**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

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

на тему: **Жизненный цикл *Activity***

Выполнил: студент гр. ИТП-31

Дашкевич Д.А.

Принял: ассистент

Белявский Е. В.

Гомель 2019

**Цель**: изучить жизненный цикл *Activity*.

**Ход работы**

**Вариант 2**

# **Задание:**

Разработать *Android* приложение, выполняющее ввод данных, вывод и редактирование в соответствии с вариантом. При запуске программы выполнять чтение из файла. При окончании работы программы сохранять данные в файл. Изменения состояния жизненного цикла *Activity* записывать в журнал. 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 – Журнал изменений жизненного цикла *Activity*

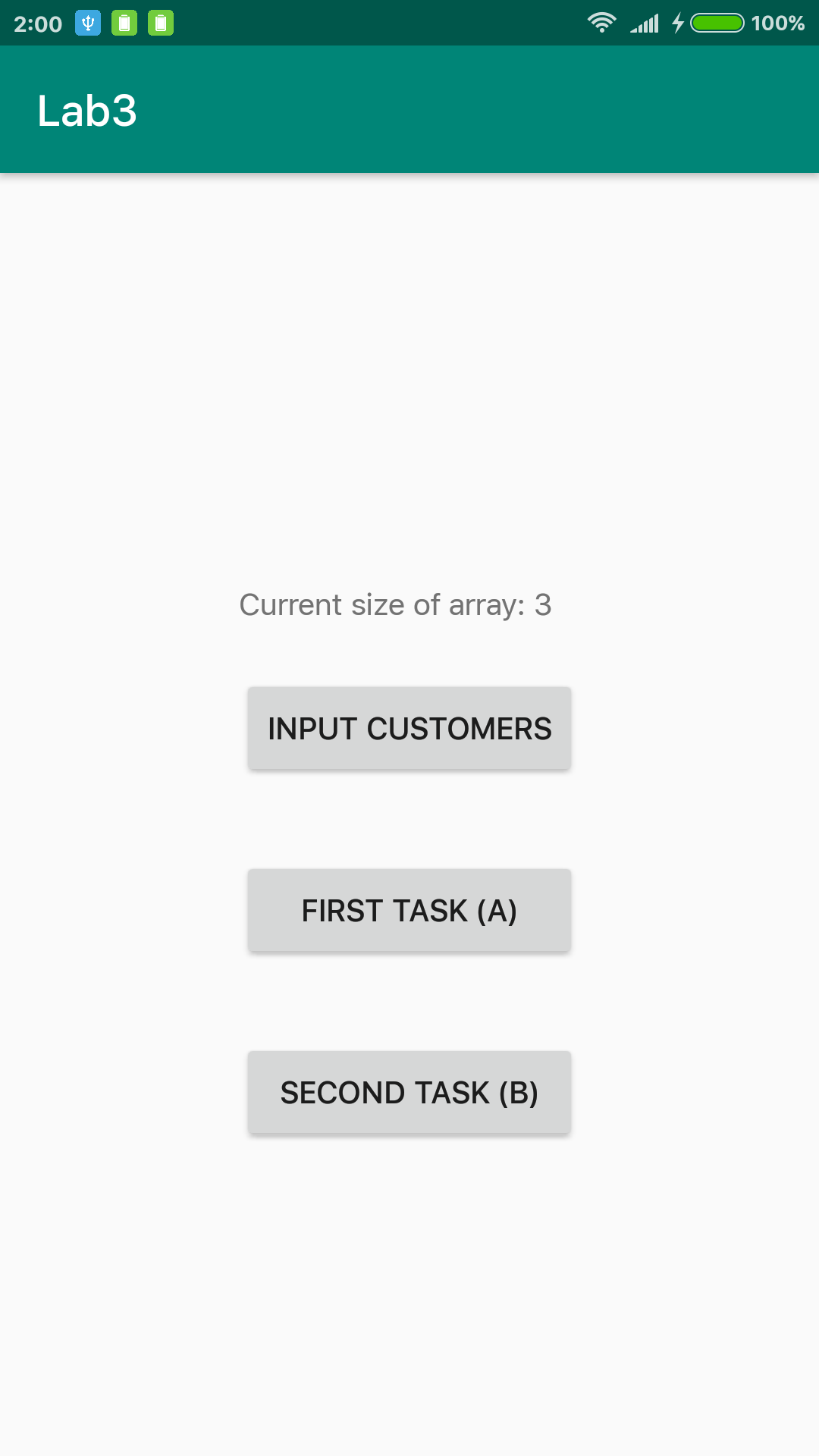


Рисунок 2 – Главное окно приложения с меню, которое было открыто после сохранения трёх записей

**Исходный код приложения**

**MainActivity.java:**

**public class MainActivity extends AppCompatActivity {  
  
 private static final String CURRENT\_SIZE\_STRING = "Current size of array: ";  
  
 private TextView mCurrentSizeTextView;  
  
