



УДК 004

Составители: А.В. Киселев, О.О. Яночкина

Рецензент

Кандидат технических наук, доцент *Ю.А. Халин*

**Разработка мобильных приложений:** методические указания к лабораторным работам для студентов направлений подготовки 09.03.01/ Юго-Зап. гос. ун-т; сост.; А.В. Киселев, О.О. Яночкина. – Курск, 2022. - 20 с.: - ил. 10 , табл. 1.– Библиогр.: с. 20

Содержат сведения по вопросам разработки мобильных приложений на базе операционной системы Android.

Предназначены для студентов направления подготовки 09.03.01 очной формы обучения.

Методические указания соответствуют рабочей программе дисциплины «Разработка мобильных приложений».

Текст печатается в авторской редакции

Подписано в печать . Формат 60\*84 1/16.  
Усл. печ. л. \_\_\_\_ . Уч.-изд. л. 1,0 . Тираж 50 экз. Заказ 900 . Бесплатно.  
Юго-Западный государственный университет.  
305040 Курск, ул. 50 лет Октября, 94.

## Содержание

|  |    |
|--|----|
| Лабораторная работа №1. Создание activity и передача параметров между ними | 4  |
| Лабораторная работа №2. Основы верстки                                     | 9  |
| Лабораторная работа №3. Работа с базой данных                              | 17 |
| Список литературы  | 20 |

## **Лабораторная работа №1. Создание activity и передача параметров между ними**

### **Цель работы:**

Знакомство с интерфейсом среды программирования. Изучение структуры проекта

### **Основные понятия Android проекта.**

#### **Структура проекта**

1. Src - «исходный код» приложения (java-классы)
2. Assets - пустая директория. Может использоваться для сохранения raw-файлов.
3. Gen - хранилище генерируемых системных файлов. В частности, здесь располагается файл R.java, в котором хранятся идентификаторы всех ресурсов, создаваемых в проекте (строковые ресурсы и т.п.).
4. Libs - различные библиотеки, используемые приложением
5. Res - ресурсы проекта.
6. AndroidManifest.xml - файл описания проекта (поддерживаемые версии SDK, версия приложения и т.п.)
7. Project.properties - файл, включающий настройки проекта, такие как build target.

#### **Ресурсы проекта**

1. anim/ Содержит XML файлы, компилируемые в анимационные объекты.
2. color/ Содержит XML файлы описывающие цвета.
3. drawable/ Содержит растровые файлы (PNG, JPEG, orGIF), 9-Patch файлы, и XML файлы, описывающие Drawableshapes или Drawableobjects включающие множественные состояния(нормальное, нажатое, состояние фокуса).
4. layout/ Содержит XML файлы описывающие макеты экрана
5. menu/ Содержит XML файлы, определяющие меню приложения.
6. raw/ Для хранения произвольных файлов.
7. values/ Содержит XML файлы, компилируемые во множество видов

ресурсов (strings.xml и т.д).

8. xml/ Содержит XML файлы конфигурирующие компоненты приложения. Например, XML файл определяющий экран настроек.

