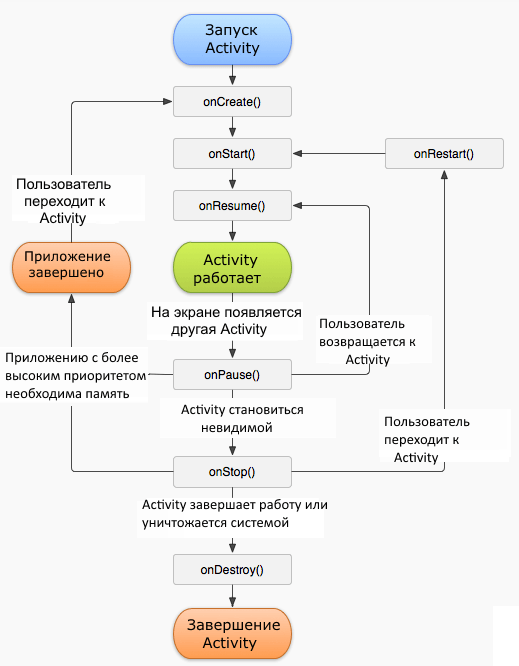
**Лабораторная работа № 3**

**Жизненный цикл Activity в Android-приложениях. Intent**

**Общие сведения.**

После запуска приложения его activity могут находиться в одном из состояний Активное (Resumed), Приостановленное (Paused), Остановленное (Stopped). При переходах из состояния в состояние вызываются методы жизненного цикла: **onCreate, onStart, onResume, onPause, onStop, onDestroy.**



Эти методы используются для управления работой приложения при сменах состояний – сохранения данных, регистрации компонент, высвобождения ресурсов.

Смена состояния activity может происходить как из-за действий пользователя, так и из-за действий самого приложения. Например, из-за переключения пользователя с одного приложения на другое, или если внутри приложения открывается новая activity.

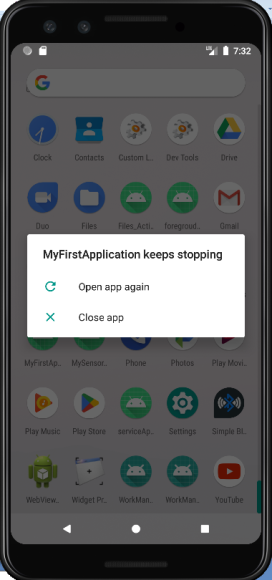
Наиболее распространенный способ наладить взаимодействие между Android приложениями или между activity в рамках одного приложения – использование класса **android.content.Intent.**

Объект этого класса необходимо в классе activity из которого будет осуществляться вызов другой activity:

Intent openIntent = **new** Intent(**this**, SecondActivity.**class**);

При создании явного intent в конструктор класса передается текущий контекст и название класса, в который будет передан intent.

Открываемая таким образом activity должна быть прописана в манифесте, иначе ее запуск вызовет сбой работы приложения.



После создания intent в нем может быть указана информация, которую необходимо передать в запускаемую activity:

openIntent.putExtra(**"hello"**, **"Hello from FirstActivity"**);

