

# Activity Lifecycle – part 2

android



# План презентации

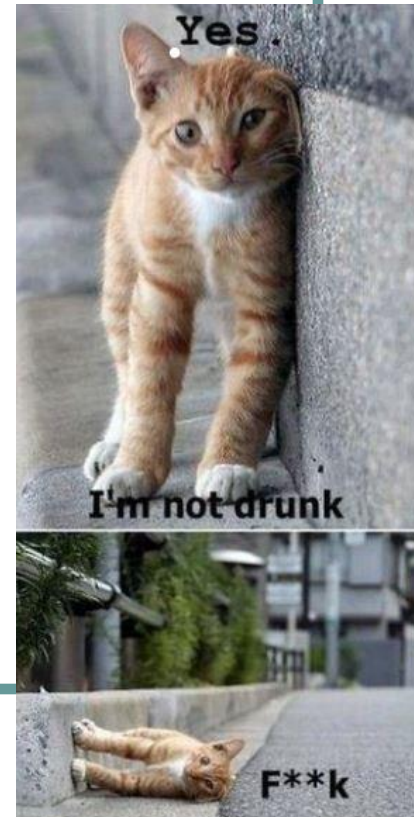
- Сохранение состояния при повороте
- Создание второго активити
- Передача данных во второе активити
- Отправка почты с помощью интента
- Flashlight v.2.0
- Голосовой ввод текста
- Практика

# На прошлом занятии мы...

- Создали поле класса **int count = 0;**
- Сделали кнопку, и обработчик **onClick**
- В обработчике написали:

```
@Override  
public void onClick(View view) {  
    count++;  
    Toast.makeText(this, count + "", Toast.LENGTH_SHORT).show();  
}
```

- нажимали на кнопку несколько раз
- Поворачивали экран
- Потом снова нажимали на кнопку
- И выяснили, что значение сбрасывается.



# Сохранение состояния

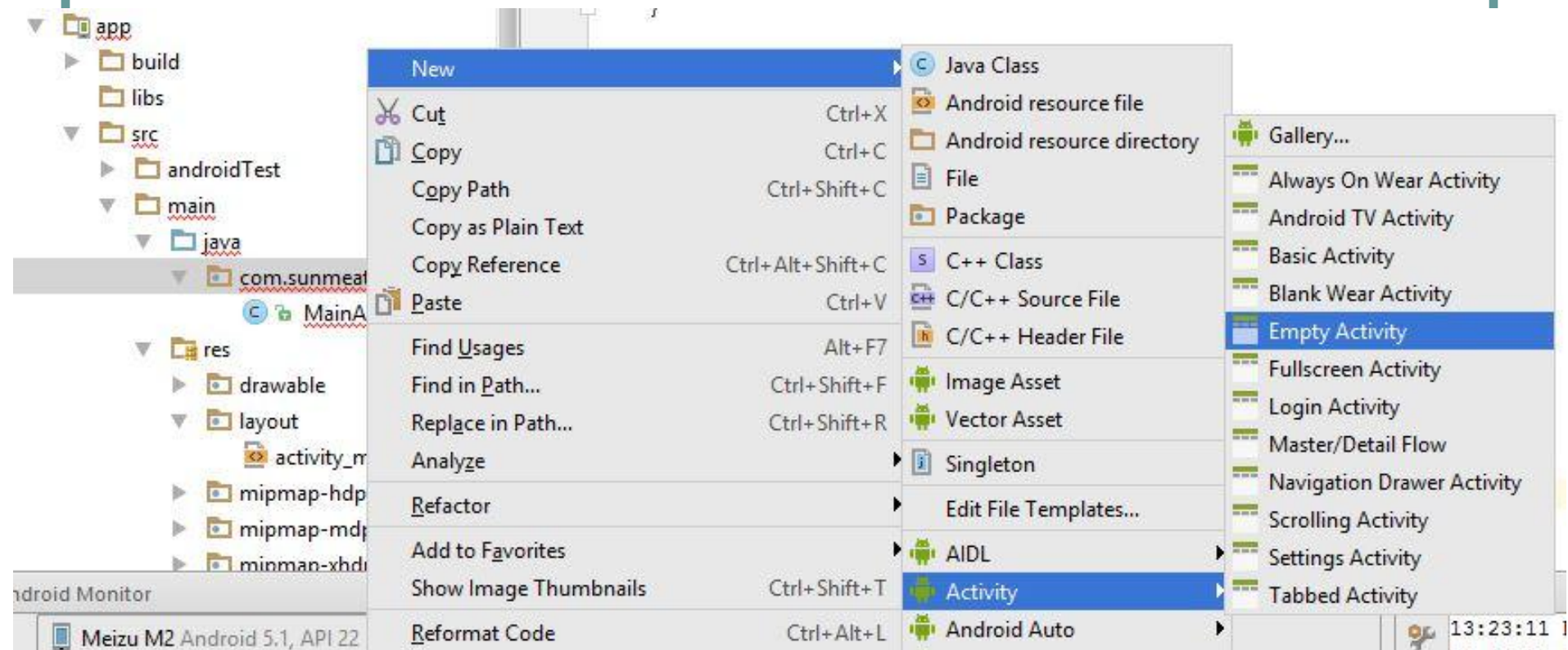
- Удалить все вызовы **setRequestedOrientation**
- Дописать в класс **MainActivity** два метода:

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    Log.i("main_activity", "on save instance state");
    outState.putInt("count", count);
}
```

```
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    Log.i("main_activity", "on restore instance state");
    count = savedInstanceState.getInt("count");
}
```

# Создаём второе активити

- Добавить новое пустое активити `SecondActivity` в проект на панели Project:



# Создаём второе активити

- Меняем тэг в MainActivity.java

```
String TAG = "FIRST_ACTIVITY";
```

- Обработчик кнопки первого активити:

```
@Override
```

```
public void onClick(View view) {  
    Intent intent = new Intent(this,  
        SecondActivity.class);  
    this.startActivity(intent);  
}
```



# Как это работает



# Что такое Intent

**Intent** (намерение) – это класс, объекты которого используются для взаимодействия между отдельными частями Android-приложения (например, двумя активити), либо между различными приложениями системы: интенты позволяют отправлять или принимать данные от других активити и сервисов, а также дают возможность оповещать о том, что произошло определённое событие.

<https://developer.android.com/reference/android/content/Intent.html>



# Создаём второе активити

- В класс `SecondActivity` добавляем поле `String TAG = "SECOND_ACTIVITY";`
- Делаем такие же логи в методах `onCreate`, `onDestroy` и тд.
- Запускаем, анализируем происходящее:



Шаг 1. Запускаем приложение. Появилось MainActivity.

MainActivity

onCreate

onStart

onResume

onPause



Шаг 2. Жмем кнопку «Go to Activity Two» на экране - появляется ActivityTwo. MainActivity остается в памяти.

ActivityTwo

onCreate

onStart

onResume

onStop

onPause

Шаг 3. Жмем кнопку Назад (Back) на эмуляторе. Мы вернулись в MainActivity. ActivityTwo уничтожено.

onRestart

onStart

onResume

onStop

onDestroy

Шаг 4. Жмем еще раз Назад и наше приложение закрылось. MainActivity уничтожено.

onPause

onStop

onDestroy



# Обмен данными

- Передать данные во второе активити

```
@Override
public void onClick(View view) {
    Intent intent = new Intent(this, SecondActivity.class);
    int data = 10;
    intent.putExtra("smth", data + "");
    this.startActivity(intent);
}
```

- Получить данные во втором активити

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    String str = getIntent().getStringExtra("smth");
    getSupportActionBar().setTitle(str);
}
```

# Практика

- Сделать на втором активити кнопку для возврата на первое активити
- Первое активити передаёт значение во второе. Второе может вернуть это значение назад, но значение должно быть на 1 больше полученного
- Первое активити должно получать значение от второго активити, и может снова передать его второму активити, но тоже на 1 больше полученного
- Выводить в заголовок активити `this.hashCode() + ""`, и попытаться понять, что происходит...