### **Пример простейшего файла AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.untitled"
android:versionCode="1"
android:versionName="1.0">
<uses-sdk android:minSdkVersion="19"/>
<application
                                android:label="@string/app_name"
android:icon="@drawable/ic_launcher">
<activity android:name="MyActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
</application>
</manifest>
```

### **Для начала работы требуется:**

1. Java Development Kit
2. Android Software Development Kit

### **Android Virtual Device Manager**

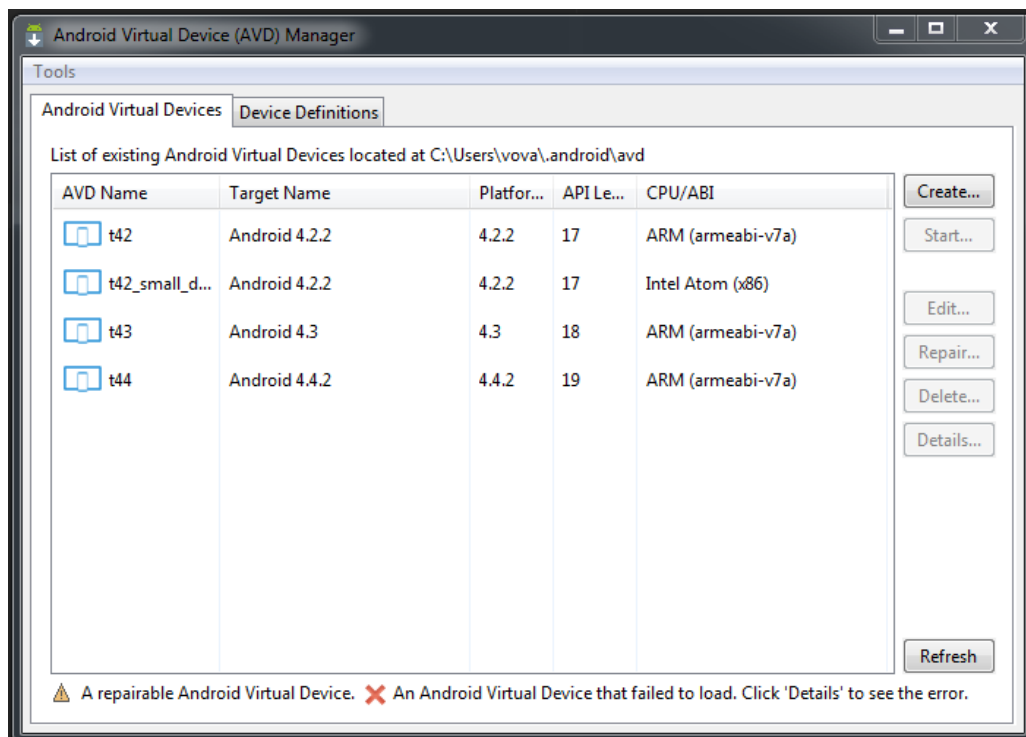


Рисунок 1.1 – Интерфейс виртуального менеджера

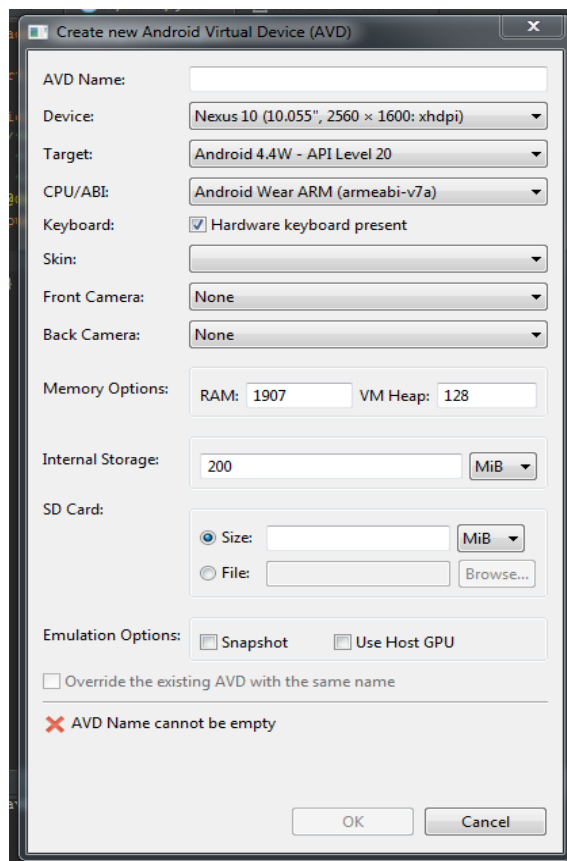


Рисунок 1.2 – Создание нового виртуального устройства

## Задания лабораторной работы

## Задание 1. Создание проекта приложения

Запустите среду программирования в IDE IntelliJ Idea

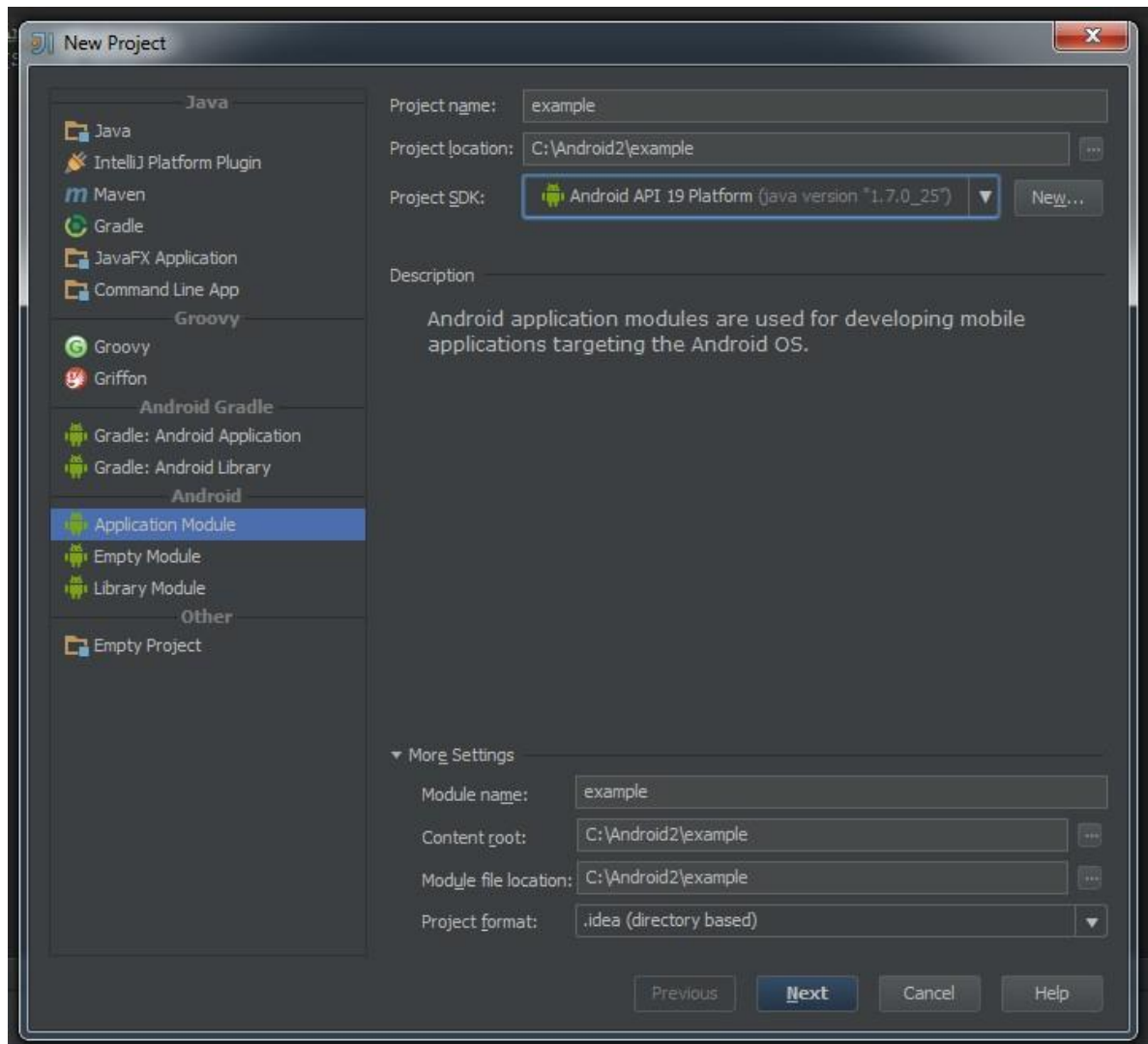


Рисунок 1.3 – Создание нового проекта в среде IntelliJ Idea

