



УНИВЕРСИТЕТ ИТМО

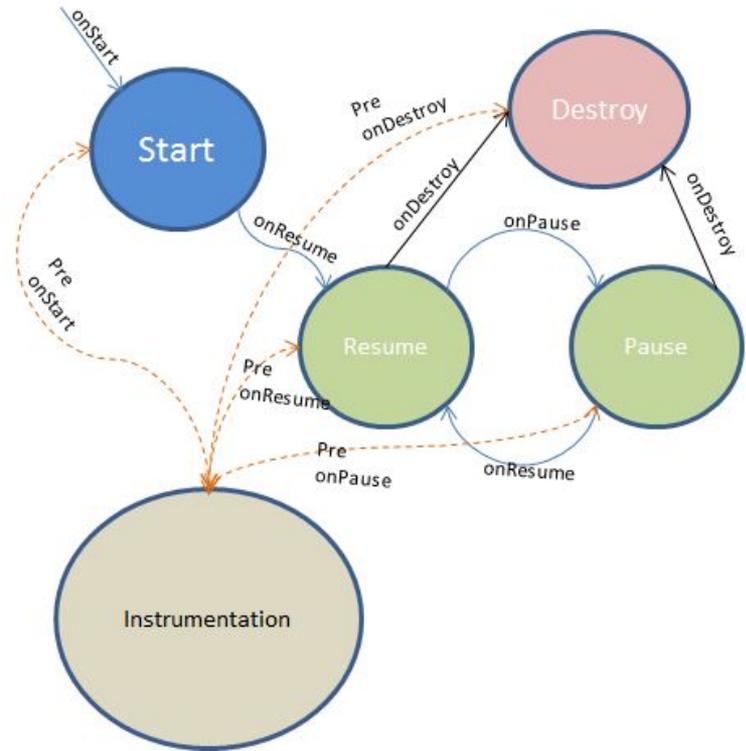
# Управление жизненным циклом операции

Санкт-Петербург, 2016

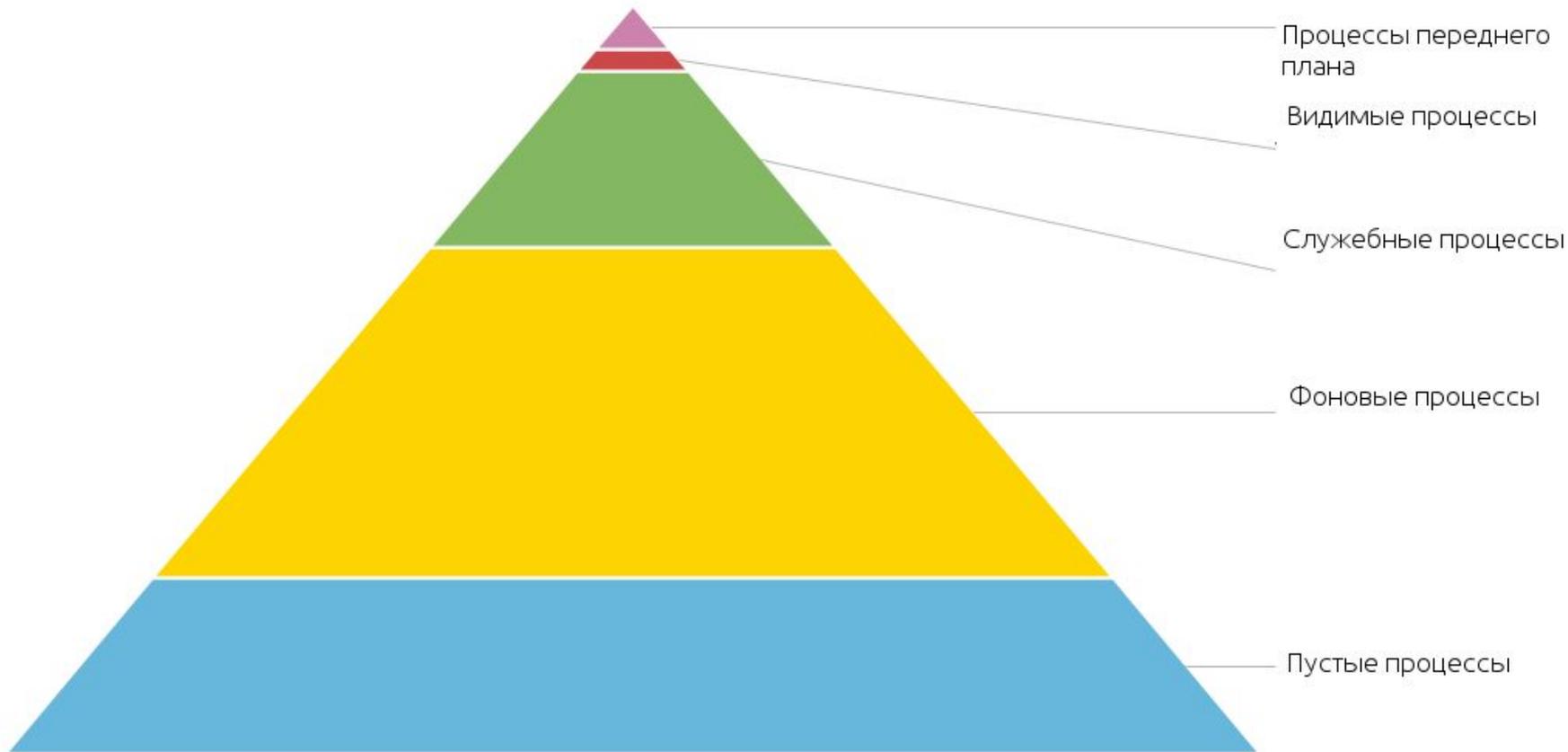
# Жизненный цикл приложения

- жизненный цикл приложения в Android жёстко контролируется системой и зависит от нужд пользователя, доступных ресурсов и т. д.
