

# Жизненный цикл

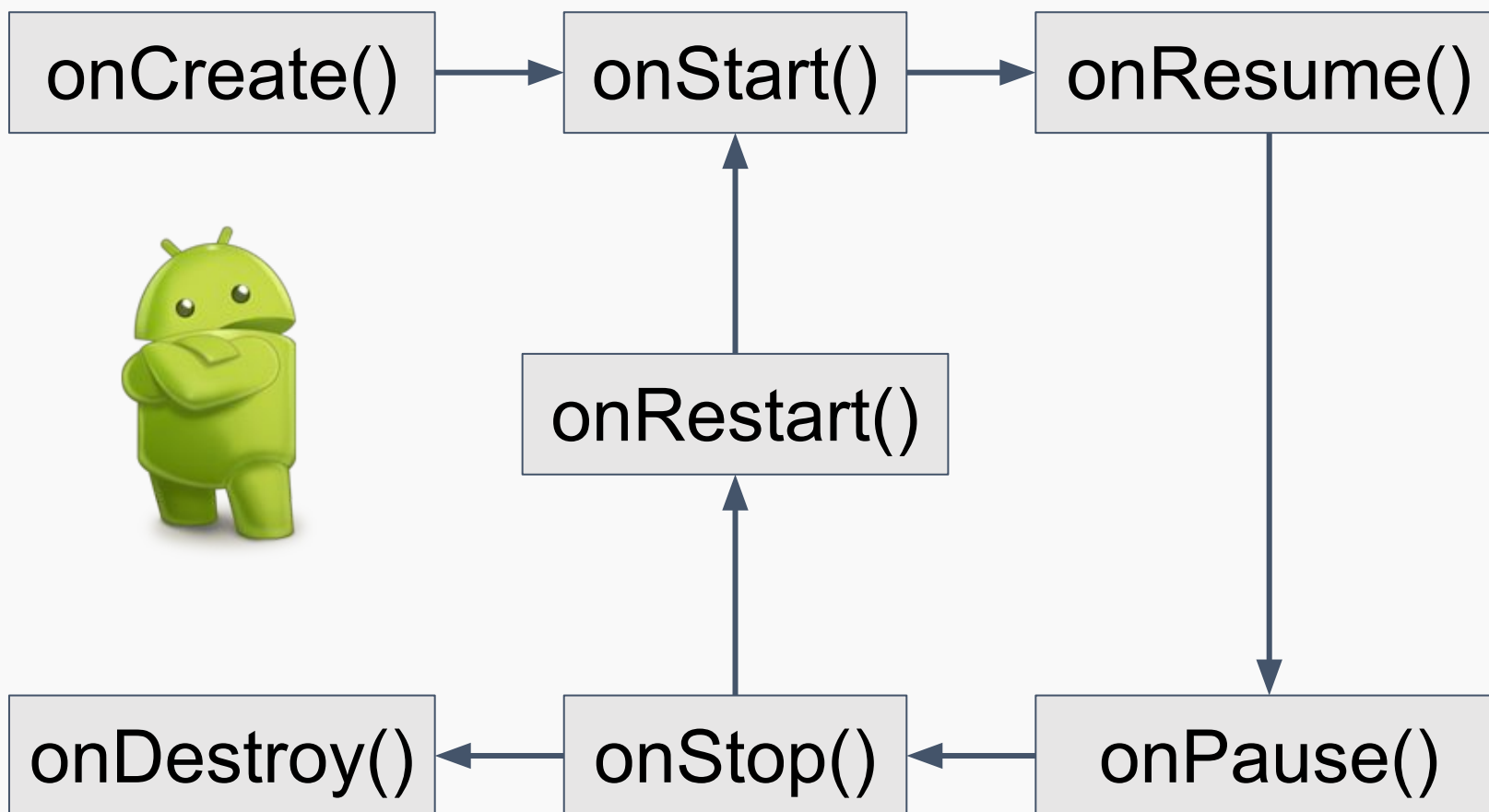
Лекция №4



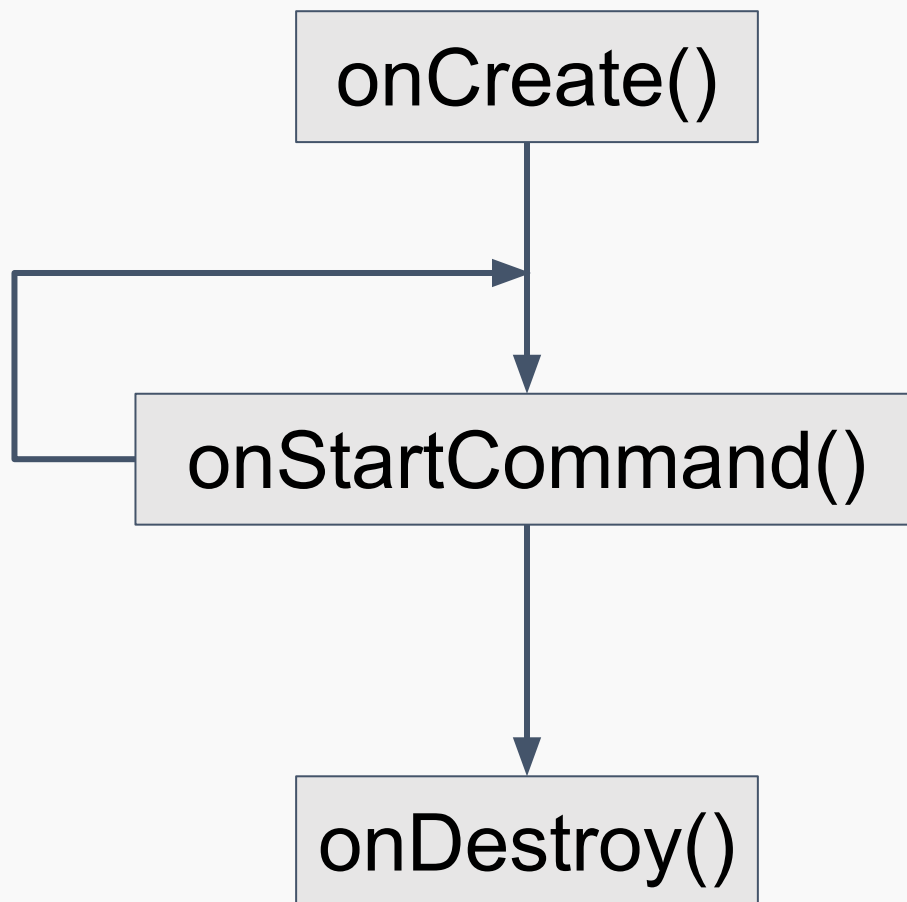
Попов Даниил

**Не забывайте  
отмечаться на занятии!**

