**Обзор RelativeLayout**

С компоновкой LinearLayout разобрались, а теперь приступим к изучению другой популярной компоновки [RelativeLayout](http://developer.android.com/reference/android/widget/RelativeLayout.html" \t "_blank).

В этой компоновке элементы в контейнере выстраиваются относительно либо контейнера, либо других элементов в этом контейнере.

Рассмотрим XML атрибуты и JAVA константы для выстраивания относительных связей.

**Расположение относительно родительского элемента**

У атрибута булево значение – true (использовать), false (не использовать – это значение по умолчанию)

**Таблица №1**

| **XML атрибут** | **JAVA константа** | **Действие** |
| --- | --- | --- |
| android:layout\_centerInParent | RelativeLayout.CENTER\_IN\_PARENT | Элемент располагается в центре родительского элемента по горизонтали и вертикали |
| android:layout\_centerHorizontal | RelativeLayout.CENTER\_HORIZONTAL | Элемент располагается в центре родительского элемента по горизонтали |
| android:layout\_centerVertical | RelativeLayout.CENTER\_VERTICAL | Элемент располагается в центре родительского элемента по вертикали |
| android:layout\_alignParentBottom | RelativeLayout.ALIGN\_PARENT\_BOTTOM | Нижняя граница элемента располагается на нижней границе контейнера RelativeLayout |
| android:layout\_alignParentLeft | RelativeLayout.ALIGN\_PARENT\_LEFT | Левая граница элемента располагается на левой границе  контейнера RelativeLayout |
| android:layout\_alignParentRight | RelativeLayout.ALIGN\_PARENT\_RIGHT | Правая граница элемента располагается на правой границе контейнера RelativeLayout |
| android:layout\_alignParentTop | RelativeLayout.ALIGN\_PARENT\_TOP | Верхняя граница элемента располагается на верхней границе контейнера RelativLayout |
| android:layout\_alignWithParentIfMissing |  | Если указанный элемент (см. таблицу ниже) не найден, то для выравнивания будет использоваться родительский контейнер компоновки |

**Расположение относительно других элементов**

Значением атрибута является id другого элемента

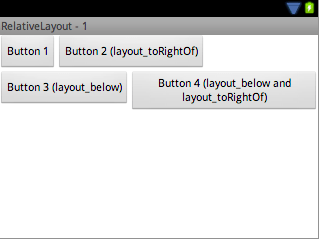
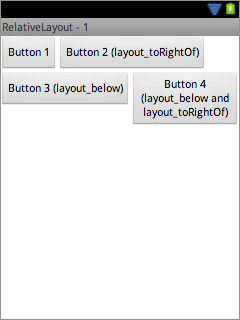
* @id/name\_id – если id или элемент с этим id уже определен в макете
* @+id/name\_id – если id не определён. Плюс указывает на создание id.
* R.id.name\_id – обращение к id из JAVA

**Таблица №2**

| **XML атрибут** | **JAVA константа** | **Действие** |
| --- | --- | --- |
| android:layout\_above | RelativeLayout.ABOVE | Расположить элемент над указанным ID элемента |
| android:layout\_below | RelativeLayout.BELOW | Расположить элемент под указанным ID элемента |
| android:layout\_toLeftOf | RelativeLayout.LEFT\_OF | Расположить элемент слева от указанного ID элемента |
| android:layout\_toRightOf | RelativeLayout.RIGHT\_OF | Расположить элемент справа от указанного ID элемента |
| android:layout\_alignBottom | RelativeLayout.ALIGN\_BOTTOM | Нижняя граница элемента выравнивается по нижней границе другого элемента |
| android:layout\_alignLeft | RelativeLayout.ALIGN\_LEFT | Левая граница элемента выравнивается по левой границе другого элемента |
| android:layout\_alignRight | RelativeLayout.ALIGN\_RIGHT | Правая граница элемента выравнивается по правой границе другого элемента |
| android:layout\_alignTop | RelativeLayout.ALIGN\_TOP | Верхняя граница элемента выравнивается по верхней границе другого элемента |
| android:layout\_alignBaseline | RelativeLayout.ALIGN\_BASELINE | Базовая линия элемента выравнивается по базовой линии другого элемента |

**Приступим к примерам**

Рассмотрим как можно расположить элементы относительно других элементов.  
Для начала картинка того, что у нас получится:



**Пример 1**

<?xml version="1.0" encoding="utf-8"?>

<**RelativeLayout** **xmlns:android**=**"http://schemas.android.com/apk/res/android"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:id**=**"@+id/button1"**

**android:text**=**"Button 1"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:id**=**"@+id/button2"**

**android:layout\_toRightOf**=**"@id/button1"**

**android:text**=**"Button 2 (layout\_toRightOf)"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:id**=**"@+id/button3"**

**android:layout\_below**=**"@id/button1"**

**android:text**=**"Button 3 (layout\_below)"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:id**=**"@+id/button4"**

**android:layout\_below**=**"@id/button1"**

**android:layout\_toRightOf**=**"@id/button3"**

**android:text**=**"Button 4 (layout\_below and layout\_toRightOf)"**/>

</**RelativeLayout**>

Разбираем код.   
  
Для начала мы создаём кнопку **Button 1** и присваиваем ей id **button1**.   
  
Кнопку **Button 2** мы хотим разместить правее кнопки **Button 1** и для этого смотрим в таблицу №2 ([Расположить элемент справа](https://devpad.ru/post/item/5/android-obzor-relativelayout.html#layout_toRightOf)) и указываем у кнопки **Button 2** данный атрибут и id 1-ой кнопки.

android:layout\_toRightOf=**"@id/button1"**

Теперь вторая кнопка у нас справа.  
  
Третью кнопку **Button 3** мы хотим разместить под кнопкой **Button 1**. Опять смотрим в таблички. Ага, [расположить элемент под указанным id](https://devpad.ru/post/item/5/android-obzor-relativelayout.html#layout_below) - это атрибут layout\_below

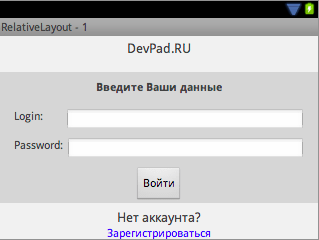
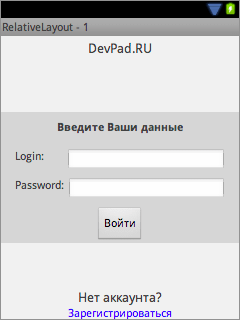
android:layout\_below=**"@id/button1"**

  4-ая кнопка будет располагаться правее кнопки **Button 3** и ниже первой кнопки.

android:layout\_below=**"@id/button1"**

android:layout\_toRightOf=**"@id/button3"**

Создадим макет поинтереснее, например, форму авторизации.



**Пример 2**

<?xml version="1.0" encoding="utf-8"?>

<**RelativeLayout** **xmlns:android**=**"http://schemas.android.com/apk/res/android"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**

**android:background**=**"#eee"**

>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerHorizontal**=**"true"**

**android:layout\_marginTop**=**"5dp"**

**android:textSize**=**"17sp"**

**android:text**=**"DevPad.RU"**/>

<**RelativeLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerInParent**=**"true"**

**android:background**=**"#ccc"**

**android:paddingTop**=**"10dp"**

>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerHorizontal**=**"true"**

**android:id**=**"@+id/title"**

**android:layout\_marginBottom**=**"20dp"**

**android:textStyle**=**"bold"**

**android:text**=**"Введите Ваши данные"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_marginLeft**=**"20dp"**

**android:layout\_marginRight**=**"30dp"**

**android:id**=**"@+id/text\_login"**

**android:layout\_below**=**"@id/title"**

**android:text**=**"Login:"**/>

<**EditText** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"30dp"**

**android:layout\_marginRight**=**"20dp"**

**android:layout\_below**=**"@id/title"**

**android:layout\_toRightOf**=**"@id/text\_login"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_marginTop**=**"20dp"**

**android:layout\_marginLeft**=**"20dp"**

**android:layout\_marginRight**=**"5dp"**

**android:id**=**"@+id/text\_password"**

**android:layout\_below**=**"@id/text\_login"**

**android:text**=**"Password:"**/>

<**EditText** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"30dp"**

**android:layout\_marginRight**=**"20dp"**

**android:layout\_marginTop**=**"20dp"**

**android:layout\_below**=**"@id/text\_login"**

**android:layout\_toRightOf**=**"@id/text\_password"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Войти"**

**android:layout\_margin**=**"20dp"**

**android:id**=**"@+id/btn\_enter"**

**android:layout\_centerHorizontal**=**"true"**

**android:layout\_below**=**"@id/text\_password"**/>

</**RelativeLayout**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentBottom**=**"true"**

