**ОГЛАВЛЕНИЕ**

## ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков создания списков.

Основными задачами выполнения лабораторной работы являются:

1. Научиться использовать компонент ExpandableListView для создания интерфейса приложения при решении практических задач.
2. Научиться использовать компонент Spinner и его обработчики событий в совместной работе разных компонентов приложения.
3. Научиться разрабатывать мобильные приложения с использованием компонента GridView.
4. Разработать эффективные приложения с учетом аппаратных ограничений мобильных устройств.
5. Уметь реализовывать логику работы приложения с учетом специфики платформы Android.

Результатами работы являются:

- Программа, использующая основные компоненты (ExpandableListView, Spinner или GridView) для решения задачи согласно варианту задания
- Подготовленный отчет

# КОМПОНЕНТЫ ANDROID-ПРИЛОЖЕНИЙ

Компоненты являются основными строительными блоками Android приложений. Каждый компонент является отдельной точкой входа в разработанное приложение. Не все компоненты являются фактически точками входа для пользователя, некоторые из них зависят друг от друга, но каждый из компонентов представляет из себя уникальный строительный блок, который помогает определить поведение приложения в системе.

Существует четыре разных типа компонентов. Каждый из них служит для различных целей, имеет свой собственный жизненный цикл, который определяет, как компонент создается и уничтожается.

**Активности (Activities)**

Активности представляют собой единый экран с пользовательским интерфейсом. Например, почтовый клиент может использовать одно явление для отображения списка входящих писем, другое явление для чтения конкретного письма, а третье явление для написания нового сообщения. И хотя все активности работают вместе, чтобы сформировать общую функциональность приложения, каждое из них независимо от других. Кроме того, приложения могут запускать активности друг друга (если это разрешено). Например, приложение фотокамера может запустить явление почтового клиента для создания нового письма с только что отснятой фотографией в качестве вложения.

Активности реализуются как подклассы базового класса Activity.

**Сервисы (Services)**

Сервис, это компонент, работающий в фоновом режиме и предназначенный для выполнения длительных операций или удаленных процессов. Сервис не предоставляет пользовательский интерфейс. Например, сервис может проигрывать музыку на заднем фоне, пока пользователь находится в другом приложении, или может закачивать данные по сети, не мешая пользователю взаимодействовать с

устройством. Другие компоненты, например, явления, могут запускать сервисы для отдельной работы или могут взаимодействовать с ними.

Службы реализуются как подклассы базового класса Service.

**Поставщики содержимого (Content Provider)**

Поставщики содержимого управляют общими данными в приложении. Например, можно хранить данные в файловой системе, в базе данных SQLite, в сети, в любом другом постоянном хранилище, к которому приложение имеет доступ. Через поставщики содержимого, другие приложения могут получить или модифицировать данные вашего приложения (если это разрешено). Например, Android предоставляет поставщик содержимого для управления контактными данными. Таким образом, любое приложение, которое имеет соответствующее разрешение, может запросить поставщики содержимого (например ContactsContract.Data) для чтения или изменения информации о каком-либо человеке.

Поставщики содержимого также удобны для чтения и записи не открытых данных приложения. Например, приложение NotePad использует поставщики содержимого для сохранения записей.

Поставщики содержимого реализуются как подклассы базового класса ContentProvider и должны содержать стандартный набор методов, которые позволят другим приложениям выполнять транзакции.

**Широковещательные приемники (Broadcast Reciever)**

Широковещательный приемник, это компонент, который реагирует на общесистемное оповещение. Многие широковещательные рассылки создает система — например, уведомление о выключении экрана, о низком заряде батареи, о снятой фотографии. Приложения также могут создавать рассылки — например, уведомление об окончании загрузки файла, чтобы другие приложения могли его использовать. Хотя широковещательные приемники не имеют пользовательского интерфейса, они могут создавать уведомления в строке состояния, чтобы предупредить пользователя о происходящем событии. Однако

чаще всего широковещательный приемник является просто мостом между другими компонентами и делает лишь малую часть работы. Например, это может быть запуск сервиса для выполнения работы, основанной на событии.

