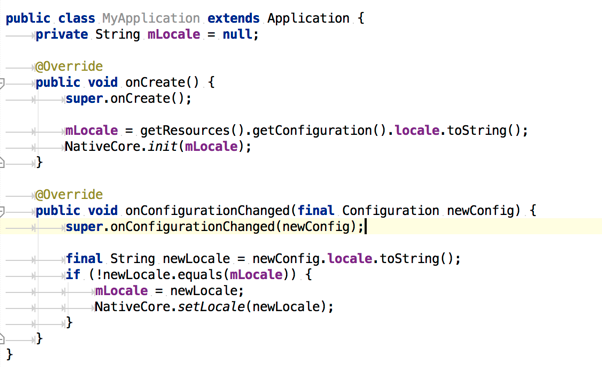
# 2. Android Components, Потоки, Resources

## Application

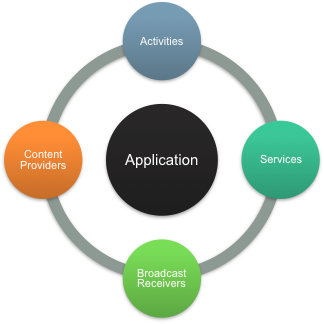
В языке Java выполняющееся приложение представляет объект android.app.Application, экземпляр которого создается при запуске и уничтожается при завершении приложения (то есть время существования экземпляра класса Application

совпадает со временем существования соответствующего процесса Linux. Когда процесс завершается и перезапускается, создается новый экземпляр класса Application.



## Компоненты Android

Основными составными частями любого Апdгоid приложения являются компоненты, управляемые средой выполнения:**Activity, Service, BroadcastReceiver и ContentProvider**.



## Intent

Intent – сущность для описания операции, которую требуется выполнить

Используется для:

* Запуска Activity

Intent intent = new Intent(context, DisplayMessageActivity.class);

startActivity(intent);

* Запуска сервиса - startService(new Intent(this, MyService.class));
* Отправки широковещательных сообщений -

Intent intent = new Intent();

intent.setAction("com.example.broadcast.MY\_NOTIFICATION");

intent.putExtra("data","Notice me senpai!");

sendBroadcast(intent);

* Выполнения стандартных, предопределенных операций

Intent browserIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse("http://www.google.com"));

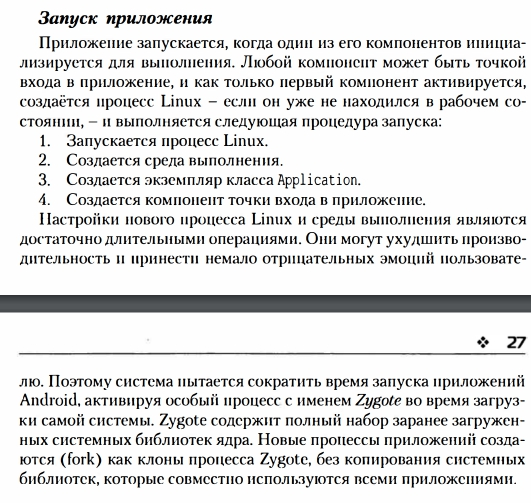
startActivity(browserIntent);

## Механизм запуска приложения, приоритет жизни процессов

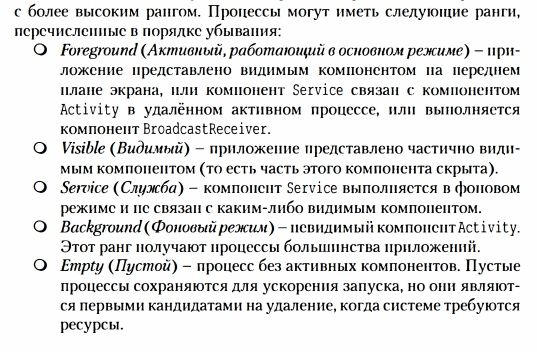
В ОС Android пакет каждого приложения имеет не по­вторяющийся в системе идентификатор пользователя, то есть при­ложение в Android соответствует отдельному пользователю в Linux и не имеет доступа к ресурсам других приложений.