# Жизненный цикл Activity



# Жизненный цикл Service



# Жизненный цикл BroadcastReceiver

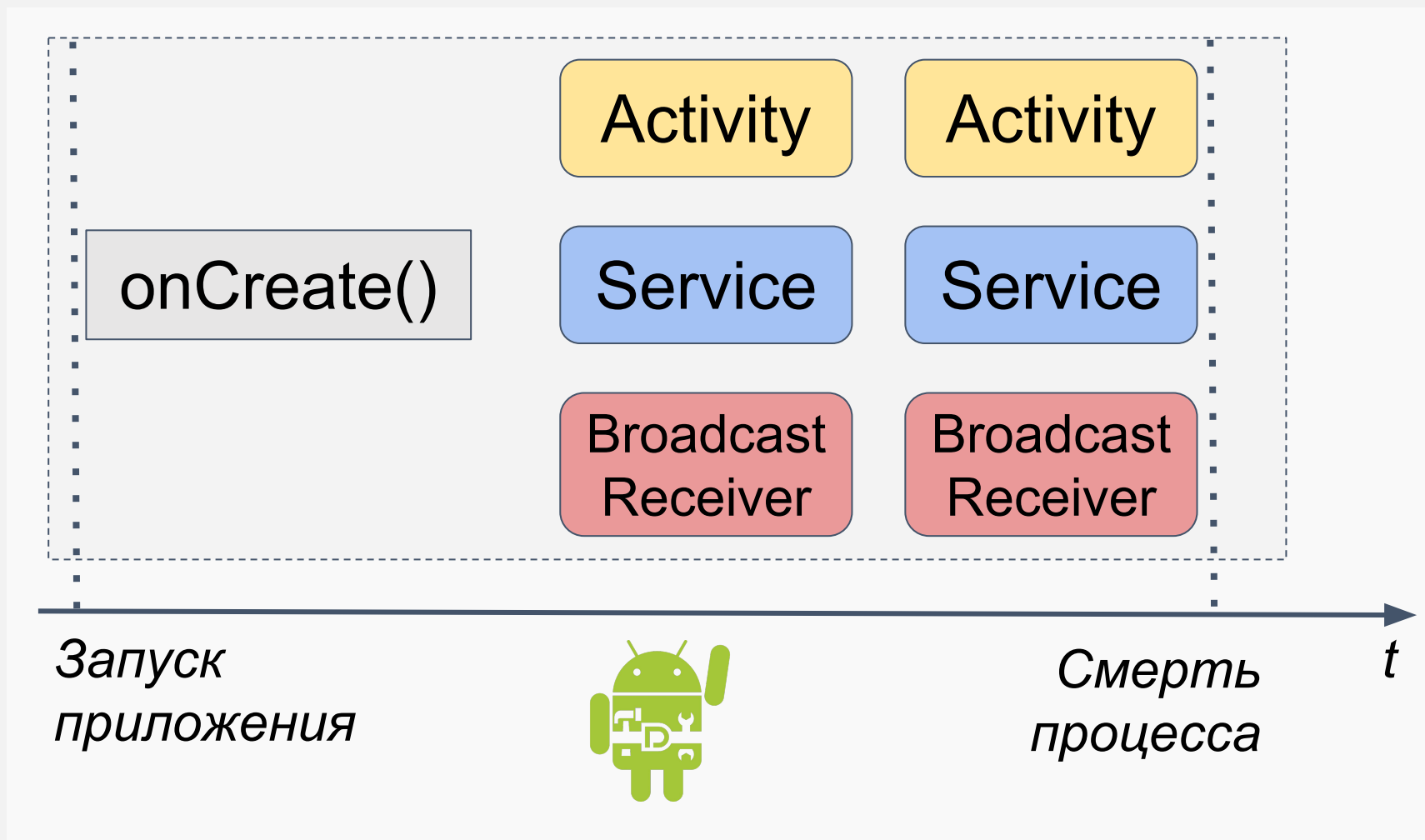
---



onReceive()