Введите данные проекта. Сохраните проект.

## Задание 2. Создание приложений с одним экраном (Activity)

Необходимо создать два **activity** и организовать переход между ними.

Содержимое Activity 1:

кнопка с именем btn1

Содержимое Activity 2:

TextView с текстом «Переданный параметр: значение\_параметра»,

значение\_параметра – принятый параметр из Activity 1.

При запуске приложения пользователь должен попадать на экран с **Activity 1**. После нажатия на кнопку btn1 необходимо осуществить переход к **Activity 2** и передавать параметр из **Activity 1**. В качестве значения передаваемого параметра использовать свою фамилию.

**Ключевые классы:** Activity, Intent, Button, TextView.



## Лабораторная работа №2. Основы верстки

### Цель работы:

Изучить основы верстки. Научиться управлять интерфейсом мобильного устройства при разработке программного приложения.

### Теоретические сведения

Просмотрите основные сведения о классах, которые понадобятся при разработке приложения.

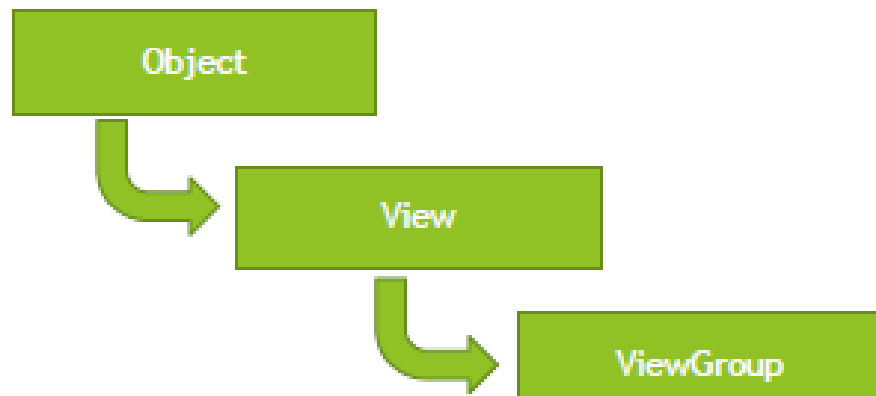


Рисунок 2.1. – Иерархия классов View и ViewGroup

Дерево представлений для Activity представлено на рисунке 2.2.

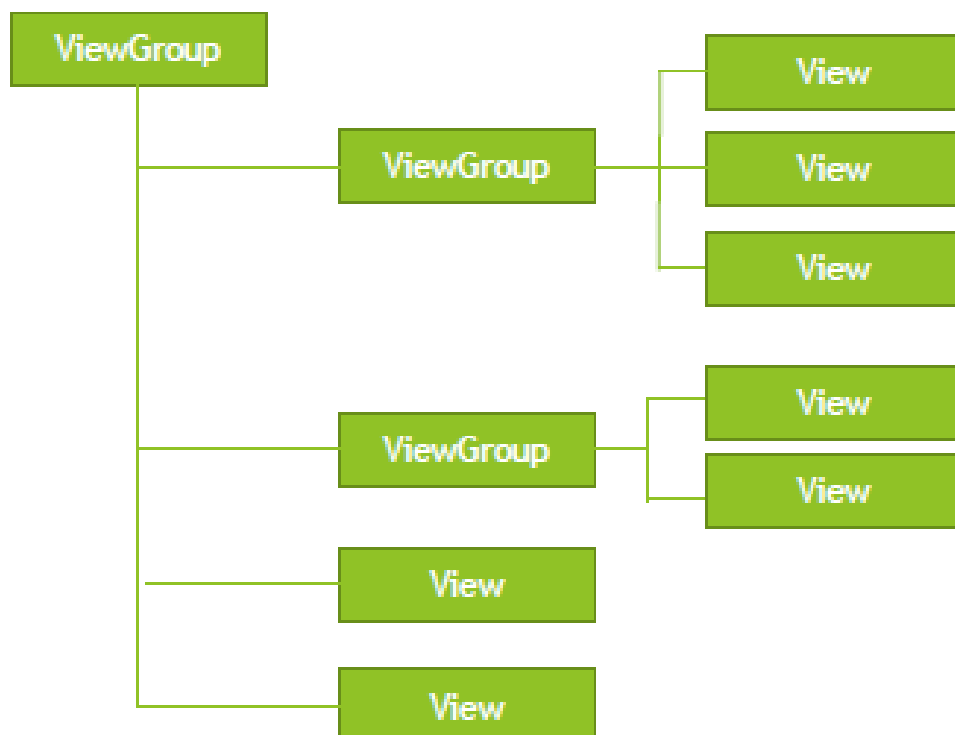


Рисунок 2.2 – Дерево представлений для Activity

Файл разметки имеет следующую структуру

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="0.25"
    android:padding="5dp">
<Button
    android:layout_width="0dp"
    android:layout_weight="0.33"
    android:layout_height="wrap_content"
    android:text="Button1"
    android:id="@+id/button3" android:layout_gravity="right"/>
<Button
    android:layout_width="0dp"
    android:layout_weight="0.33"
    android:layout_height="wrap_content"
    android:text="Button2"
    android:id="@+id/button"/>
<Button
    android:layout_width="0dp"
    android:layout_weight="0.33"
    android:layout_height="wrap_content"
    android:text="Button3"
    android:id="@+id/button2" android:layout_gravity="center_horizontal"/>