Широковещательные приемники реализуются как подклассы базового класса BroadcastReceiver и каждая рассылка передается как объект типа Intent.

Уникальной особенностью Android является возможность запуска одним приложением компонентов другого. Например, пользователь хочет сделать фотографию. Наверняка на устройстве уже есть приложение, которое это умеет делать, и можно использовать его, вместо того, чтобы разрабатывать подобный функционал заново. При этом не нужно добавлять исходный код другого приложения и даже ссылки на него. Вместо этого нужно просто запустить активность приложения Камера и сделать фотографию. После спуска затвора, фотография будет передана в приложение, где ее можно использовать. Для пользователя будет казаться, что работа с камерой — это часть разработанного приложения.

Когда система запускает компонент, запускается процесс для приложения (если оно не было уже запущено) и создаются экземпляры классов, необходимые для работы компонента. Например, если приложение запустило активность другого приложения, это активность будет работать в процессе другого приложения. Поэтому, в отличие от приложений в большинстве других систем, Android приложения не имеют единой точки входа в программу (например, нет основной функции main()).

Поскольку система запускает каждое приложение в отдельном процессе с разрешением для файлов, которое ограничивает доступ других приложений, приложение не может напрямую активировать компоненты другого приложения. Но это может сделать система Android. Таким образом, чтобы запустить компонент другого приложения, следует передать в систему сообщение, которое определяет намерение пользователя для запуска конкретного компонента.

## EXPANDABLELISTVIEW

Двухуровневый список ExpandableListView представляет собой объект view, в котором каждый элемент списка содержит вложенный список. Компонент является расширенным вариантом компонента ListView. Основное отличие - разворачивающий список второго уровня. За работу списка отвечает адаптер ExpandableListViewAdapter, который загружает данные по каждому пункту и вложенному списку.

Рассмотрим наиболее полезные методы, которые используются при работе с классом ExpandableListView:

- Метод setChildIndicator(Drawable) используется для отображения индикатора возле каждого элемента списка.
- Метод setGroupIndicator(Drawable) устанавливает индикатор возле каждого элемента, обозначая развернут он или свернут. Если группа пуста, то будет установлено состояние state_empty. Если группа развернута, будет установлено состояние state_expanded.
- Метод getGroupView() возвращает view для группы элементов списка
- Метод getChildView() возвращает view дочернего элемента списка

Для каждого элемента используется свой тип слушателя:

- Интерфейс ExpandableListView.OnChildClickListener переопределяется для отслеживания нажатий на дочерние элементы раскрывающегося списка
- Интерфейс ExpandableListView.OnGroupClickListener переопределяется для отслеживания нажатий на группу элементов
- Интерфейс ExpandableListView.OnGroupCollapseListener используется для уведомления о том, что группа элементов была свернута

– Интерфейс ExpandableListView.OnGroupExpandListener используется для уведомления о том, что группа элементов была развернута

Реализация:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/LinearLayout1"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <ExpandableListView
      android:id="@+id/expListView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" >
   </ExpandableListView>

</LinearLayout>
```

ExpandableListView редко используется в составе разметки с другими элементами. Обычно такой список занимает весь экран, поэтому для подобных целей лучше использовать специальный класс ExpandableListViewActivity, который уже содержит в своём составе компонент ExpandableListView.

## SPINNER

Компонент Spinner из раздела Widgets похож на выпадающий список (ComboBox), используемый в ОС Windows. В закрытом состоянии компонент показывает одну строчку, при раскрытии выводит список в виде диалогового окна с переключателями.

Реализация:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Spinner
        android:id="@+id/cities"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

В качестве источника данных, как и для ListView, для Spinner может служить простой список или массив, созданный программно, либо ресурс string-array. Взаимодействие с источником данных также будет идти через адаптер. В данном случае определим источник программно в виде массива в коде MainActivity:

```
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.ArrayAdapter
import android.widget.Spinner

class MainActivity : AppCompatActivity() {
```

```
    var cities = arrayOf("Москва", "Самара", "Вологда", "Волгоград",
        "Саратов", "Воронеж")
    override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

      val spinner = findViewById(R.id.cities) as Spinner
      // Создаем адаптер ArrayAdapter с помощью массива строк
      // и стандартной разметки элемета spinner
      val adapter = ArrayAdapter(this,
          android.R.layout.simple_spinner_item, cities)
      // Определяем разметку для использования при выборе
элемента

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dro
pdown_item)
      // Применяем адаптер к элементу spinner
      spinner.adapter = adapter
    }
  }
```

Используемый при создании ArrayAdapter ресурс android.R.layout.simple_spinner_item предоставляется платформой и является стандартной разметкой для создания выпадающего списка.

С помощью метода adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item) устанавливаются дополнительные визуальные возможности списка. А передаваемый в метод ресурс android.R.layout.simple_spinner_dropdown_item используется для визуализации выпадающего списка и также предоставляется платформой.

## GRIDVIEW

Компонет GridView представляет собой плоскую таблицу. Для GridView можно использовать собственные поля для отображения элементов данных, создав класс, производный от класса ArrayAdapter или BaseAdapter и т.п, и переопределив его метод getView().

Находится в разделе Containers.

Число столбцов для GridView чаще задается статически. Число строк в элементе определяется динамически на основании числа элементов, которые предоставляет адаптер.

Существует следующий набор свойств:

- android:numColumns — определяет количество столбцов. Если поставлено значение auto_fit, то система вычислит количество столбцов, основанное на доступном пространстве
- android:verticalSpacing — устанавливает размер пустого пространства между ячейками таблицы
- android:columnWidth — устанавливает ширину столбцов
- android:stretchMode — указывает, куда распределяется остаток свободного пространства для таблицы с установленным значением android:numColumns="auto_fit". Принимает значения columnWidth для распределения остатка свободного пространства между ячейками столбцов для их увеличения или spacingWidth — для увеличения пространства между ячейками

Реализация:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >
   <GridView
      android:id="@+id/gridview"
      android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:numColumns="2"
        android:verticalSpacing="16dp"
        android:horizontalSpacing="16dp"
        android:stretchMode="columnWidth" />

</LinearLayout>
```

С помощью атрибута android:numColumns можно настроить количество столбцов в гриде.

Теперь надо установить связь с адаптером:

```
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.GridView
import android.widget.Toast

class MainActivity : AppCompatActivity() {

   var countries = arrayOf("Бразилия", "Аргентина", "Чили", "Колумбия",
       "Уругвай", "Парагвай")
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

      // получаем элемент GridView
      val countriesList = findViewById<GridView>(R.id.gridview)

      // создаем адаптер
      val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
countries)
```

```
        countriesList.adapter = adapter

        val itemListener = AdapterView.OnItemClickListener { parent, v,
position, id ->
            Toast.makeText(
                applicationContext, "Вы выбрали " +
parent.getItemAtPosition(position).toString(),
                Toast.LENGTH_SHORT
            ).show()
        }
        countriesList.onItemClickListener = itemListener
    }
}
```

Для обработки нажатия в GridView применяется слушатель AdapterView.OnItemClickListener.

## ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

Программа должна быть реализована на языке высокого уровня Kotlin.