**android:layout\_centerHorizontal**=**"true"**

**android:textSize**=**"15sp"**

**android:textColor**=**"#00f"**

**android:id**=**"@+id/reg\_link"**

**android:text**=**"Зарегистрироваться"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerHorizontal**=**"true"**

**android:layout\_above**=**"@id/reg\_link"**

**android:textSize**=**"17sp"**

**android:text**=**"Нет аккаунта?"**/>

</**RelativeLayout**>

Разбираемся.  
Вначале мы заливаем экран светло серым фоном, а затем добавляем заголовок к нашей форме:

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerHorizontal**=**"true"**

**android:layout\_marginTop**=**"5dp"**

**android:textSize**=**"17sp"**

**android:text**=**"DevPad.RU"**/>

Обратите внимание на [android:layout\_centerHorizontal](https://devpad.ru/post/item/5/android-obzor-relativelayout.html" \l "layout_centerHorizontal)="true" – это обозначает, что нужно выровнять текст по центру горизонтальной направленности. В разметке LinearLayout для этого мы бы использовали android:layout\_gravity="center\_horizontal".  
  
Далее мы создадим ещё один контейнер разметки для того, чтобы выровнять его по центру экрана. Для этого мы воспользуемся атрибутом [android:layout\_centerInParent](https://devpad.ru/post/item/5/android-obzor-relativelayout.html" \l "layout_centerInParent). Чтобы визуально выделить этот фрагмент, мы зададим ему фоновый серый цвет.

<**RelativeLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerInParent**=**"true"**

**android:background**=**"#ccc"**

**android:paddingTop**=**"10dp"**

>

Теперь по уже знакомому сюжету мы добавляем текст по середине экрана в горизонтальном направлении и присваиваем ему id равное **title**. Под ним с помощью android:layout\_below="@id/title" располагаем текст «Login:» и устанавливаем ему id android:id="@+id/text\_login". Правее текста располагаем поле текстового ввода. Для это используем android:layout\_below="@id/title" иandroid:layout\_below="@id/title"и  android:layout\_toRightOf="@id/text\_login". По аналогии делаем тоже самое с полем ввода для пароля. Под вводом пароля по середине экрана разместим кнопку входа. Для этого зададим ей уже знакомый атрибут android:layout\_centerHorizontal="true" и укажем, что кнопка должна быть после ввода пароля android:layout\_below="@id/text\_password". Всё, вложенный контейнер RelativeLayout закрываем.  
Нам остаётся сейчас разместить внизу экрана ссылку на регистрацию. Смотрим в таблицы и находим, как можно расположить элемент относительно контейнера [в нижней границе](https://devpad.ru/post/item/5/android-obzor-relativelayout.html#layout_alignParentBottom).

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentBottom**=**"true"**

**android:layout\_centerHorizontal**=**"true"**

**android:textSize**=**"15sp"**

**android:textColor**=**"#00f"**

**android:id**=**"@+id/reg\_link"**

**android:text**=**"Зарегистрироваться"**/>

В итоге имеем текст написанный синим цветом внизу экрана по середине. Его можно сделать  кликабельным, но это не входит в тему этого поста.   
Хорошо бы ещё что-то написать поверх ссылки зарегистрироваться. Для этого есть [атрибут](https://devpad.ru/post/item/5/android-obzor-relativelayout.html#layout_above), позволяющий расположить элемент над указанным элементом – layout\_above.

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

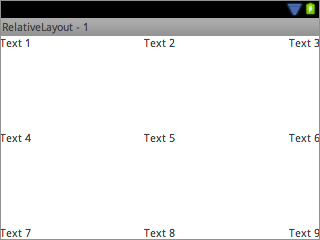
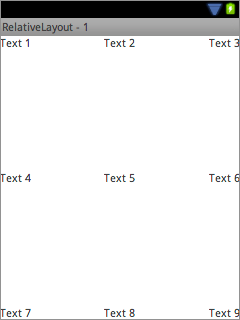
**android:layout\_centerHorizontal**=**"true"**

**android:layout\_above**=**"@id/reg\_link"**

**android:textSize**=**"17sp"**

**android:text**=**"Нет аккаунта?"**/>

Вот таким нехитрым примером мы и разобрали некоторые атрибуты контейнера.  
  