# Жизненный цикл приложения



# Приоритет приложений в системе

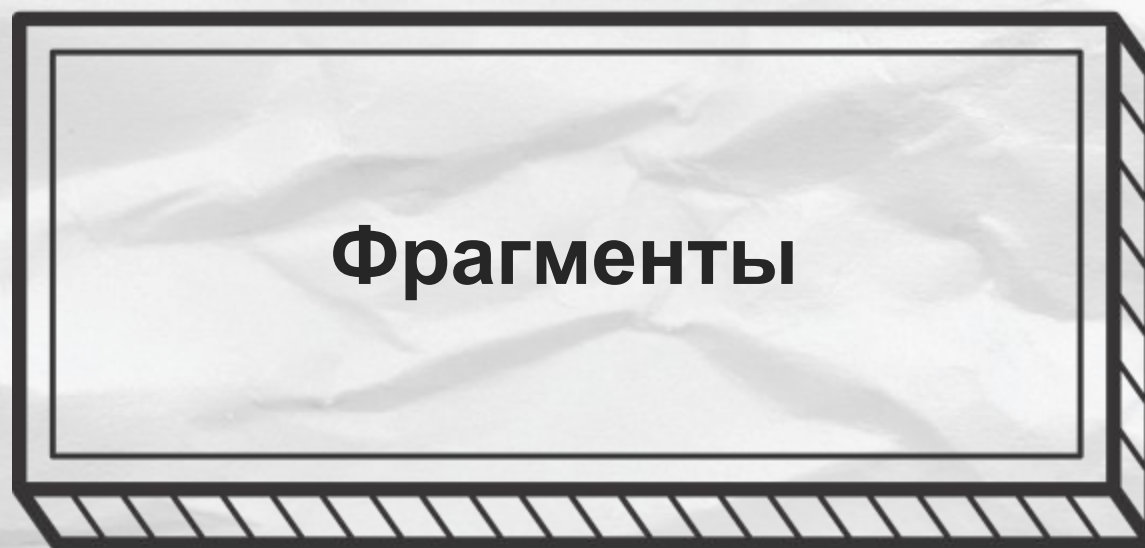
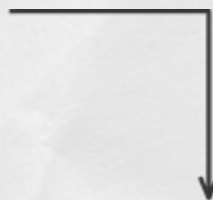
---