- Android всегда уничтожает процессы, а не компоненты.

Во время навигации пользователя по вашему приложению, экземпляры [Activity](#) внутри приложения переключаются между разными состояниями их жизненного цикла



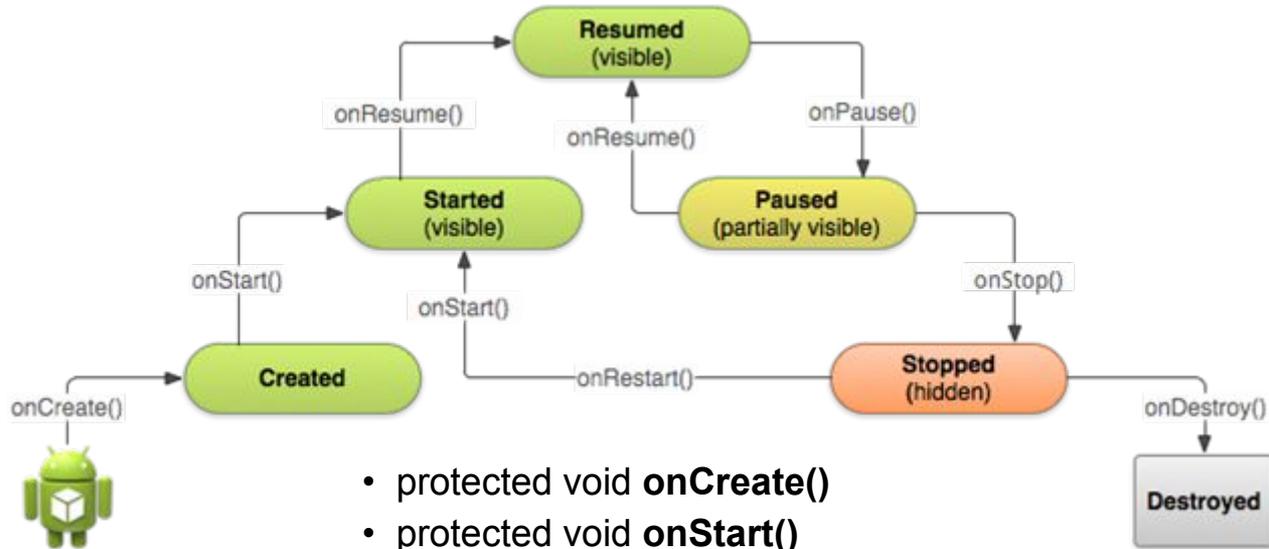
# Кому жить, а кому умереть: приоритеты процессов в Android