  
А сейчас мы рассмотрим расположение элементов относительно родительского элемента (см. [таблица 1](https://devpad.ru/post/item/5/android-obzor-relativelayout.html##table1)).  
Расположим элементы по всем краям контейнера.



**Пример 3**

<?xml version="1.0" encoding="utf-8"?>

<**RelativeLayout** **xmlns:android**=**"http://schemas.android.com/apk/res/android"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentLeft**=**"true"**

**android:text**=**"Text 1"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentTop**=**"true"**

**android:layout\_centerHorizontal**=**"true"**

**android:text**=**"Text 2"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentTop**=**"true"**

**android:layout\_alignParentRight**=**"true"**

**android:text**=**"Text 3"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentLeft**=**"true"**

**android:layout\_centerVertical**=**"true"**

**android:text**=**"Text 4"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerInParent**=**"true"**

**android:text**=**"Text 5"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentRight**=**"true"**

**android:layout\_centerVertical**=**"true"**

**android:text**=**"Text 6"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentBottom**=**"true"**

**android:text**=**"Text 7"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_centerHorizontal**=**"true"**

**android:layout\_alignParentBottom**=**"true"**

**android:text**=**"Text 8"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_alignParentBottom**=**"true"**

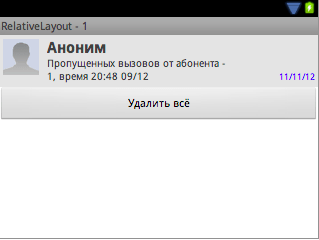
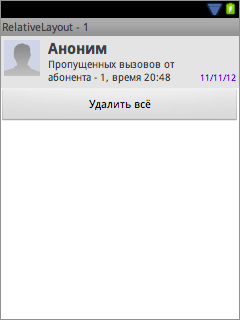
**android:layout\_alignParentRight**=**"true"**

**android:text**=**"Text 9"**

/>

</**RelativeLayout**>

Отдельных пояснений по примеру не будет. Если смотреть на пример, на результат в виде скриншота и на таблицу с описание атрибутов, то всё должно стать на свои места.  
  
В посте «[Всё о LinearLayout – 1](http://devpad.ru/post/item/2/android-vse-o-linearlayout-1.html)» мы рассматривали один из примеров с [макетом сообщения](http://devpad.ru/post/item/2/android-vse-o-linearlayout-1.html#linearlayout_message_example).  
Давайте теперь сделаем тоже самое, но уже используя разметку RelativeLayout.



**Пример 4**

<?xml version="1.0" encoding="utf-8"?>

<**RelativeLayout** **xmlns:android**=**"http://schemas.android.com/apk/res/android"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**

>

<**RelativeLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:paddingTop**=**"5dp"**

**android:paddingBottom**=**"5dp"**

**android:background**=**"#ddd"**

**android:id**=**"@+id/list"**

>

<**ImageView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_gravity**=**"center"**

**android:layout\_marginRight**=**"10dp"**

**android:layout\_marginLeft**=**"5dp"**

**android:scaleType**=**"fitCenter"**

**android:src**=**"@drawable/anonym"**

**android:id**=**"@+id/contact\_image"**

/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Аноним"**

**android:textSize**=**"20sp"**

**android:textStyle**=**"bold"**

**android:layout\_toRightOf**=**"@id/contact\_image"**

**android:id**=**"@+id/contact\_name"**

/>

<**LinearLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_toRightOf**=**"@id/contact\_image"**

**android:layout\_below**=**"@id/contact\_name"**

>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Пропущенных вызовов от абонента - 1, время 20:48 09/12"**

**android:layout\_weight**=**"0.9"**

**android:textSize**=**"13sp"**

**android:maxLines**=**"2"**/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_weight**=**"1.9"**

**android:layout\_gravity**=**"bottom"**

**android:gravity**=**"right"**

**android:paddingRight**=**"5dp"**

**android:text**=**"11/11/12"**

**android:textSize**=**"12sp"**

**android:textColor**=**"#00f"**

/>

</**LinearLayout**>

</**RelativeLayout**>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Удалить всё"**

**android:gravity**=**"center"**

**android:layout\_below**=**"@id/list"**

/>

</**RelativeLayout**>

В этом коде, мы поместили в корневой контейнер, контейнер RelativeLayout и задали ему серый фоновый цвет.  
Внутрь контейнера мы поместили изображение контакта. Правее изображения (android:layout\_toRightOf="@id/contact\_image"), мы поместили имя контакта. Под именем контакта и правее изображения контакта, мы поместили другой контейнер компоновки LinearLayout.

android:layout\_height=**"wrap\_content"**

android:layout\_toRightOf=**"@id/contact\_image"**

android:layout\_below=**"@id/contact\_name"**

Этот контейнер хорош, когда нужно пропорционально распределить элементы внутри при помощи android:layout\_weight. В контейнере RelativeLayout к сожалению нет данного атрибута. Внутрь контейнера добавляем 2 текстовых элемента.