## ВАРИАНТЫ ЗАДАНИЙ

1. Создать список ExpandableListView. В качестве групп указать геометрические фигуры (не меньше 3 вариантов). В качестве элементов – цвет фигуры (не меньше 4 вариантов). При нажатии на кнопку отобразить фигуру с заданными параметрами на другой Activity.
2. Первая Activity содержит компонент EditText, Button и список ExpandableListView. В качестве групп выступают цвета (не менее 4 вариантов). В качестве элементов – размер шрифта (не менее 3 вариантов). При нажатии на кнопку отобразить введенный пользователем текст на другой Activity.
3. Создать список ExpandableListView, содержащий 4 группы. В качестве групп указать марки автомобилей. В качестве элементов модель и мощность двигателя автомобиля. При раскрытии группы списка в компоненте TextView отобразить модель с максимальной скоростью для выбранной группы.
4. Создать список ExpandableListView, содержащий не менее 4 пронумерованных групп, и компонент TextView. Каждая группа содержит не менее 3 элементов, представляющих собой строки. При выборе элемента отображать его значение в компоненте TextView. При разворачивании и сворачивании любой группы необходимо изменять цвет фона TextView в случайном порядке.
5. На основе компонентов ImageButton создать палитру цветов 3x3. Создать список ExpandableListView, в качестве групп для которого выступают геометрические фигуры (не меньше 3 вариантов). В качестве элементов – область отображения фигуры (в центре, слева, справа, вверху, внизу). При нажатии кнопки отобразить на другой Activity фигуру с выбранными параметрами.

6. На основе компонентов ImageButton создать палитру цветов 3x3. Создать список ExpandableListView, в качестве групп для которого выступают геометрические фигуры (не меньше 3 вариантов). Элементами группы могут быть разные размеры фигур ли другой варьируемый параметр. При нажатии кнопки отобразить на другой Activity фигуру с выбранными параметрами.

7. Создать три выпадающих списка Spinner и компонент EditText. В выпадающих списках выбирать цвет текста, размер и цвет фона. При нажатии кнопки отображать на другой Activity текст с заданными параметрами, используя компонент TextView.

8. Создать список ExpandableListView, содержащий не менее 4 пронумерованных групп. Каждая группа содержит не менее 3 элементов, представляющих собой строки. С помощью двух компонентов Spinner выбирать номер группы и номер элемента в группе. При нажатии кнопки отображать значение выбранного элемента на другой Activity с использованием TextView.

9. Создать 2 выпадающих списка и компонент GridView. Первый список определяет цвет ячейки компонента GridView, второй – цвет текста. При нажатии на кнопку отобразить на другой Activity GridView с заданными значениями в произвольной ячейке.

10. Создать три выпадающих списка Spinner и компонент EditText. В выпадающих списках выбирать цвет текста, размер и вид анимации. При нажатии кнопки отображать на другой Activity анимированный текст с заданными параметрами, используя компонент TextView. Особенности начертания определять с помощью трех компонентов CheckButton (курсив, полужирный, подчеркнутый).

11. Создать три выпадающих списка Spinner и компонент EditText. В выпадающих списках выбирать тип фигуры, цвет, размер. При нажатии кнопки отображать на другой Activity фигуру с заданными параметрами, используя компонент TextView. Область отображения фигуры определять с помощью компонентов RadioButton.

12. Создать 2 выпадающих списка и компонент GridView. Первый список определяет параметр numColumns компонента GridView, второй –

HorizontalSpacing. При нажатии на кнопку отобразить на другой Activity GridView с заданными значениями ячеек. Цвет ячейки определяется с помощью компонентов RadioButton (не менее 6).

13. Создать 2 выпадающих списка и компонент GridView. Первый список определяет параметр columnWidth компонента GridView, второй – StretchMode. При нажатии на кнопку отобразить на другой Activity GridView с произвольными значениями ячеек и заданными значениями параметров.

14. Создать 2 выпадающих списка и компонент GridView. Первый список определяет параметр numColumns компонента GridView, второй – HorizontalSpacing. При нажатии на кнопку отобразить на другой Activity GridView с заданными значениями ячеек. Цвет ячейки определяется с помощью компонентов ImageButton (не менее 6).