Правильная реализация методов жизненного цикла ([Activity](#) обеспечивает нормальную работу приложения):

- Не прекращает работу, если пользователь получает телефонный звонок или переключается на другое приложение во время использования вашего приложения.
- Не потребляет ценные системные ресурсы, когда пользователь не использует его активно.
- Сохраняет состояние приложения, если пользователь выходит из него и возвращается позднее.
- Не закрывается с ошибкой и не теряет данные пользователя при повороте экрана (смене ориентации между книжной и альбомной).

# Основные методы жизненного цикла



- protected void **onCreate()**
- protected void **onStart()**
- protected void **onRestart()**
- protected void **onResume()**
- protected void **onPause()**
- protected void **onStop()**
- protected void **onDestroy()**

# onCreate()

Метод **onCreate()** вызывается при создании или перезапуске активности.

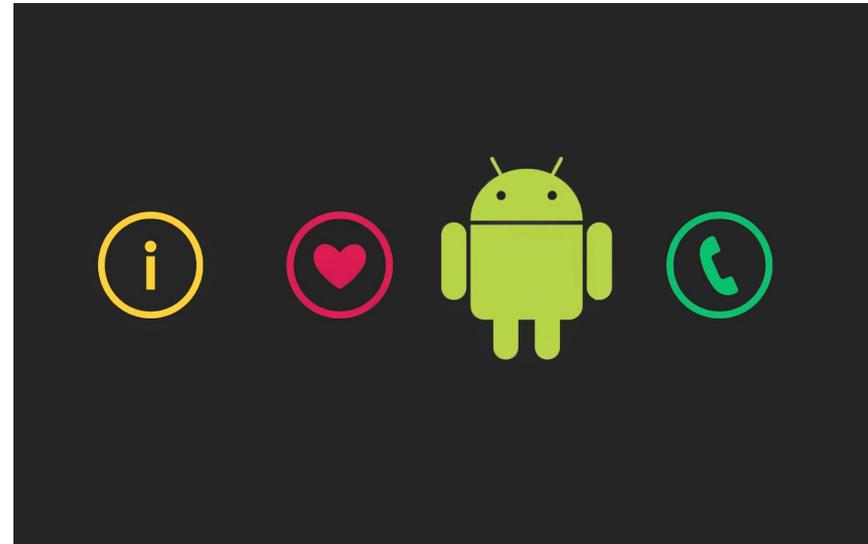
Система может запускать и останавливать текущие окна в зависимости от происходящих событий.

Внутри данного метода настраивают статический интерфейс активности.

Инициализирует статические данные активности, связывают данные со списками и т.д.

Связывает с необходимыми данными и ресурсами.

Задаёт внешний вид через метод **setContentview()**.



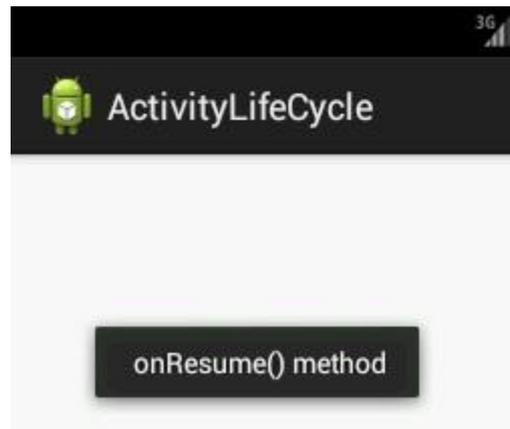
# onStart()

При вызове **onStart()** окно еще не видно пользователю, но вскоре будет видно. Вызывается непосредственно перед тем, как активность становится видимой пользователю.



