

# **Введение в разработку под Android**

**Семинар 3**

# На этом семинаре

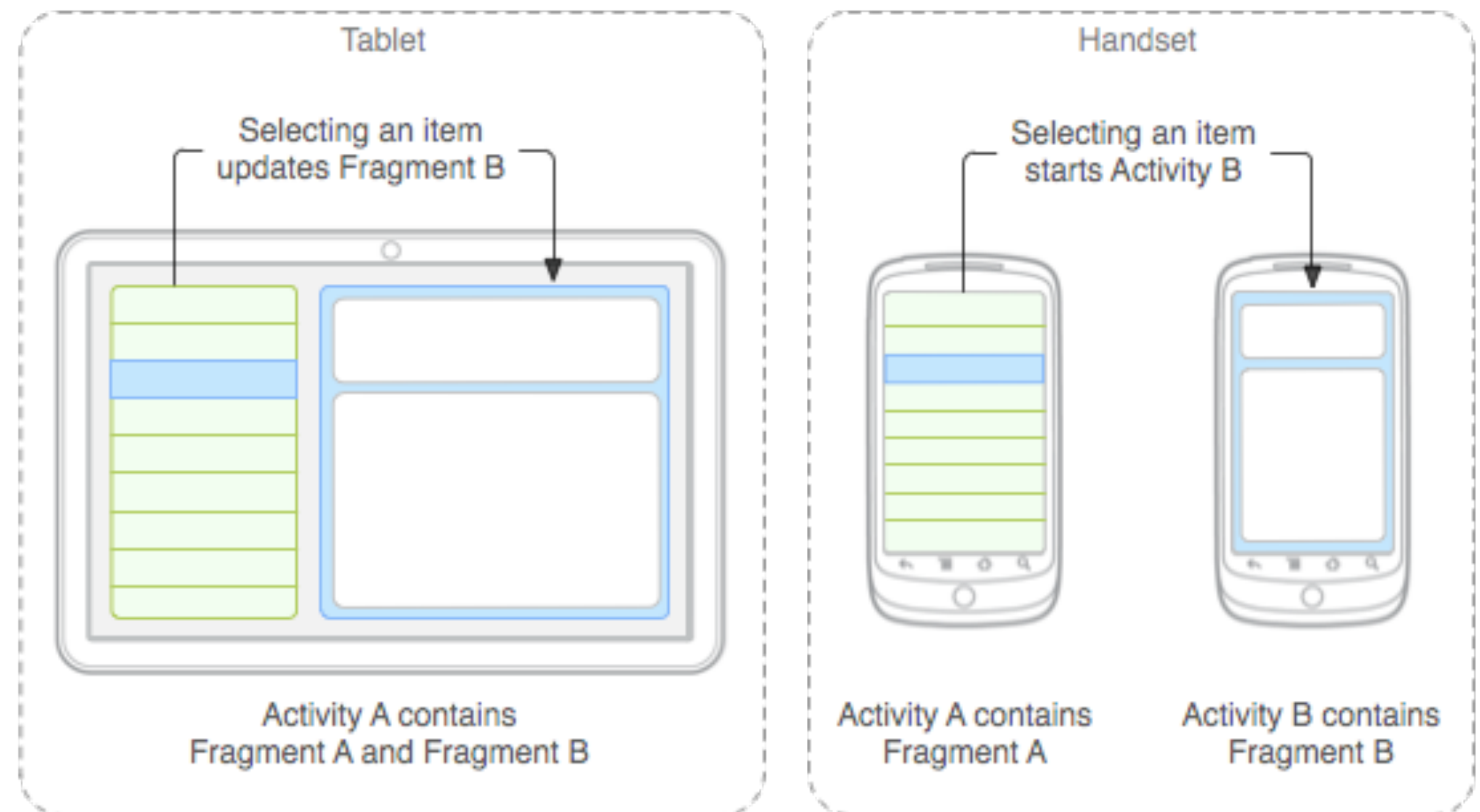
- Фрагменты и работа с ними
- RecyclerView
- Пример приложения

# Фрагменты

<https://developer.android.com/guide/components/fragments>

Переиспользуемые части Activity.  
Обычно определяют часть UI.