```

```
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:paddingLeft="40dp"
    android:paddingRight="40dp"
    android:layout_weight="0.5"
    android:gravity="center_vertical">
    <Button
        android:layout_width="0dp"
        android:layout_weight="0.33"
        android:layout_height="wrap_content"
        android:text="Button4"
        android:id="@+id/button3"/>
    <Button
        android:layout_width="0dp"
        android:layout_weight="0.33"
        android:layout_height="wrap_
content"
        android:text="Button5"
        android:id="@+id/button"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="0.25"
    android:padding="5dp"
    android:gravity="bottom">
...
</LinearLayout>
```

</LinearLayout>

## Распространенные виды макетов

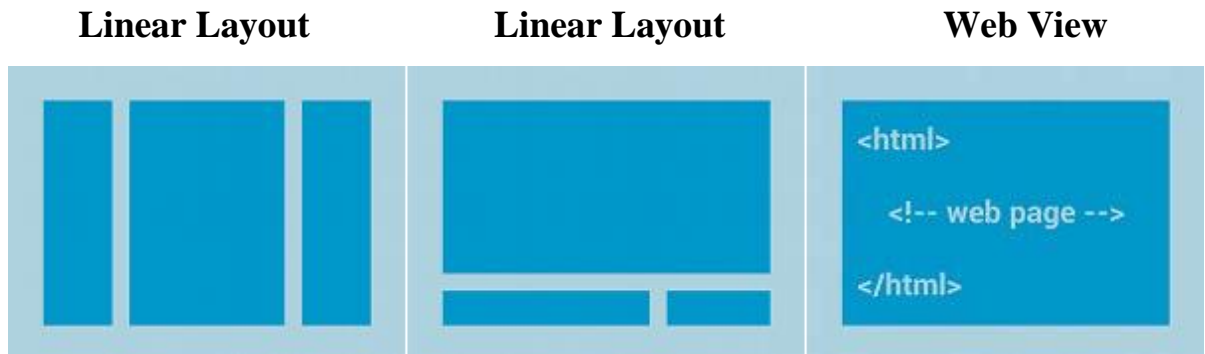


Рисунок 2.3. – Распространенные виды макетов

## Макеты с адаптером

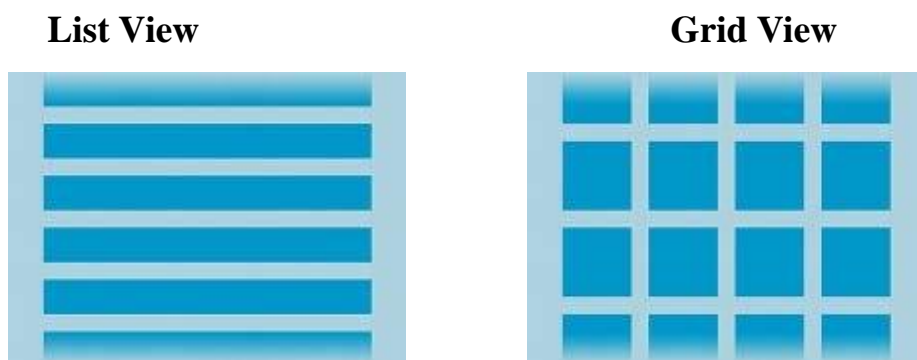


Рисунок 2.4 – Интерфейс макетов с адаптером

Таблица 2.1 – Атрибуты LinearLayout

| Attribute Name                    | Related Method                             | Description  |
|-----------------------------------|--|--|
| android:baselineAligned           | setBaselineAligned(boolea<br>n)            | When set to false, prevents the layout from aligning its children's baselines.   |
| android:baselineAlignedChildIndex | setBaselineAlignedChildIndex(int)          | When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which childTextView). |
| android:divider                   | setDividerDrawable(Drawable)               | Drawable to use as a vertical divider between buttons.   |
| android:gravity                   | setGravity(int)                            | Specifies how an object should position its content, on both the X and Y axes, within its own bounds.  |
| android:measureWithLargestChild   | setMeasureWithLargestChildEnabled(boolean) | When set to true, all children with a weight will be considered having the minimum size of the largest child.  |
| android:orientation               | setOrientation(int)                        | Should the layout be a column or row? Use "horizontal" for a row, "vertical" for a column.   |
| android:weightSum                 |  | Defines the maximum weight sum.  |