Обратите внимание, что внутри этого контейнера LinearLayout, уже не работают атрибуты из RelativeLayoutи вам необходимо использовать для вложенных элементов только разрешённые атрибуты его "родителя"

Ну и на последок добавим под сообщением кнопку "Удалить всё". Для этого разместим её после вложенного контейнера RelativeLayout и укажем на его id.

android:layout\_below=**"@id/list"**

**Программное создание компоновки на JAVA**

Создание компоновки мало чем отличается от варианта с LinearLayout.  
Однако, чтобы задать элементам относительное расположение, нужно воспользоваться вложенным классом RelativeLayout.LayoutParams и его методом addRule. Первым параметром указывается константа из [таблиц, находящихся вначале поста](https://devpad.ru/post/item/5/android-obzor-relativelayout.html#table1) и если нужно указать id элемента, то его добавляют вторым параметром.

Рассмотрим пример:

Button button1 = **new** Button(**this**);

RelativeLayout.LayoutParams buttonParam1 = **new** RelativeLayout.LayoutParams(

ViewGroup.LayoutParams.WRAP\_CONTENT,

ViewGroup.LayoutParams.WRAP\_CONTENT

);

button1.setText(**"Button 1"**);

button1.setId(BUTTON\_ID\_1);

button1.setLayoutParams(buttonParam1);

Button button2 = **new** Button(**this**);

RelativeLayout.LayoutParams buttonParam2 = **new** RelativeLayout.LayoutParams(

ViewGroup.LayoutParams.WRAP\_CONTENT,

ViewGroup.LayoutParams.WRAP\_CONTENT

);

buttonParam2.addRule(RelativeLayout.BELOW, button1.getId());

buttonParam2.addRule(RelativeLayout.ALIGN\_PARENT\_RIGHT);

button2.setText(**"Button 2"**);

button2.setLayoutParams(buttonParam2);

RelativeLayout relativeLayout = **new** RelativeLayout(**this**);

RelativeLayout.LayoutParams param = **new** RelativeLayout.LayoutParams(

ViewGroup.LayoutParams.FILL\_PARENT,

ViewGroup.LayoutParams.FILL\_PARENT

);

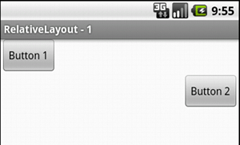
relativeLayout.setLayoutParams(param);

relativeLayout.addView(button1);

relativeLayout.addView(button2);

setContentView(relativeLayout);

В результате у нас получилось следующее:



В начале мы создали 2 кнопки. 2-ой кнопке мы применили правила расположиться после первой кнопки и по правому краю контейнера RelativeLayout.

RelativeLayout.LayoutParams buttonParam2 = **new** RelativeLayout.LayoutParams(

ViewGroup.LayoutParams.WRAP\_CONTENT,

ViewGroup.LayoutParams.WRAP\_CONTENT

);

buttonParam2.addRule(RelativeLayout.BELOW, button1.getId());

buttonParam2.addRule(RelativeLayout.ALIGN\_PARENT\_RIGHT);

После задания параметров у кнопок, мы создаём корневой контейнер RelativeLayout и добавляем в него эти кнопки

RelativeLayout relativeLayout = **new** RelativeLayout(**this**);

RelativeLayout.LayoutParams param = **new** RelativeLayout.LayoutParams(

ViewGroup.LayoutParams.FILL\_PARENT,

ViewGroup.LayoutParams.FILL\_PARENT

);

relativeLayout.setLayoutParams(param);

relativeLayout.addView(button1);

relativeLayout.addView(button2);

После чего отображаем компоновку на экране

setContentView(relativeLayout);

На этом всё. Спасибо за внимание.  
  
Исходники примеров на гитхабе - <https://github.com/devpad/Android-RelativeLayout-1>