Как отправить объект:

<http://stackoverflow.com/questions/2139134/how-to-send-an-object-from-one-android-activity-to-another-using-intents>



# Нормальный возврат

- Новый вызов второй активности

**@Override**

```
public void onClick(View view) {  
    Intent intent = new Intent(this,  
    SecondActivity.class); // явный и.  
    this.startActivityForResult(intent,  
    1);  
}
```

# Нормальный возврат

- Обработчик кнопки второго активити

**@Override**

```
public void onClick(View view) {  
    Intent intent = new Intent();  
    intent.putExtra("smth", "sometext");  
    setResult(500, intent);  
    finish();  
}
```

# Нормальный возврат

- Добавить метод в MainActivity

@Override

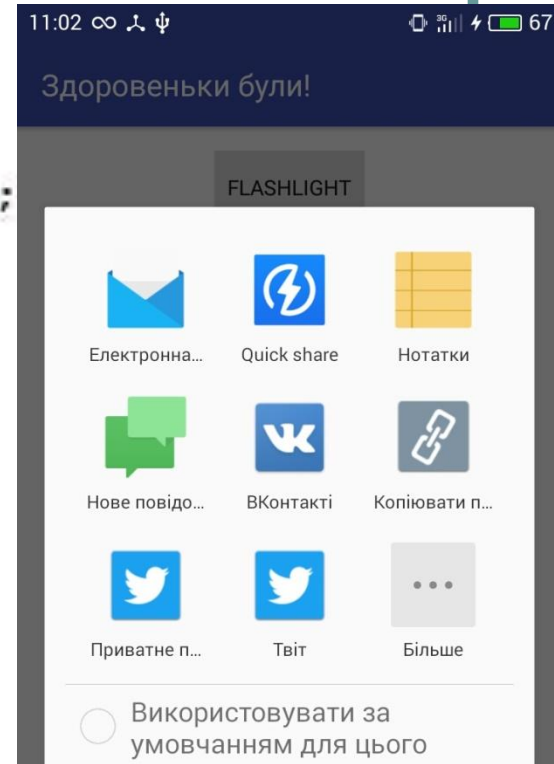
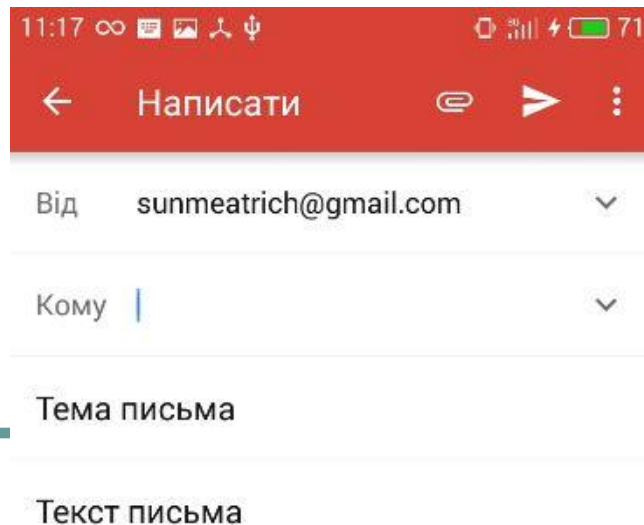
```
protected void onActivityResult(int requestCode, int  
resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    String str = data.getStringExtra("smth");  
    // getSupportActionBar().setTitle(requestCode + " " +  
resultCode + " " + str);  
    getSupportActionBar().setTitle(this.hashCode()+"");  
}
```



# Другое применение интенгов

Изменить код обработчика нажатия на кнопку в MainActivity:

```
@Override
public void onClick(View view) {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_SUBJECT, "Тема письма");
    intent.putExtra(Intent.EXTRA_TEXT, "Текст письма");
    this.startActivity(intent);
}
```