# onResume()

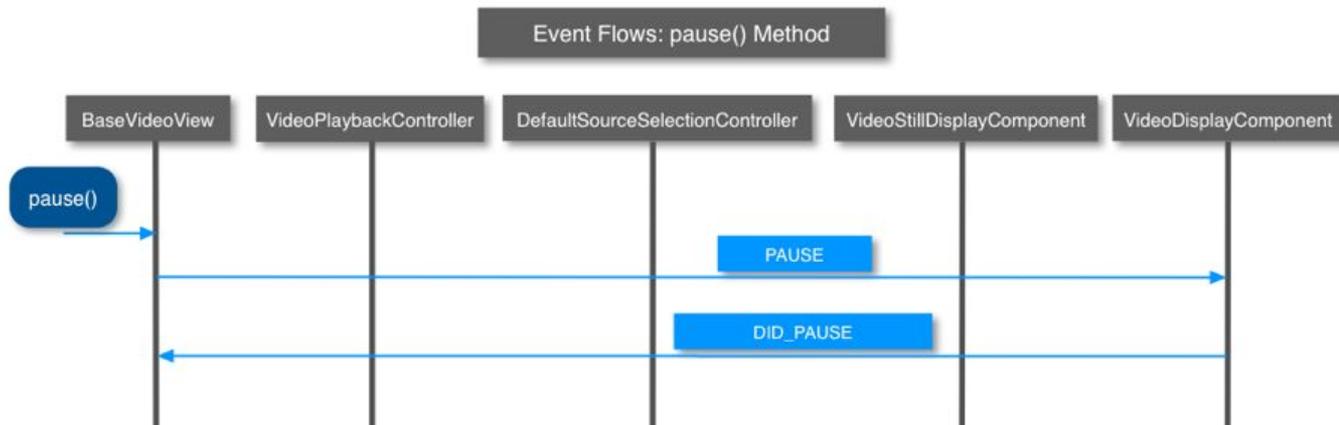
Метод **onResume()** вызывается после **onStart()**, даже когда окно работает в приоритетном режиме и пользователь может его наблюдать. В этот момент пользователь взаимодействует с созданным вами окном. Приложение получает монопольные ресурсы. Запускает воспроизведение анимации, аудио и видео. Также может вызываться после **onPause()**.



# onPause()

Когда пользователь решает перейти к работе с новым окном, система вызовет для прерываемого окна метод **onPause()**. По сути происходит свёртывание активности.

Сохраняет незафиксированные данные. Деактивирует и выпускает монопольные ресурсы. Останавливает воспроизведение видео, аудио и анимацию. От **onPause()** можно перейти к вызову либо **onResume()**, либо **onStop()**.



# onStop()

Метод **onStop()** вызывается, когда окно становится невидимым для пользователя. Это может произойти при её уничтожении, или если была запущена другая активность (существующая или новая), перекрывшая окно текущей активности. Всегда сопровождает любой вызов метода **onRestart()**, если активность возвращается, чтобы взаимодействовать с пользователем, или метода **onDestroy()**, если эта активность уничтожается.



# onRestart()

Если окно возвращается в приоритетный режим после вызова **onStop()**, то в этом случае вызывается метод **onRestart()**. Т.е. вызывается после того, как активность была остановлена и снова была запущена пользователем. Всегда сопровождается вызовом метода **onStart()**.