По умолчанию, между приложением и процессом устанавливается отношение Один к одному, но при необходимости приложение можно запустить в нескольких процессах или организовать выполнение нескольких приложений в одном процессе.



**Завершение приложений -** приложения убиваются системой, разработчик не должен этим управлять



## Потоки

С точки зрения приложения существуют три типа потоков: UI поток, (поток пользовательского интерфейса), связующие потоки и фоновые потоки.

Под потоком будем понимать некую последовательность команд.

**UI-поток** создается при запуске приложения и продолжает существовать на протяжении всего жизненного цикла соответствующего процесса Linux. Ul-поток является главным потоком приложения, используемым для выполнения компонентов Android и обновления элементов пользовательского интерфейса на экране. Если платформа обнаружит попытки обновления пользовательского интерфейса из какого-либо другого потока, она тут же оповестит об этом приложе­ние посредством исключения CalledFromWrongThreadException.

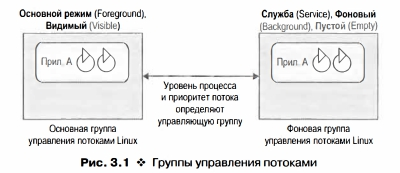
Важно не загружать поток длительными операциями, это напрямую влияет на восприятие приложения пользователем.

Рассказать про бд, сеть, чтение файлов.

**Связующие потоки** используются для взаимодействия между потоками, работающими в разных процессах.   
  
**Фоновые потоки** - все потоки, которые приложение создает явно, являются фоновыми потоками (background threads). Они не имеют предопределенного на­значения, и каждый представляет всего лишь пустую среду выполнения, ожидающую выполнения любой задачи. Фоновые потоки - это объекты, производные от UI-потока, поэтому они наследуют все его свойства, например приоритет.

Пото­ки принадлежат тому процессу, в котором они были созданы, то есть процесс является родителем каждого своего потока. Потоки и процессы во многом схожи и отличаются лишь подходом к совместному использованию ресурсов. Процесс обеспечивает изолированное выполнение программы в среде, защищенной от других процессов, тогда как потоки совместно используют ресурсы внутри своего процесса. Наиболее важное отличие процессов от потоков состоит в том, что процессы не имеют общего адресного пространства, а потоки, напро­тив, пользуются общей памятью внутри процесса. Такое совместное использование памяти обеспечивает более высокую скорость обмена данными между потоками, по сравнению с процессами, требующими выполнения вызовов удаленных процедур, которые увеличивают накладные расходы.

Когда процесс начинает работу, для него автоматически создается один поток - ui thread.

Потоки конкурируют между собой за процессорное время на основании приоритетов. Видимые потоки имеют больший приоритет 

## Единицы измерения

**dp** или **dip** - Density-independent Pixels. Абстрактная ЕИ, позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях.

**sp** - Scale-independent Pixels. То же, что и dp, только используется для размеров шрифта в View элементах

**pt** - 1/72 дюйма, определяется по физическому размеру экрана. Эта ЕИ [из типографии](http://en.wikipedia.org/wiki/Point_(typography)).

**px** – пиксел, не рекомендуется использовать т.к. на разных экранах приложение будет выглядеть по-разному.

**mm** – миллиметр, определяется по физическому размеру экрана

**in** – дюйм, определяется по физическому размеру экрана

Используем в основном dp, sp для размера текста. Остальное - в крайних случаях.

Основные размеры, рассказать про скетч и zeplin, векторная графика.