# Фильтр подходящих приложений

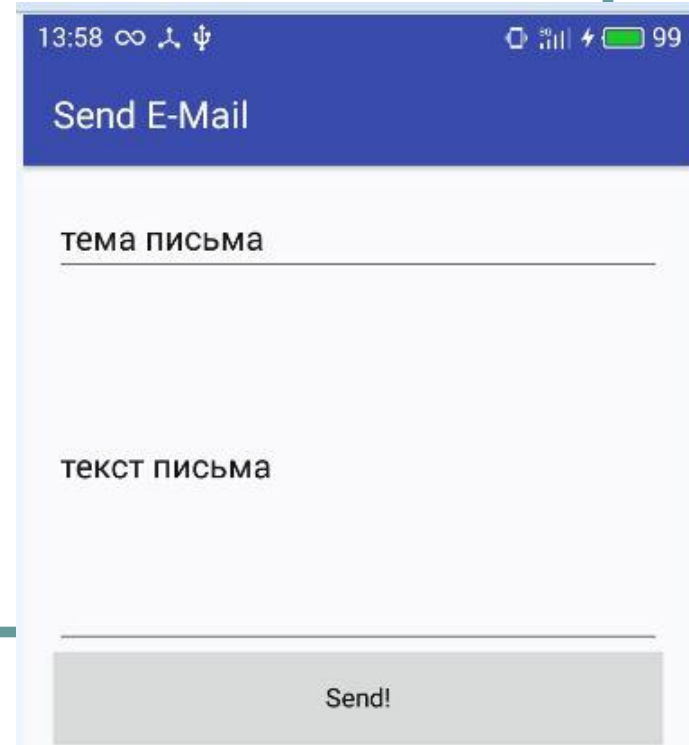
К сожалению, нет простого решения для показа лишь определённых приложений в стандартном диалоге выбора. В примере по ссылке ниже интендом будут вызваны все приложения, название пакета которых включает сочетание "**vk**". Как вариант, можно самостоятельно создать кастомный диалог, в котором будет отображенный список нужных приложений. О том, как создать элемент **ListView/GridView**, заполнить его информацией при помощи **BaseAdapter**, и разместить это всё в **DialogFragment**, вы узнаете в следующих занятиях.

<https://git.io/vij2b>

<http://stackoverflow.com/questions/6827407/how-to-customize-share-intent-in-android-29654#9229654>

# EditText

- XML-файл разметки MainActivity:  
<https://git.io/vifBR>
- Java-код класса MainActivity:  
<https://git.io/vifBg>



13:58 ∞ 人 ψ 99

Send E-Mail

тема писъма

текст писъма

Send!

# Flashlight v.2

Java-код класса MainActivity:

<https://git.io/vifaj>

(в XML нужна только одна кнопка)

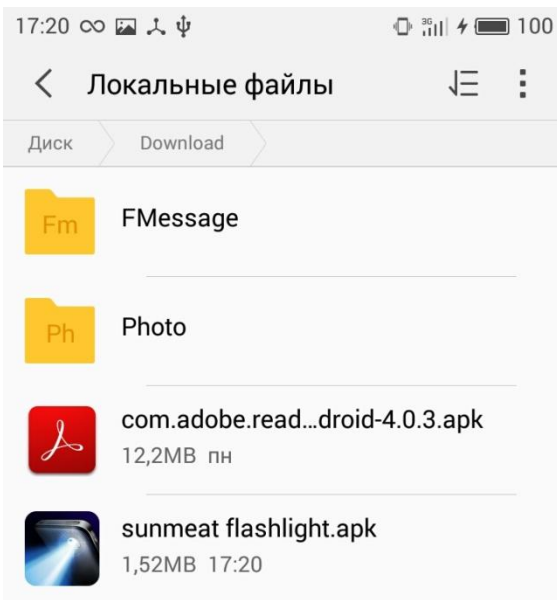
# Результаты



16:15 ∞ [signal] [wifi] [battery] 99  
LIGHT ON!

Включи

Выключить!



17:20 ∞ [signal] [wifi] [battery] 100



Flashlight

Версия: 1.0

Установить это приложение? Оно может получить следующие разрешения:

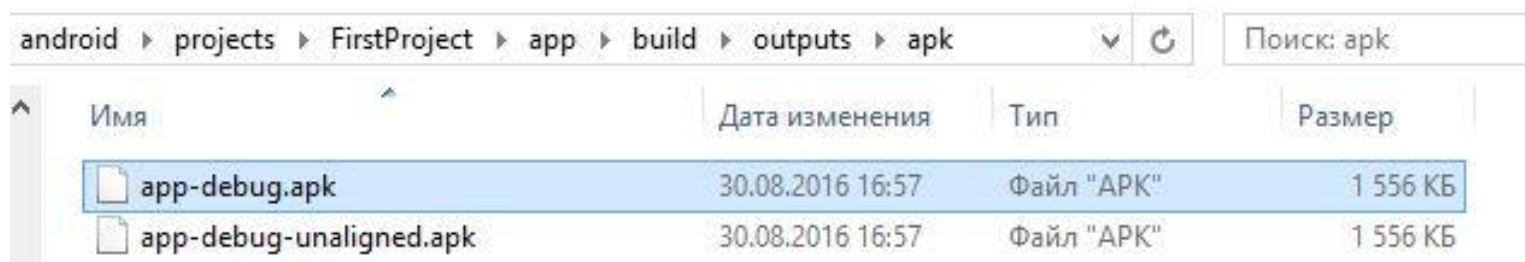
- Камера  
Фото- и видеосъёмка
- Батарея  
Управление вспышкой