Способны обрабатывать  
пользовательские события, имеют  
свой жизненный цикл, не умеют  
“запускаться” отдельно от Activity.



# Тезисно о фрагментах

<https://developer.android.com/guide/components/fragments>

- Fragment - комбинация XML layout'а и Java класса (похожего на Activity).
- Через “support-lib”, фрагменты могут быть использованы на всех “релевантных” версиях андроида.
- Фрагменты инкапсулируют логику обработки View, что позволяет переиспользовать их для разных Activity.
- Фрагменты отдельные компоненты, которые содержат свои View, обрабатывают “свои” события и реализуют собственную “бизнес логику”.

# Когда использовать фрагменты

<https://developer.android.com/guide/components/fragments>

- Переиспользование View & Logic.
- Поддержка планшетов.
- Поддержка различных ориентаций экрана.

# Activity vs Fragment

<https://developer.android.com/guide/components/fragments>

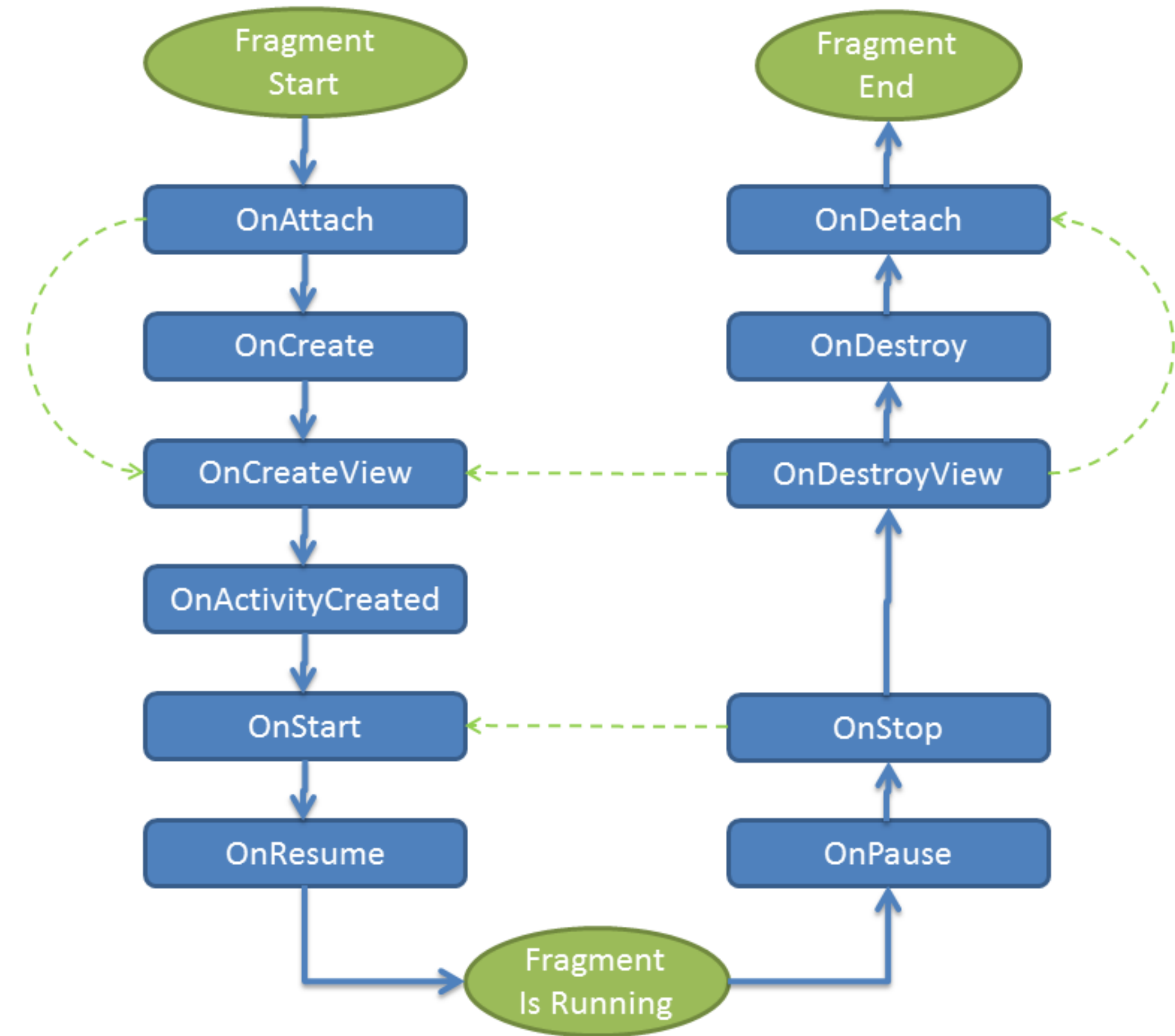
**Activities are navigation controllers primarily responsible for:**