Основной дизайн идет в **360x 640**

* ldpi = -> 0.75
* mdpi = -> 1.0 - 360x 640 dp, 1px = 1dp
* hdpi = -> 1.5
* xhdpi = -> 2.0
* xxhdpi = -> 3.0
* xxxhdpi = -> 4.0

## Типы ресурсов

<https://developer.android.com/guide/topics/resources/providing-resources.html>

animator - анимация свойств

anim - анимация преобразований (fade in)

color - список состояний цветов

drawable

mipmap

layout

menu

raw

values

xml

## Предоставление альтернативных ресурсов

<https://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources> таблица 2

|  |  |
| --- | --- |
| Язык и регион | Примеры:  en  fr  en-rUS |

|  |  |
| --- | --- |
| smallestWidth | sw<N>dp  Примеры:  sw320dp  sw600dp  sw720dp  и т. д. |

|  |  |
| --- | --- |
| Ориентация экрана | port  land |

|  |  |
| --- | --- |
| Плотность пикселов на экране (dpi) | ldpi  mdpi  hdpi  xhdpi  xxhdpi  xxxhdpi  nodpi  tvdpi |

|  |  |
| --- | --- |
| Версия платформы (уровень API) | Примеры:  v3  v4  v7  и т. д. |

Кратко затронуть локализацию.

## Доступ к ресурсам

Во время компиляции приложения создается класс R, в котором находятся идентификаторы для всех ресурсов в каталоге res/. Для каждого типа ресурсов предусмотрен подкласс R (например, класс R.drawable для элементов дизайна), а для каждого ресурса указанного типа существует статическая целочисленная переменная (например, R.drawable.icon).

Существует два способа доступа к ресурсу.

* **Из кода:** с помощью статической целочисленной переменной из подкласса вашего класса R , например:
* R.string.hello

ImageView imageView = (ImageView) findViewById(R.id.myimageview);

imageView.setImageResource(**R.drawable.myimage**);

* **Из XML:** с помощью особого синтаксиса XML, который также соответствует идентификатору ресурса, заданному в классе R, например:
* @string/hello

<Button

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="**@string/submit**" />

Рассказать про ссылки на атрибуты стиля -

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?android:attr/actionBarSize"/>

Рассказать про наличие стандартных лейаутов

[setListAdapter](https://developer.android.com/reference/android/app/ListActivity.html#setListAdapter(android.widget.ListAdapter))(new [ArrayAdapter](https://developer.android.com/reference/android/widget/ArrayAdapter.html)<String>(this, **android.R.layout.simple\_list\_item\_1**, myarray));

## Drawables

Вкратце рассказать про пнг и svg.

<https://i.stack.imgur.com/1xUPY.png> - объяснить диаграмму.

* **Shape Drawable**

<?xml version="1.0" encoding="utf-8"?>

<[shape](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#shape-element)

xmlns:android="http://schemas.android.com/apk/res/android"

android:shape=["rectangle" | "oval" | "line" | "ring"] >

<[corners](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#corners-element)

android:radius="integer"

android:topLeftRadius="integer"

android:topRightRadius="integer"

android:bottomLeftRadius="integer"

android:bottomRightRadius="integer" />

<[gradient](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#gradient-element)

android:angle="integer"

android:centerX="float"

android:centerY="float"

android:centerColor="integer"

android:endColor="color"

android:gradientRadius="integer"

android:startColor="color"

android:type=["linear" | "radial" | "sweep"]

android:useLevel=["true" | "false"] />

<[padding](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#padding-element)

android:left="integer"

android:top="integer"

android:right="integer"

android:bottom="integer" />

<[size](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#size-element)

android:width="integer"

android:height="integer" />

<[solid](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#solid-element)

android:color="color" />

<[stroke](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#stroke-element)

android:width="integer"

android:color="color"

android:dashWidth="integer"

android:dashGap="integer" />

</shape>

* Layer List

<?xml version="1.0" encoding="utf-8"?>

<[layer-list](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#layerlist-element)

xmlns:android="http://schemas.android.com/apk/res/android" >

<[item](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#layerlist-item-element)

android:drawable="@[package:]drawable/drawable\_resource"

android:id="@[+][package:]id/resource\_name"

android:top="dimension"

android:right="dimension"

android:bottom="dimension"

android:left="dimension" />

</layer-list>

* State List

<?xml version="1.0" encoding="utf-8"?>

<[selector](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#selector-element) xmlns:android="http://schemas.android.com/apk/res/android"

android:constantSize=["true" | "false"]

android:dither=["true" | "false"]

android:variablePadding=["true" | "false"] >

<[item](https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru#item-element)

android:drawable="@[package:]drawable/drawable\_resource"