15. Создать три выпадающих списка Spinner и компонент EditText. Каждый из списков содержит значения от 0 до 255 (не менее 20). При нажатии кнопки отображать анимированный текст выбранного цвета на другой Activity. Тип анимации определять с помощью трех компонентов RadioButton.

16. Создать три выпадающих списка Spinner. В списках содержатся возможные значения коэффициентов квадратного уравнения (от -100 до 100). При нажатии кнопки решить квадратное уравнение, результат отобразить на другой Activity с использованием компонента TextView (в результате отображать все два корня, в том числе и комплексные).

17. Создать компонент EditText. При нажатии на кнопку на другой Activity отобразить компонент GridView, содержащий по одному символу введенного текста в каждой ячейке. Цвет ячейки определяется с помощью трех компонентов RadioButton, а значение параметра HorizontalSpacing компонента GridView выбирается из выпадающего списка Spinner.

## КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Раскройте понятие компонент в Android приложении.
2. Перечислите виды компонентов.
3. Опишите механизмзапуска одним приложением компонентов другого
4. Опишите роль компонент ExpandableListView.
5. Опишите реализацию ExpandableListView.
6. Опишите роль компонент Spinner.
7. Опишите реализацию Spinner.
8. Опишите роль компонент GridView.
9. Опишите реализацию Spinner.

## ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 1,5 занятия (3 академические часа: 2 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), листинг (xml код разметки экрана, графическое представление, соответствующее разметке экрана, Kotlin код программы и при необходимости наличие кода дополнительных классов), результаты выполнения работы (графические изображения примеров работы приложения), выводы.

## ОСНОВНАЯ ЛИТЕРАТУРА


1. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android / А. Семакова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 103 с. : ил. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429181
2. Введение в разработку приложений для ОС Android / Ю.В. Березовская, О.А. Юфрякова, В.Г. Вологдина и др. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 434 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428937
3. Разработка приложений для смартфонов на ОС Android / Е.А. Латухина, О.А. Юфрякова, Ю.В. Березовская, К.А. Носов. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 252 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428807


## ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА


4. Дэвид, Х. Разработка приложений Java EE 6 в NetBeans 7. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2013. — 330 с. — Режим доступа: http://e.lanbook.com/book/58693 — Загл. с экрана.
5. Сильвен, Р. Android NDK. Разработка приложений под Android на C/C++. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2012. — 496 с. — Режим доступа: http://e.lanbook.com/book/9126 — Загл. с экрана.
6. Ретабоуил, С. Android NDK: руководство для начинающих. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2016. — 518 с. — Режим доступа: http://e.lanbook.com/book/82810 — Загл. с экрана.

7. Соколова, В.В. Разработка мобильных приложений: учебное пособие / В.В. Соколова ; Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет», Министерство образования и науки Российской Федерации. - Томск: Издательство Томского политехнического университета, 2015. - 176 с.: ил., табл., схем. - Библиогр. в кн.. - ISBN 978-5-4387-0369-3; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=442808 (15.06.2017).
8. Баженова, И.Ю. Язык программирования Java / И.Ю. Баженова. - М.: Диалог-МИФИ, 2008. - 254 с. : табл., ил. - ISBN 5-86404-091-6 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=54745 (15.06.2017).
9. Джошуа Блох Java. Эффективное программирование [Электронный ресурс]/ Джошуа Блох— Электрон. текстовые данные. — Саратов: Профобразование, 2017.— 310 с.— Режим доступа: http://www.iprbookshop.ru/64057.html .— ЭБС «IPRbooks»

**Электронные ресурсы:**

10. Научная электронная библиотека http://eLIBRARY.RU
11. Электронно-библиотечная система http://e.lanbook.com
12. Электронно-библиотечная система «Университетская библиотека онлайн» http://biblioclub.ru
13. Электронно-библиотечная система IPRBook http://www.iprbookshop.ru
14. Базовый сайт о системе Android - https://www.android.com/intl/ru_ru
15. Разработка приложения на базе Android - https://developer.android.com/index.html