# onDestroy()

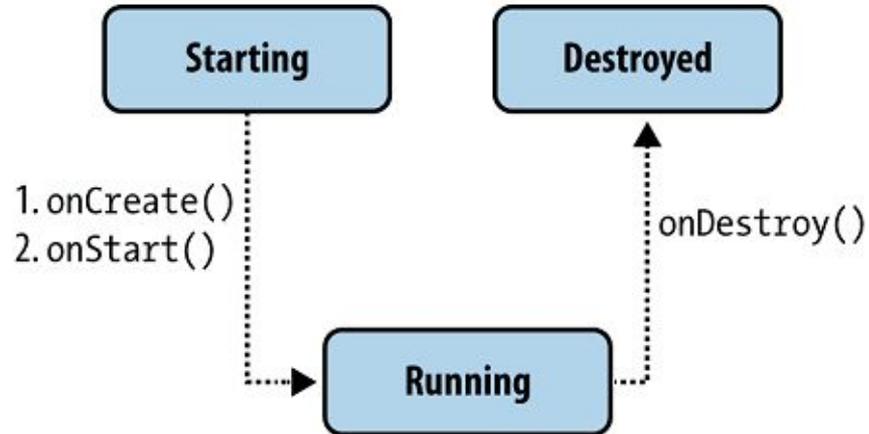
Метод вызывается по окончании работы активности, при вызове метода **finish()** или в случае, когда система уничтожает этот экземпляр активности для освобождения ресурсов. Эти два сценария уничтожения можно определить вызовом метода **isFinishing()**. Вызывается перед уничтожением активности.



# Пересоздание активности

## Android пересоздает Activity:

- При изменении конфигурации устройства, например когда
  - изменяется ориентация экрана
  - пользователь меняет язык системы в настройках Android и т.п.
- При возврате пользователя к процессу, который был убит Android для освобождения ресурсов.



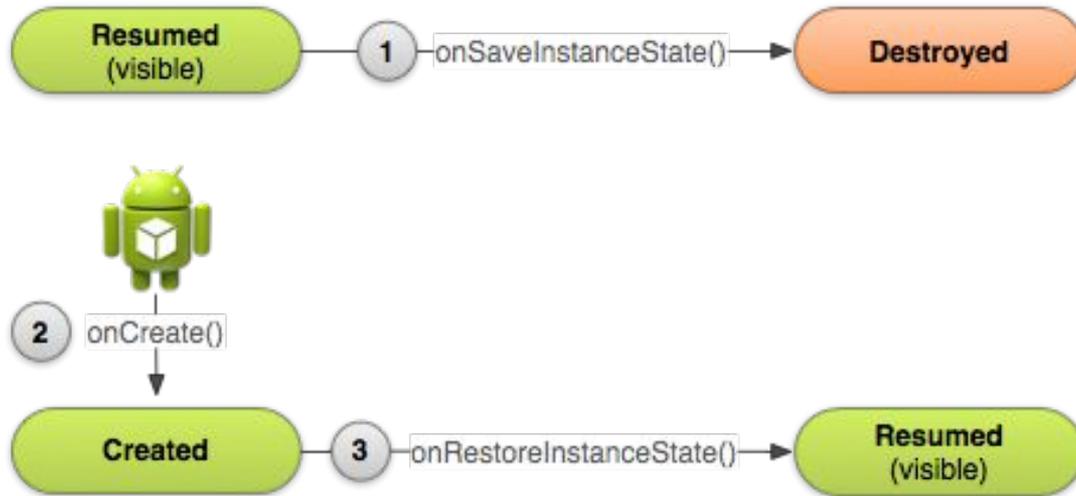
# Параметр configChanges

android:configChanges="orientation|screenSize"

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT){
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }
}
```

# Сохранение состояния при пересоздании Activity



Когда система начинает останавливать операцию, она вызывает [onSaveInstanceState\(\)](#) Когда система начинает останавливать операцию, она вызывает `onSaveInstanceState()`, чтобы вы могли указать дополнительные данные состояния, которые нужно сохранить на случай необходимости воссоздания экземпляра [Activity](#).

# Материалы и ссылки

<https://habrahabr.ru/post/276381/>

<http://developer.alexanderklimov.ru/android/theory/lifecycle.php>

<https://developer.android.com/training/basics/activity-lifecycle/index.html>

<https://developer.android.com/training/basics/activity-lifecycle/starting.html>

<https://developer.android.com/training/basics/activity-lifecycle/pausing.html>

<https://developer.android.com/training/basics/activity-lifecycle/stopping.html>

<https://developer.android.com/training/basics/activity-lifecycle/recreating.html>