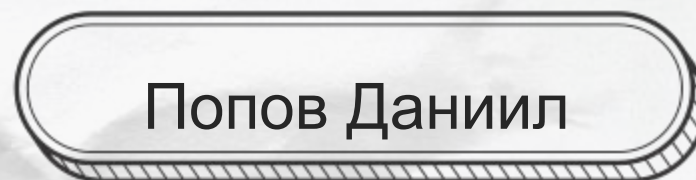
1. Foreground Process
2. Visible Process
3. Service Process
4. Background (cached) Process
5. Empty Process







Лекция №4

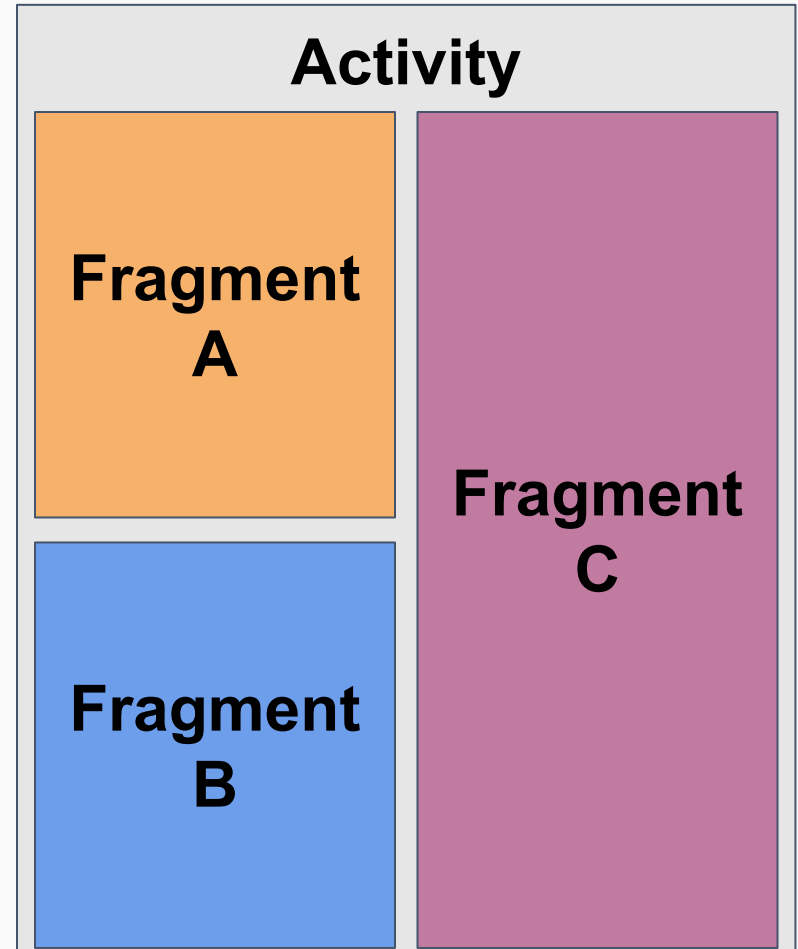




# Что такое Fragment?



1. **Fragment** - “кусочек” *Activity*;
2. Имеет свой жизненный цикл, связанный с *Activity*;
3. Взаимодействует с пользователем;
4. Могут добавляться на *Activity* в *runtime*;
1. Появились в *Api Level 11*.



# Философия Fragment'ов

---



1. Разработка “гибкого” пользовательского интерфейса, реагирующего на действия пользователя;
2. Рациональное использование пространства экрана на планшетах, а также в альбомной ориентации устройства;
3. Переиспользование кода и верстки.



