public** String getMiddlename() {  
 **return middlename**;  
 }  
  
 **public** String getAddress() {  
 **return mAddress**;  
 }  
  
 **public long** getCreditCardNumber() {  
 **return mCreditCardNumber**;  
 }  
  
 **public long** getBankAccountNumber() {  
 **return mBankAccountNumber**;  
 }  
}

**CustomerList.java:**

**public class** CustomerList {  
  
 **private static** CustomerList *sCustomerList* = **null**;  
  
 **private** Customer[] **mArray**;  
 **private int mRecentlyAddedIndex** = 0;  
  
 **private** CustomerList(**int** maxSize) {  
 **this**.**mArray** = **new** Customer[maxSize];  
 }  
  
 **public static void** createInstance(**int** size) {  
 *sCustomerList* = **new** CustomerList(size);  
 }  
  
 **public static** CustomerList getInstance() {  
 **return** *sCustomerList*;  
 }  
  
 **public void** addCustomer(Customer customer) {  
 **if** (**mRecentlyAddedIndex** < **mArray**.**length**) {  
 **mArray**[**mRecentlyAddedIndex**++] = customer;  
 }  
 }  
  
 **public int** getRecentlyAddedIndex() {  
 **return mRecentlyAddedIndex**;  
 }  
  
 **public int** getSize() {  
 **return mArray**.**length**;  
 }  
  
 **public** Customer[] getCustomerArrayInAlphabeticalOrder() {  
 Customer[] customers = Arrays.*copyOf*(**mArray**, **mArray**.**length**);  
 Arrays.*sort*(customers, **new** Comparator<Customer>() {  
 @Override  
 **public int** compare(Customer o1, Customer o2) {  
 **return** o1.getSurname().compareTo(o2.getSurname());  
 }  
 });  
 **return** customers;  
 }  
  
 **public** Customer[] getCustomerArrayWithCardNumbersInRange(**long** begin, **long** end) {  
 ArrayList<Customer> list = **new** ArrayList<>();  
 **for** (Customer customer : **mArray**) {  
 **if** (customer.getCreditCardNumber() >= begin && customer.getCreditCardNumber() <= end) {  
 list.add(customer);  
 }  
 }  
 Customer[] customers = **new** Customer[list.size()];  
 customers = list.toArray(customers);  
 **return** customers;  
 }  
}

**activity\_main.xml:**

*<?***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=".controller.MainActivity"**>  
  
 <**TextView  
 android:id="@+id/currentSizeOfArrayTextView"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="180dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/current\_size\_of\_array"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/inputCustomersButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="220dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/input\_customers"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/firstTaskButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="300dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/first\_task\_a"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/secondTaskButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="380dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/second\_task\_b"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**input\_array\_size\_dialog.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**EditText  
 android:id="@+id/sizeOfArrayEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_size\_of\_array"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
</**LinearLayout**>

**input\_customer\_dialog.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**EditText  
 android:id="@+id/customerIdEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_id"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerSurnameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_surname"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerNameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_name"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerMiddlenameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_middlename"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerAddressEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_address"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerCreditCardNumberEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_credit\_card\_number"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerBankAccountNumberEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_bank\_account\_number"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
</**LinearLayout**>

**Вывод:** в результате выполнения лабораторной работы была изучена архитектура «Модель-Представление-Контроллер» в *Android*, также написано и запущено приложение.