



# Виды приложений и их структура

Лекция 6

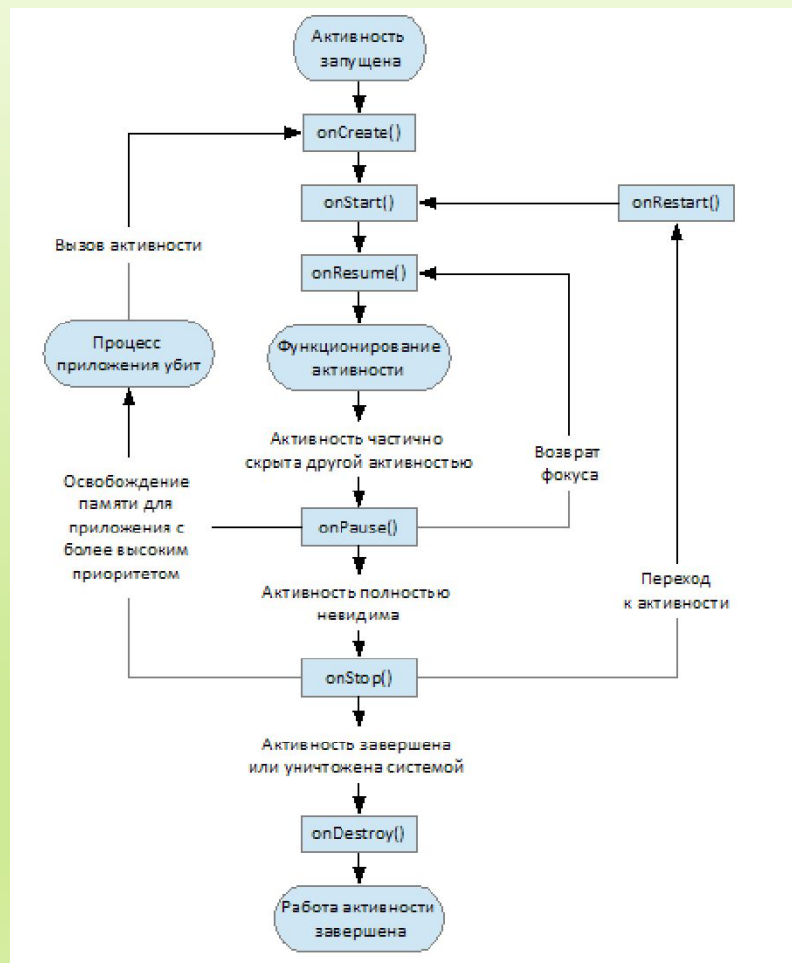
Введение в разработку приложений для смартфонов на ОС Android