# Жизненный цикл Fragment'a



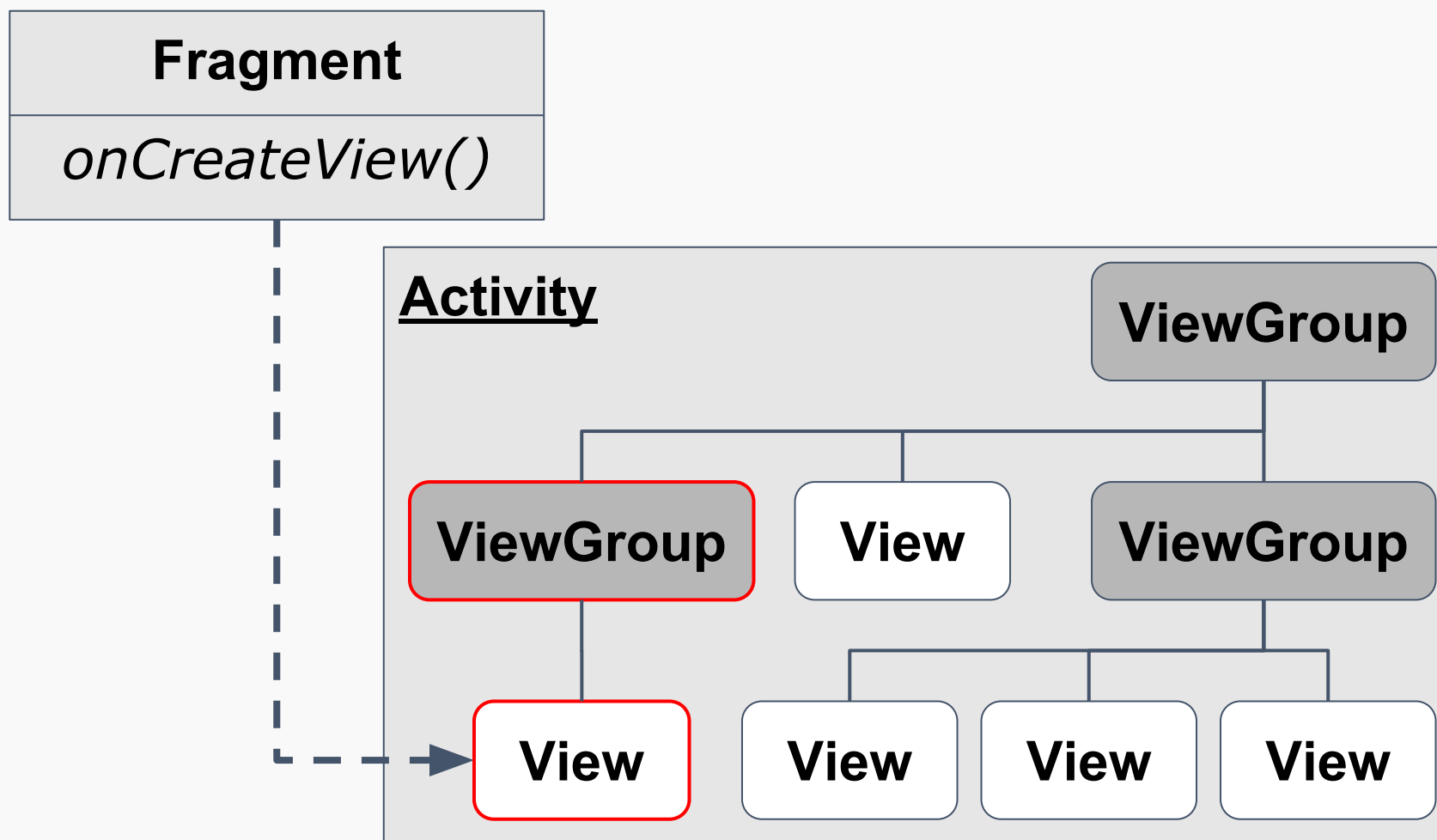
Activity State	Fragment Lifecycle Callback
Created	<ul style="list-style-type: none"><li>● <code>onAttach()</code></li><li>● <code>onCreate()</code></li><li>● <code>onCreateView()</code></li><li>● <code>onActivityCreated()</code></li></ul>
Started	<ul style="list-style-type: none"><li>● <code>onStart()</code></li></ul>
Resumed	<ul style="list-style-type: none"><li>● <code>onResume()</code></li></ul>

# Жизненный цикл Fragment'a



Activity State	Fragment Lifecycle Callback
Paused	<ul style="list-style-type: none"><li>● onPause()</li></ul>
Stopped	<ul style="list-style-type: none"><li>● onStop()</li></ul>
Destroyed	<ul style="list-style-type: none"><li>● onDestroyView()</li><li>● onDestroy()</li><li>● onDetach()</li></ul>

# Как это работает?



# Добавление Fragment'а в Activity



Существует два способа:

1. Через xml-разметку с помощью тега `<fragment>`
2. Программно, используя `FragmentManager`