android:state\_pressed=["true" | "false"]

android:state\_focused=["true" | "false"]

android:state\_hovered=["true" | "false"]

android:state\_selected=["true" | "false"]

android:state\_checkable=["true" | "false"]

android:state\_checked=["true" | "false"]

android:state\_enabled=["true" | "false"]

android:state\_activated=["true" | "false"]

android:state\_window\_focused=["true" | "false"] />

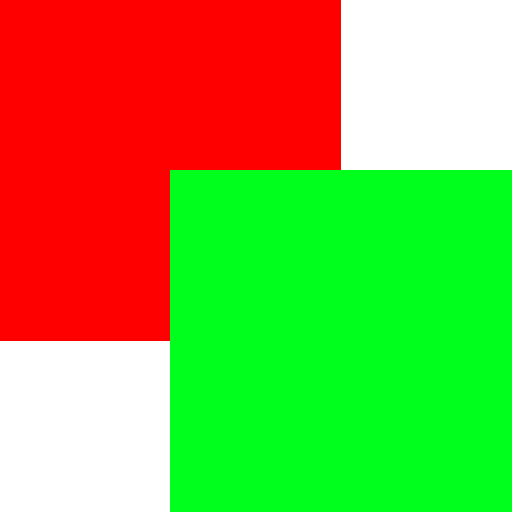
</selector>

#### Полезные ссылки:

1. типы ресурсов - <https://developer.android.com/guide/topics/resources/providing-resources.html>
2. обращение к ресурсам - <https://developer.android.com/guide/topics/resources/accessing-resources.html>
3. Изучить самостоятельно - сохранение <https://developer.android.com/guide/topics/resources/runtime-changes.html>
4. Изучить самостоятельно - drawables <https://developer.android.com/guide/topics/resources/drawable-resource.html?hl=ru>
5. Handler <http://startandroid.ru/ru/uroki/vse-uroki-spiskom/143-urok-80-handler-nemnogo-teorii-nagljadnyj-primer-ispolzovanija.html>

#### Задание:

Каждый пункт выполнять на отдельно взятом активити. Должно быть:

1. Активити с тестом. В текст вставить любое четверостишие Пушкина.  
   Необходимо, чтобы на русской локали устройства (values-ru) он отображался, а на всех остальных было написано - “Sorry, there is nothing”.  
   Необходимо, чтобы на устройствах, ширина которых больше 325 dp (values-sw325dp) текст был черный, а на остальных - красный.  
   Необходимо, чтобы в горизонтальном режиме телефона на экране находилась такая картинка <https://upload.wikimedia.org/wikipedia/commons/4/45/Pushkin_Signature.svg> , а в вертикальном такая <https://upload.wikimedia.org/wikipedia/commons/b/b1/Saint_Andrew%27s_cross_black.svg> (drawables-land)  
   Для импорта svg - правой на res, new - vector assets  
   На данном активити должна быть кнопка, надпись на которой меняется в зависимости от версии Android (values v21): на Androd 5.0 и выше должно быть написано - “Радость и счастье”, а ниже 5.0 - “Боль и страдание”.
2. Активити со следующими элементами:   
   Реализовать следующую xml drawable (использовать layer list, элементами которого будут shape android:shape="rectangle")  
     
   a) Ширина всего полотна = высоте = 100dp. Сторона каждого из квадратов = 60 dp.  
     
   Установить данный xml в качестве фона ImageView. Посмотреть в редакторе, как эта картинка выглядит на планшетах и на телефонах  
     
   b) Сделать кнопку (Button), к которой обычное состояние будет такое: (использовать State List из shape drawable)  
     
   а нажатое такое:  
     
   Текст на кнопке - “Кнопка 1”  
   Скругление - 7dp, цвет - #d46610, кнопка располагается под элементом из первого пункта.  
   c) Взять картинку   
   <https://habrastorage.org/storage/habraeffect/ca/8f/ca8f5771374b2ca12de20f01832c1b4e.png>   
   Сделать из нее 9-patсh картинку. Установить фоном активити из этого задания. Убедиться в правильности пропорций