## Задания лабораторной работы

**Задание 1.** Разработать мобильное приложение, состоящее из четырех activity.

После запуска приложения пользователь должен попадать на экран с activity1. На данном экране должно быть представлено меню, состоящее из четырех кнопок. Высота кнопок должна составлять 20% от высоты экрана. Расстояние между кнопками – 2%. Первая и последняя кнопка должны быть на равном расстоянии от краев экрана. Ширина кнопок 75%, выравнивание по середине.

После нажатия на первую кнопку пользователь должен переходить к activity2, его внешний вид представлен на рисунке 1. Верстка должна осуществляться с использованием LinearLayout, ширина кнопок должна задаваться в процентах от ширины экрана.

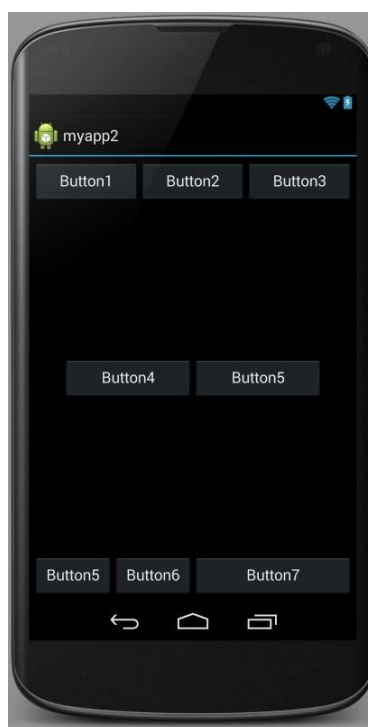


Рисунок 2.5 – Внешний вид экрана для первого задания

После нажатия на вторую кнопку в activity1 пользователь должен переходить к activity3, его внешний вид представлен на рисунке 2. Верстка должна осуществляться с использованием RelativeLayout (не использовать

LinearLayout).

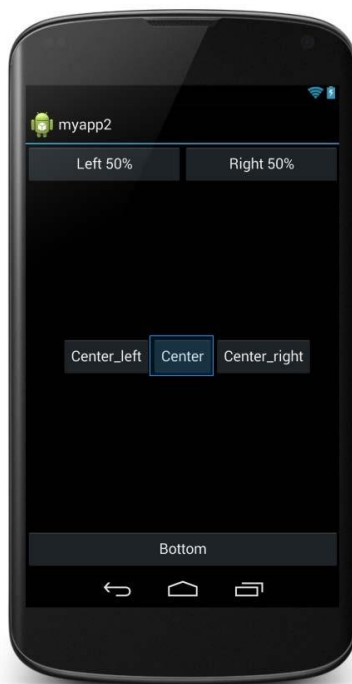


Рисунок 2.6. – Результат первого этапа выполнения задания  
Третья кнопка в activity1 должна создавать activity3.



Рисунок 2.7 – Интерфейс приложения на этапе activity3.

Кнопка должна быть выровнена по центру экрана. Цвет обводки кнопки

#505050. Толщина обводки в соответствии с месяцем вашего рождения (от 1 до 12). Радиус скругления 24dp. Цвет фона экрана #FFFFFF. При нажатии на кнопку ее цвет должен изменяться на светло-зеленый.

Нажатие на четвертую кнопку в activity1 должно приводить к закрытию приложения.



## **Лабораторная работа №3. Работа с базой данных**

### **Цель работы**

Изучить работу Android приложения с базой данных.

### **Теоретические сведения.**

#### **SQLite**

- SQLite—компактная встраиваемая реляционная база данных с открытым исходным кодом. Поддерживает динамическое типизирование данных.
- Возможные типы полей: INTEGER, REAL, TEXT, BLOB.