# FragmentManager



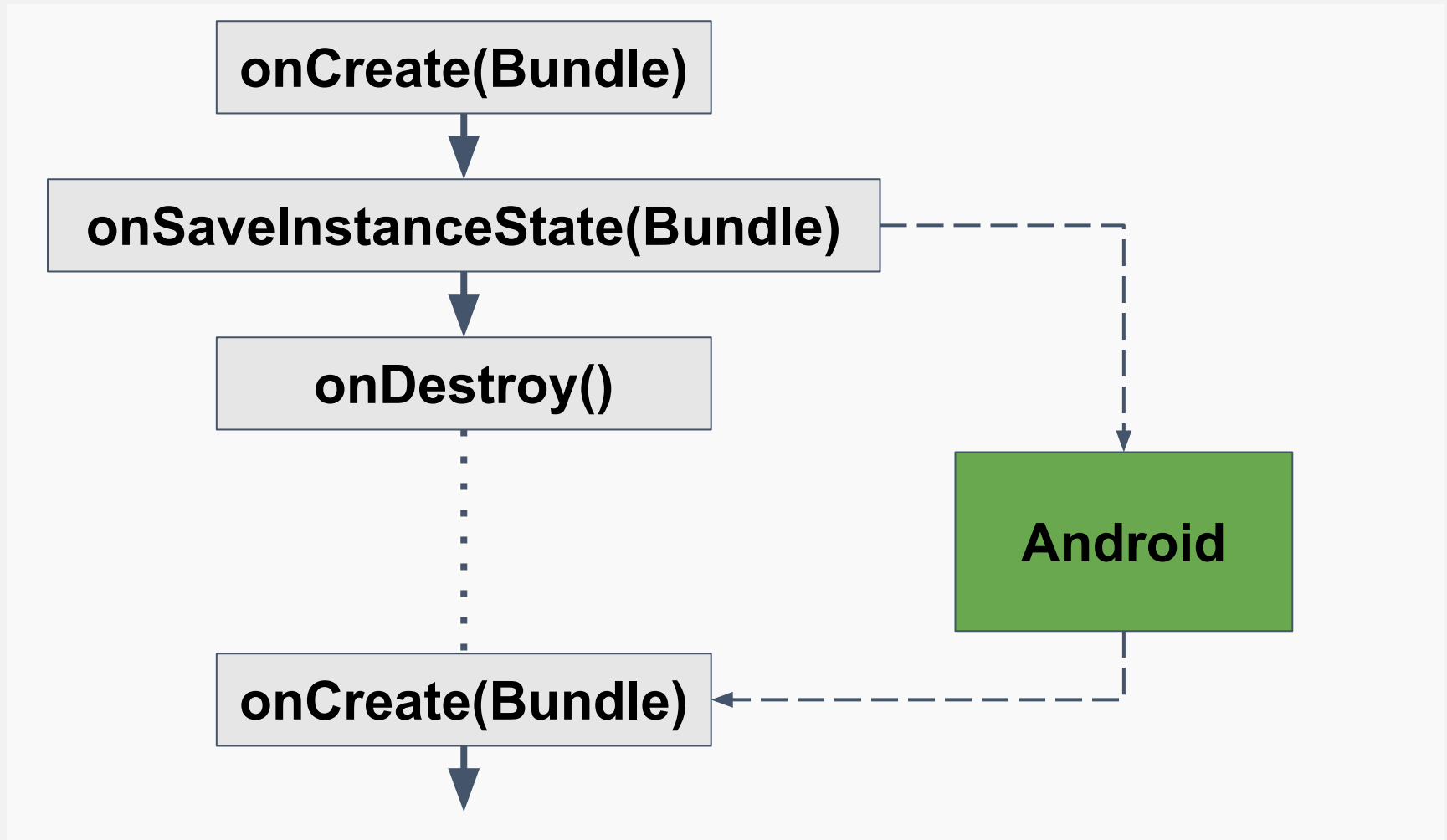
Любые действия с фрагментами обязательно выполняются в рамках **транзакции**. Доступны следующие операции:

- Добавление фрагмента;
- Удаление фрагмента;
- Замена фрагмента;
- Добавление транзакции в *back stack*;
- Установка анимации для транзакции.





# Немного о сохранении состояния



# Завершение транзакции



Существует два способа:

- `commit()` - завершение транзакции с учетом потери состояния (бросает исключение)
- `commitAllowingStateLoss()` - завершение транзакции с игнорированием потери состояния



**Дайте, пожалуйста,  
обратную связь!**



**Спасибо за внимание!**

**А теперь к практике!**