Отмена

Установить

# Настройки проекта

- Где лежит арк-файл?



- Как сделать иконки?



A simple service to resize your app icon to all needed resolutions with rocket speed.

<https://resizeappicon.com/>



How it works. **As fast as possible.**

Just drag and drop the original image on the upload area or upload manually. The image is uploaded and the icons are generated automatically.

Activate the instant download option to directly download all icons.

We also provide **App Icon Management** and **App Stores for your Client**.

Best results are achieved with resolutions of 512x512 or 1024x1024 pixels.

Sizes (px)	Format and naming
Launcher icons	
48 × 48 (mdpi)	.png
72 × 72 (hdpi)	
96 × 96 (xxdpi)	
144 × 144 (xxxhdpi)	
192 × 192 (xxxhdpi)	
512 × 512 (Google Play store)	

# Настройки проекта

- Что делать с иконками?  
Онлайн-сервис подготовит все иконки нужных размеров.  
Останется только разложить их по подкаталогам папки **mirmap**!



GooglePlayStore.  
png



hdpi.png



ldpi.png



mdpi.png



xhdpi.png



xxhdpi.png



xxxhdpi.png



Android.

[All](#) | [High](#) | [Reset](#)

☒ 36x36 Pixel (LDPI)

☒ 48x48 Pixel (MDPI)

☒ 72x72 Pixel (HDPI)

☒ 96x96 Pixel (XHDPI)

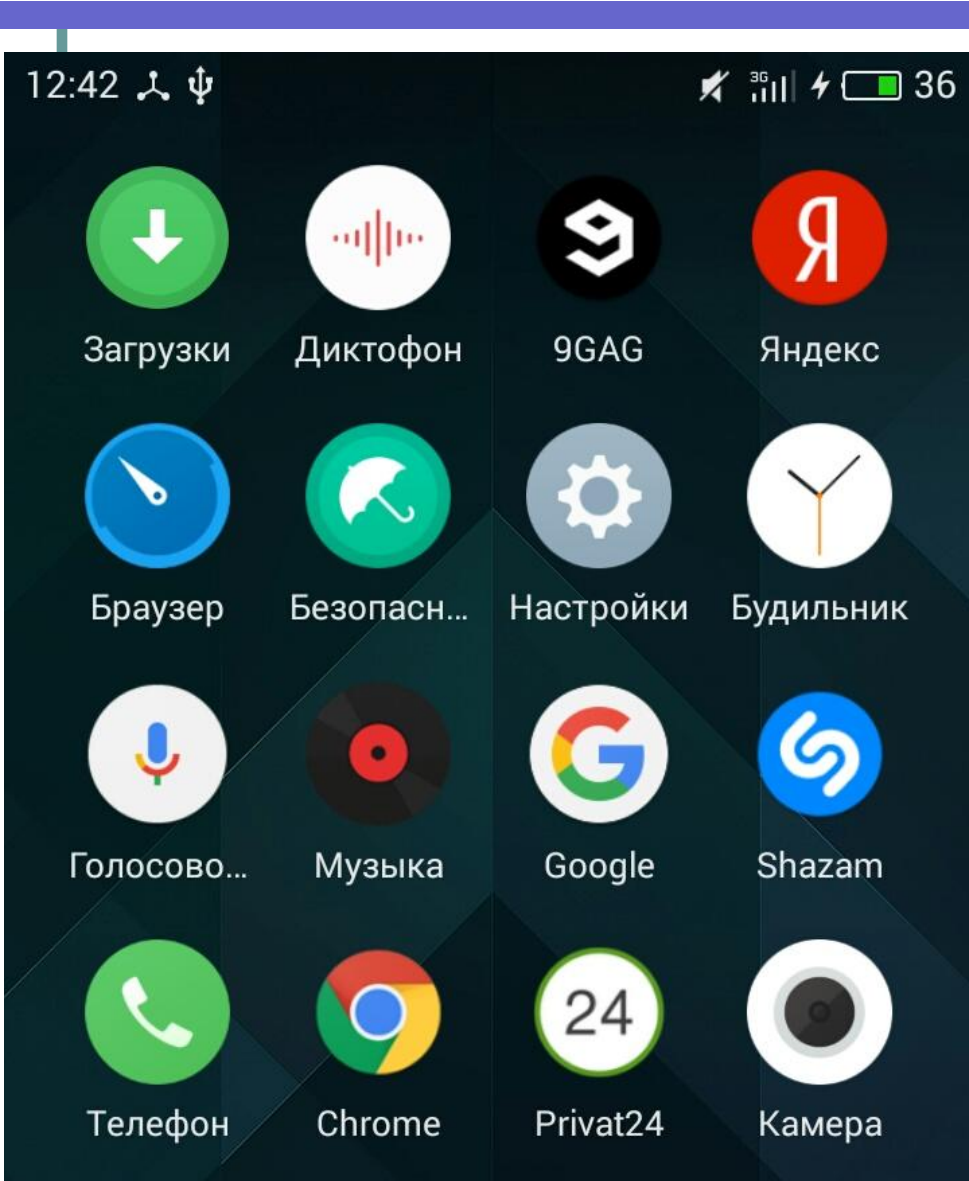
☒ 144x144 Pixel (XXHDPI)