Запуск activity осуществляется путем передачи созданного intent в метод

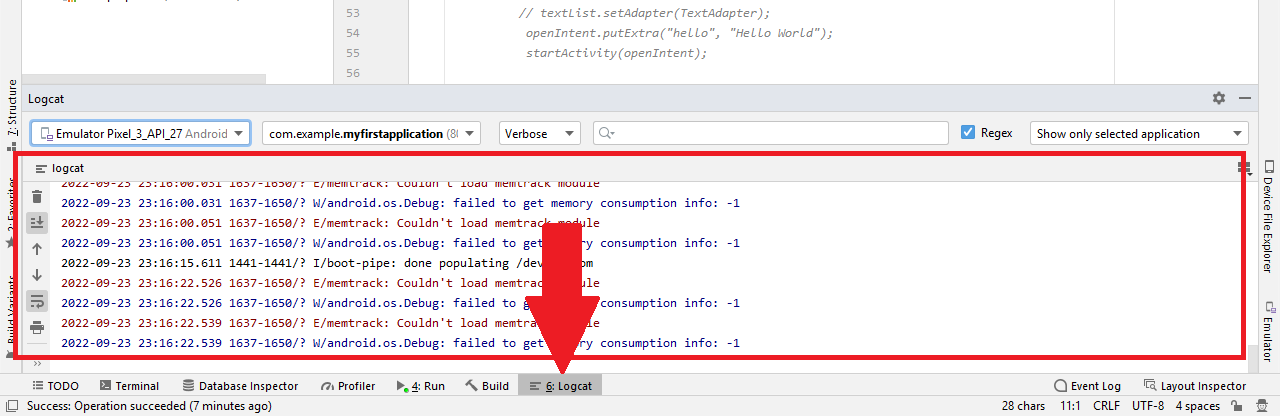
startActivity(openIntent);

В запускаемой activity передаваемая через intent информация доступна для извлечения и дальнейшего использования:

Bundle arguments = getIntent().getExtras();  
String str = arguments.get(**"hello"**).toString();

Отследить условия, при которых происходит смена состояния жизненного цикла приложения, можно в системном журнале **android.util.Log** или использовать всплывающие сообщения **android.widget.Toast.**

Сообщения системного журнала выводятся в консоль AndroidStudio.



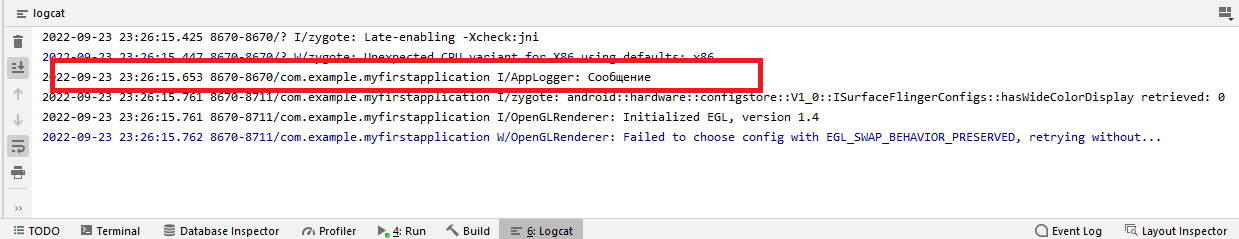
Сообщения разбиваются по категориям и для отображения каждой категории существуют собственные методы:

* **Log.e()** - ошибки (error)
* **Log.w()** - предупреждения (warning)
* **Log.i()** - информация (info)
* **Log.d()** - отладка (degub)
* **Log.v()** - подробности (verbose).

Пример вызова метода:

Log.*i*(**"AppLogger"**, **"Сообщение"**);

И результат:



Для создания всплывающего сообщения необходимо создать экземпляр класса:

Toast mytoast = **new** Toast(**this**);

Затем вызвать метод *makeText,*в который передать контекст текущей activity, само сообщение и параметр длительности его отображения на экране, после чего вызвать метод show.

Toast.*makeText*(HelloActivity.**this**,  
 **"Message"**, Toast.***LENGTH\_SHORT***).show();

**Ход выполнения проекта.**

1 Откройте проект, созданный в лабораторной работе № 2. Он должен содержать два класса activity. Главная activity предствляет собой форму авторизации. Вторая – форму с ListView для отображения списка данных и элементами пользовательского интерфейса для добавления и удаления данных.

2 Измените код в соответствии с заданием.

**Задание**:

1 Переопределить методы жизненного цикла первой activity. Например,

@Override  
**protected void** onStart() {  
 **super**.onStart();  
  
}

2 Добавить в переопределенные методы жизненного цикла код для вывода сообщений на экран или в системный журнал.

3 Создать intent для вызова второй activity.

4 Проанализировать порядок вызова методов жизненного цикла на основе выводимых сообщений при различных действиях пользователя.

5 Передать данные из одной activity в другую при помощи intent и продемонстрировать результат.