# Архитектура приложения

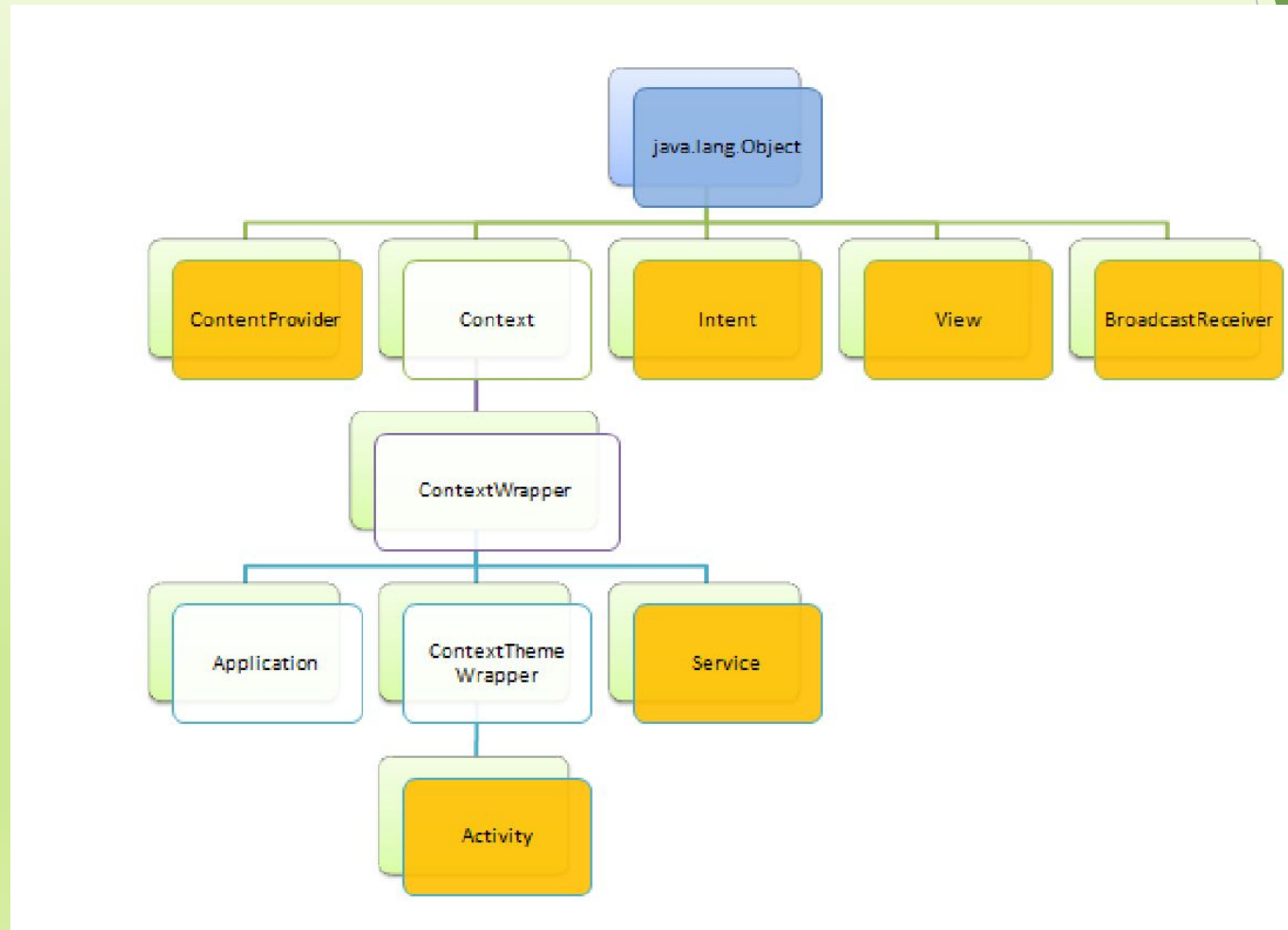
- ▶ Архитектура Android приложений основана на идее многократного использования компонентов, которые являются основными строительными блоками
- ▶ Каждый компонент является отдельной сущностью и помогает определить общее поведение приложения



# Жизненный цикл активности



# Иерархия классов Android SDK



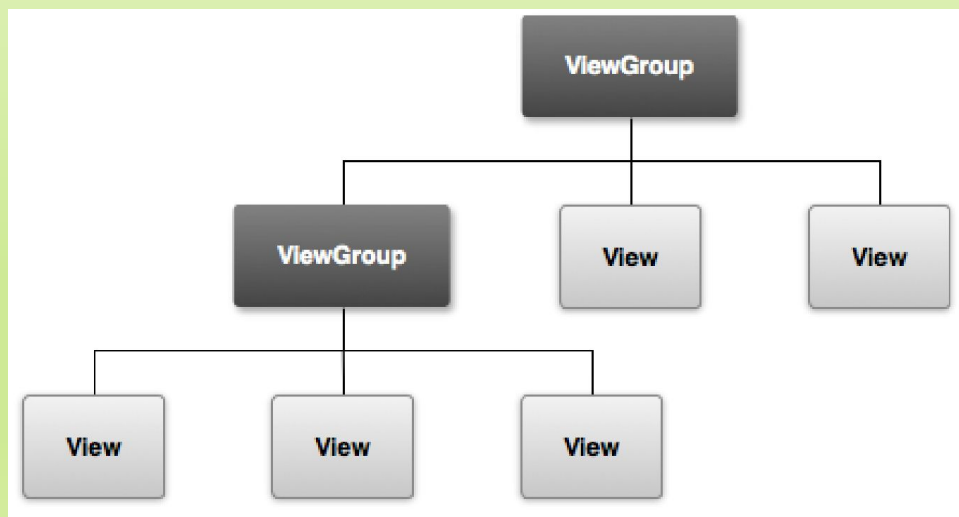
# Класс View

- ▶ Основной строительный блок для компонентов пользовательского интерфейса (UI)
- ▶ Определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий



# Класс View

- ▶ Базовый класс для виджетов (GUI widgets) и класса ViewGroup



# Класс Intent

- ▶ Используются для передачи сообщений между основными компонентами приложений
- ▶ Содержит описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса



# Класс Intent

Механизмы передачи намерений:

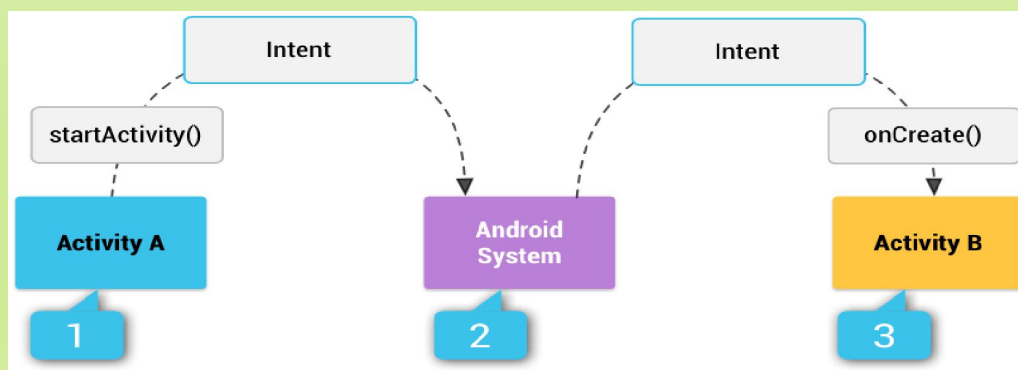
- ▶ Запуск активности, новое действие `Context.startActivity()` или `Activity.startActivityForResult()`
- ▶ Запуск сервиса `Context.startService()`, связь вызывающего компонента и сервиса `Context.bindService()`
- ▶ Доставка объекта-намерения приемникам широковещательных сообщений  
`Context.sendOrderedBroadcast()`,  
`Context.sendStickyBroadcast()`, `Context.sendBroadcast()`





# Класс Intent

- ▶ В этой системе сообщений не случается накладок: сообщение-намерение, отправленное определенному компоненту, будет получено именно этим компонентом и никем другим



# Манифест приложения

- ▶ Корневой каталог каждого приложения под Android должен содержать файл **AndroidManifest.xml**
- ▶ Содержит всю необходимую информацию, используемую системой для запуска и выполнения приложения



# Манифест приложения

Основная информация в манифесте:

- ▶ Имя Java пакета приложения
- ▶ Описание компонентов приложения
- ▶ Определение процессов
- ▶ Объявление полномочий, которыми должно обладать приложение для доступа к защищенным частям API и взаимодействия с другими приложениями



# Манифест приложения

Основная информация в манифесте:

- ▶ Объявление полномочий, которыми должны обладать другие приложения для взаимодействия с компонентами данного
- ▶ Список вспомогательных классов
- ▶ Определение минимального уровня Android API для приложения
- ▶ Список библиотек связанных с приложением



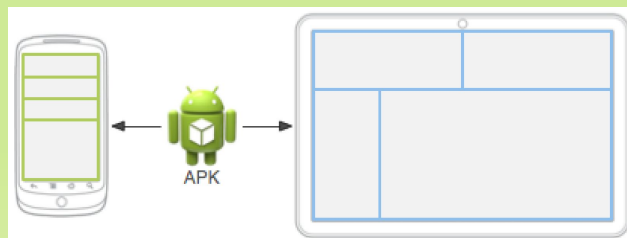
# Ресурсы

- ▶ Каждый тип ресурсов необходимо размещать в специальной поддиректории папки res/

а) используется компоновка по умолчанию (приложение не содержит альтернативы)



б) каждое устройство использует соответствующую компоновку



# Ресурсы

- ▶ **animator/** – свойства анимации
- ▶ **anim/** –анимация преобразований
- ▶ **color/** –списки цветов
- ▶ **drawable/** – графические файлы
- ▶ **layout/** – компоновка элементов пользовательского интерфейса
- ▶ **menu/** – все меню приложения
- ▶ **values/** –простые значения строк, чисел, цвета



# Ресурсы

- ▶ Все ресурсы после определения могут быть доступны по ссылке на их ID, которые определены в автоматически генерируемом классе **R**. Для каждого типа ресурсов в **R** классе существует подкласс, например, **R.drawable** для всех графических ресурсов