#### **SQLiteOpenHelper**

##### **Абстрактные методы**

- onCreate( ) -вызывается при первом создании базы данных;
- onUpgrade( ) -вызывается при модификации базы данных

##### **Реализация метода onCreate**

```
public void onCreate( SQLiteDatabase db) {  
    db.execSQL( "CREATE TABLE"+  
    TABLE_NAME  
    + "( _id INTEGER PRIMARY KEY AUTOINCREMENT ,"  
    + COL NAME + " TEXT ,"+ COL PHONE + " TEXT ) ; " );  
}
```

##### **Реализация метода onUpgrade**

```
@ Override  
public void onUpgrade( SQLiteDatabase db, int oldVersion, int newVersion)  
{  
    db.execSQL( "DROP TABLE IF EXISTS"+ TABLE_NAME );  
    onCreate( db );  
}
```

##### **Чтение из базы данных**

```
Cursor query( String table, String [ ]  
columns, String selection, String [ ]
```

selectionArgs,  
String groupBy, String having , String sortOrder)

### **Позиционирование курсора**

1. moveToFirst( ) -перемещает курсор в первую запись в выборке;
2. moveToLast( ) -перемещает курсор в последнюю запись в выборке;
3. moveToNext( ) -перемещает курсор в следующую запись и одновременноопределяет, существует ли эта запись. Метод moveToNext( ) возвращает true, если курсор указывает на другую строку после перемещения, и false,если текущая запись была последней в выборке;
4. moveToPrevious( ) -перемещает курсор в предыдущую запись;
5. moveToPosition( ) -перемещает курсор в указанную позицию;
6. getPosition( ) -возвращает текущий индекс позиции курсора.
7. isFirst( ) ;
8. isLast( ) ;
9. isBeforeFirst( ) ;
- 10.isAfterLast( ) .

### **Обновление и удаление записей**

- intupdate(String table, ContentValuesvalues, String whereClause, String [ ] whereArgs)
- intdelete ( String table, String whereClause, String [ ] whereArgs)

### **Задания на лабораторную работу**

**Задание 1.** Необходимо создать приложение, взаимодействующее с базой данных. Первое активити должно содержать три кнопки. При нажатии на первую кнопку должно открываться новое активити, выводящее информацию из таблицы «Одногруппники» в удобном для восприятия формате.

При запуске приложения необходимо:

1. Создать БД, если ее не существует.
2. Создать таблицу «Одногруппники», содержащую поля:
  - ID;
  - ФИО;
  - Время добавления записи.
3. Удалять все записи из БД, а затем вносить 5 записей об одногруппниках.

При нажатии на вторую кнопку необходимо внести еще одну запись в таблицу.

При нажатии на третью кнопку необходимо заменить ФИО в последней внесенной записи на Иванов Иван Иванович.

**Задание 2.** Создать новое отдельное приложение на основе приложения, созданного в части 1.

Переопределить функцию onUpgrade. При изменении версии БД необходимо удалить таблицу «Одногруппники», создать таблицу «Одногруппники» содержащую следующие поля:

- ID;
- Фамилия;
- Имя;
- Отчество;
- Время добавления записи.

Изменить версию базы данных.

## Список литературы

1. Куркин, А.В. Программирование под платформу Android [Текст]: учебное пособие / Куркин А.В. – СПб.: Университет ИТМО, 2015. - 35 с.
2. Григорьева, И.И. Разработка мобильных приложений [Текст]: учебное пособие / И.И. Григорьева, С.С. Самборецкий. – Тюмень, 2015. - 125 с.
3. Дэрсси, Л. Android за 24 часа. Программирование приложений под операционную систему Google [Текст] / Л.Дэрсси, Ш.Кондер. - ; М.: Рид Групп, 2011. - 464 с.
4. Колисниченко, Д. Программирование для Android [Текст]: самоучитель / Д. Колисниченко. – М.: БХВ-Петербург - Москва, 2012. - 272 с.
5. Майер, Р. Android 4. Программирование приложений для планшетных компьютеров и смартфонов [Текст] / Р. Майер. – М.: Эксмо, 2013. - 816 с.
6. Медникс, З. Программирование под Android [Текст] / З.Медникс, Л.Дорнин, Б.Мик, М.Накамура. – СПб.: Питер, 2013. – 560 с.
7. Ретабоуил, С. Android NDK. Разработка приложений под Android на C/C++ [Текст] / С. Ретабоуил. – М.: ДМК Пресс, 2012. - 496 с.
8. Цехнер, М. Программирование игр под Android [Текст] / М. Цехнер. – СПб.: Питер, 2013. – 688 с.