- Navigation to other activities through intents.
- Presenting navigational components such as the navigation drawer or the viewpager.
- Hiding and showing relevant fragments using the fragment manager.
- Receiving data from intents and passing data between fragments.

**Fragments are content controllers and contain most views, layouts, and event logic including:**

- Layouts and views displaying relevant app content.
- Event handling logic associated with relevant views.
- View state management logic such as visibility or error handling.
- Triggering of network request through a client object.
- Retrieval and storage of data from persistence through model objects.

# Fragments LifeCycle



# Взаимодействие с фрагментами

<https://developer.android.com/guide/components/fragments>

Существует 3 “вида общения” Activity и Fragment:

- **Bundle** - Activity может создать фрагмент и задать ему аргументы.
- **Methods** - Activity может вызывать методы инстанса фрагмента.
- **Listener** - Fragment может посылать сообщения активности по средствам “подписок”.



# FragmentManager

<https://developer.android.com/guide/components/fragments>

Ответственный за runtime управление фрагментами.

- Может добавлять, удалять, заменять, скрывать, ... фрагменты по средствам **атомарных транзакций**.
- Позволяет управлять поведением “back” по средствам “тэгированого” стека (**backstack**).

# Навигация между фрагментами

<https://developer.android.com/guide/components/fragments>

Темы, которые сегодня не затронем, но будет полезно “пощупать”

- **TabLayout** - tabs at the top
- **Fragment Navigation Drawer** - slide out navigation menu
- **ViewPager** - swiping between fragments

# Bindings

**<https://developer.android.com/topic/libraries/view-binding>**

Позволяют по layout'у сгенерировать класс, который позволяет безопасно и удобно общаться с элементами View

# Recycler View

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

Сущность пришедшая на замену стандартным **ListView** и **GridView**

- **Adapter** - сущность-провайдер элементов View для отображения.
- **LayoutManager** - сущность ответственная за размещение элементов.
- **ViewHolder** - отображаемый элемент.

# Использование RecyclerView

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

- Выбираем подходящий **LayoutManager** (или реализуем свой):
  - **LinearLayoutManager** - организация в виде 1D списка.
  - **GridLayoutManager** - организация в виде 2D сетки.
  - **StaggeredGridLayoutManager** - 2D сетки, но ячейки могут быть различными.
- Описываем каждый элемент списка (как выглядит, как себя ведет).
- Реализуем **ViewHolder**.
- Реализуем свой **Adapter**, который предоставляет данные в View:
  - **onCreateViewHolder** - создает View, но еще не заполняет его данными.
  - **onBindViewHolder** - заполнение View данными.
  - **getItemCount** - количество элементов.

# Дополнительно

<https://developer.android.com/courses/fundamentals-training>

Выполнить урок "Unit 1: Get started" -> "Lesson 2: Activities and intents" -> "2.2: Activity lifecycle and state"