☒ 192x192 Pixel (XXHDPI)

☒ 512x512 Pixel (Google Play Store)



# Гугл продвигает круглые иконки

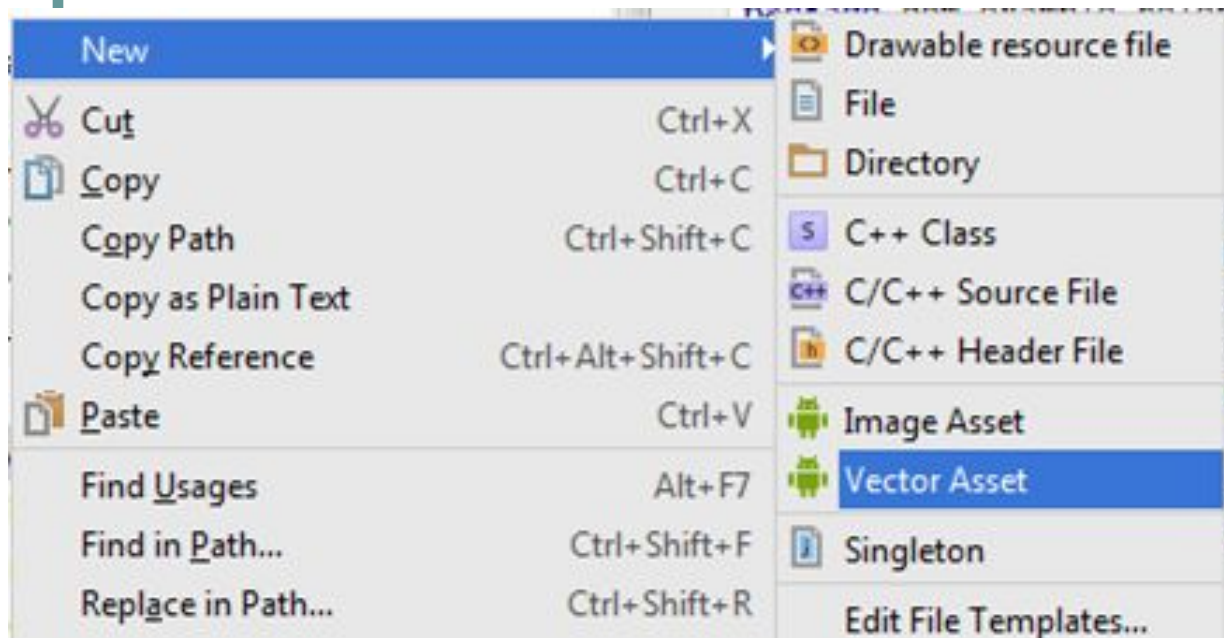


# Практика

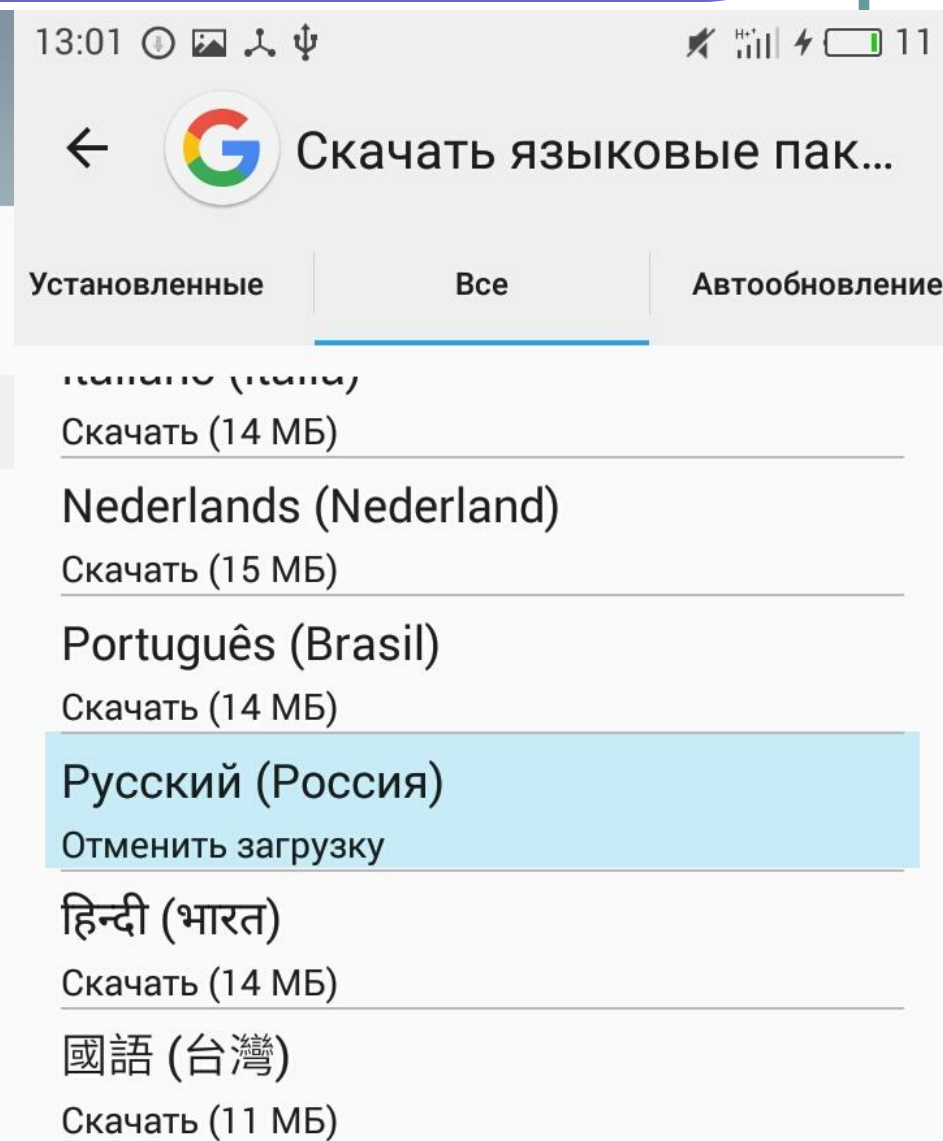
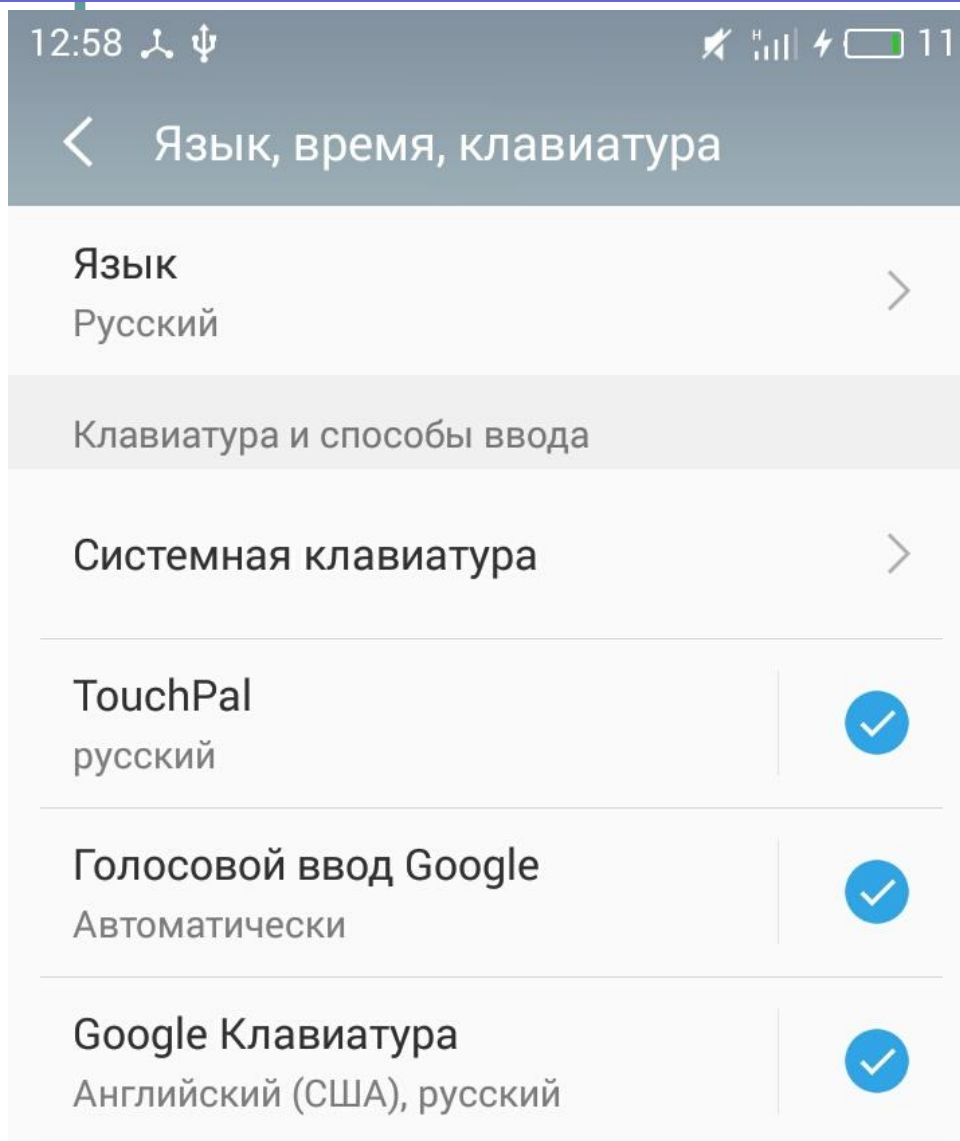
- Установить иконку для своего приложения.

# Подготовка изображений

- ПКМ по папке Drawable > New > Vector > Asset > Choose...




# Голосовой ввод текста



# ГОЛОСОВОЙ ВВОД

- strings.xml:
- <https://git.io/viSfV>
- colors.xml:
- <https://git.io/viSfi>
- activity\_main.xml:
- <https://git.io/viSf5>
- MainActivity.java:
- <https://git.io/viSfb>

## Speech To Text

йога ноги на угле  
зимой на ноги где мы  
танцуем буги вуги под  
люли люли дикий  
маугли джунгли самсы  
google в гугле матрешки  
куклы в кукле это день w  
w разделе лесной  
команды пока модули w  
да с виду л 

Нажми на кнопку!

# Домашнее задание

- Передать на второе активити через интент объект типа **Cat** или **Student**. Вытащить из переданного объекта основные данные, показать их в заголовке активити, либо тостом.
- Реализовать этап регистрации в приложении (полный текст задания в вордовском файле).