 @Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**Log.d("MAIN", "onCreate");**

**setContentView(R.layout.activity\_main);**

**bindButtons();**

**mCurrentSizeTextView = findViewById(R.id.currentSizeOfArrayTextView);**

**}**

**@Override**

**protected void onStart() {**

**super.onStart();**

**Log.d("MAIN", "onStart");**

**}**

**@Override**

**protected void onResume() {**

**super.onResume();**

**int size = CustomerList.readFile(this);**

**String t = CURRENT\_SIZE\_STRING + size;**

**mCurrentSizeTextView.setText(t);**

**Log.d("MAIN", "onResume");**

**}**

**@Override**

**protected void onPause() {**

**super.onPause();**

**Log.d("MAIN", "onPause");**

**}**

**@Override**

**protected void onStop() {**

**super.onStop();**

**CustomerList.writeFile(this);**

**Log.d("MAIN", "onStop");**

**}**

**@Override**

**protected void onDestroy() {**

**Log.d("MAIN", "onDestroy");**

**super.onDestroy();**

**}  
  
 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();  
 }  
  
 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);  
 String t = CURRENT\_SIZE\_STRING + size;  
 mCurrentSizeTextView.setText(t);  
 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 final** String ***FILENAME*** = **"file.txt"**;  
  
 **private static** CustomerList *sCustomerList* = **null**;  
  
 **private** List<Customer> **mCustomers**;  
 **private int mRecentlyAddedIndex** = 0;  
  
 **private** CustomerList(**int** maxSize) {  
 **this**.**mCustomers** = **new** ArrayList<>(maxSize);  
 }  
  
 **public static void** createInstance(**int** size) {  
 *sCustomerList* = **new** CustomerList(size);  
 }  
  
 **public static** CustomerList getInstance() {  
 **return** *sCustomerList*;  
 }  
  
 **public void** addCustomer(Customer customer) {  
 **if** (**mRecentlyAddedIndex** < **mCustomers**.size()) {  
 **mCustomers**.set(**mRecentlyAddedIndex**++, customer);  
 }  
 }  
  
 **private** List<Customer> getArray() {  
 **return mCustomers**;  
 }  
  
 **public int** getRecentlyAddedIndex() {  
 **return mRecentlyAddedIndex**;  
 }  
  
 **public int** getSize() {  
 **return mCustomers**.size();  
 }  
  
 **public** List<Customer> getCustomerArrayInAlphabeticalOrder() {  
 List<Customer> customers = **new** ArrayList<>(**mCustomers**);  
 Collections.*sort*(customers, **new** Comparator<Customer>() {  
 @Override  
 **public int** compare(Customer o1, Customer o2) {  
 **return** o1.getSurname().compareTo(o2.getSurname());  
 }  
 });  
 **return** customers;  
 }  
  
 **public** List<Customer> getCustomerArrayWithCardNumbersInRange(**long** begin, **long** end) {  
 List<Customer> customers = **new** ArrayList<>();  
 **for** (Customer customer : **mCustomers**) {  
 **if** (customer.getCreditCardNumber() >= begin && customer.getCreditCardNumber() <= end) {  
 customers.add(customer);  
 }  
 }  
 **return** customers;  
 }  
  
 **public static void** writeFile(Context context) {  
 **if** (CustomerList.*getInstance*() == **null**) {  
 **return**;  
 }  
 **int** size = CustomerList.*getInstance*().getSize();  
 **if** (size > 0) {  
 **try** {  
 ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(context  
 .openFileOutput(***FILENAME***, Context.***MODE\_PRIVATE***));  
 objectOutputStream.writeInt(CustomerList.*getInstance*().getSize());  
 List<Customer> array = CustomerList.*getInstance*().getArray();  
 **for** (**int** i = 0; i < size; i++) {  
 objectOutputStream.writeObject(array.get(i));  
 }  
 objectOutputStream.close();  
 } **catch** (IOException ex) {  
 Log.*d*(**"MAIN"**, ex.getMessage());  
 }  
 }  
 }  
  
 **public static int** readFile(Context context) {  
 **try** {  
 ObjectInputStream objectInputStream = **new** ObjectInputStream(context.openFileInput(***FILENAME***));  
 **int** size = objectInputStream.readInt();  
 CustomerList.*createInstance*(size);  
 **for** (**int** i = 0; i < size; i++) {  
 CustomerList.*getInstance*().addCustomer((Customer) objectInputStream.readObject());  
 }  
 **return** size;  
 } **catch** (IOException | ClassNotFoundException ex) {  
 Log.*d*(**"MAIN"**, ex.getMessage() == **null** ? **""** : ex.getMessage());  
 **return** 0;  
 }  
 }  
}

**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 был изучен жизненный цикл *Activity*, также написано и запущено приложение, которое производит запись и чтение данных в файл.