

## LANGUAGE







<u>Присоединяйтесь</u> к Jabber чату StartAndroid

# ВАКАНСИИ РАЗРАБОТЧИКС

На нашем форуме рекрутеры периодически пополняют <u>список</u> <u>вакансий</u> Android разработчиков. Будет время загляните, возможно вас

### МАТЕРИАЛЫ ПО СМЕЖНЫМ ТЕМАМ

- Урок 9. Обработчики событий на примере Button.
- Урок 10. Оптимизируем реализацию обработчиков.
- Урок 16. Программное создание экрана. LayoutParams
- Урок 18. Меняем layoutParams в рабочем приложении
- Урок 40. LayoutInflater. Учимся использовать.
- Урок 120. Виджеты. Обработка нажатий



В этом уроке мы:

- добавляем компоненты на экран прямо из приложения

заинтересуют эти предложения.



## ПИАР-ПОДДЕРЖКА

Вы создали приложение/ проект/стартап, о котором хотите рассказать? У меня к вам есть предложение. На прошлом уроке мы создавали компоненты в методе Activity.onCreate, т.е. при создании приложения. На этом уроке будем создавать уже в работающем приложении. Создавать будем Button-ы, т.к. они наглядней всего отображаются. Будем указывать текст, который будет отображен на кнопке и выравнивание: слева, по центру или справа. Также предусмотрим возможность удаления созданных элементов.

Создадим проект:

Project name: P0171\_DynamicLayout2
Build Target: Android 2.3.3
Application name: DynamicLayout2
Package name: ru.startandroid.develop.dynamiclayout2
Create Activity: MainActivity

Создадим экран, который поможет нам создавать View-компоненты. Открываем **main.xml** и пишем там следующее:

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

	<linearlayout< th=""></linearlayout<>
	xmlns:android="http://schemas.android.com/apk/res/android"
	android:orientation="vertical"
	android:layout_width=" <i>fill_parent</i> "
	android:layout_height="fill_parent">
	<radiogroup< th=""></radiogroup<>
	android:layout_height=" <i>wrap_content</i> "
	android:layout_width="match_parent"
	android:orientation="horizontal"
	android:id="@+ <i>id/rgGravity</i> ">
	<radiobutton< th=""></radiobutton<>
	android:layout_height=" <i>wrap_content</i> "
	android:layout_width="wrap_content"
	android:checked=" <i>true</i> "
	android:text=" <i>Left</i> "
	android:id="@+ <i>id/rbLeft</i> ">
ì	<radiobutton< th=""></radiobutton<>
	android:layout_height=" <i>wrap_content</i> "
1	android:layout_width=" <i>wrap_content</i> "
	android:text="Center"
	android:id="@+ <i>id/rbCenter</i> ">
er	
1	<radiobutton< td=""></radiobutton<>
36	android:layout_height=" <i>wrap_content</i> "
ic	android:layout_width=" <i>wrap_content</i> "
2	android:text=" <i>Right</i> "

ПОДДЕРЖКА ПРОЕКТА

#### <u>Яндекс</u> **410011180491924**

Alfa-Bank 5486734918678877

**WebMoney** 

R248743991365 Z551306702056

## ePayService D434155

PayPal



א RSS



Сообщение сер Данный сайт з

http://vk.com/wie app=0&width=2

Урок 17. Создание View-компонент в рабочем приложении

	android:id="@+ <i>id/rbRight</i> ">						
	<linearlayout< th=""><th></th></linearlayout<>						
	android:id="@+ <i>id/linearLayout1</i> "						
	android:layout_width=" <i>match_parent</i> "						
<u> </u>	android:orientation="horizontal"						
	android:layout_height=" <i>wrap_content</i> ">						
	<edittext< th=""><th></th></edittext<>						
FOLLOW US ON CU	android:layout_height=" <i>wrap_content</i> "						
	android:layout_width="wrap_content"						
	android:layout_weight="1"						
	android:id="@+id/etName"						
Сборник уро	android:fadeScrollbars=" <i>true</i> ">						
3	<button< td=""><td></td></button<>						
	android:layout_width=" <i>wrap_content</i> "						
	android:layout_height=" <i>wrap_content</i> "						
	android:text="Create"						
	android:id="@+ <i>id/btnCreate</i> ">						
	<button< th=""><th></th></button<>						
	android:layout_width=" <i>wrap_content</i> "						
Google And	android:layout_height=" <i>wrap_content</i> "						
это несл	android:text=" <i>Clear</i> "						
	android:id="@+ <i>id/btnClear</i> ">						
Рекламное м	<linearlayout< th=""><th></th></linearlayout<>						
своболн	android:layout_width=" <i>match_parent</i> "						
oboodii	android:layout_height=" <i>match_parent</i> "						
	android:id="@+ <i>id/llMain</i> "						
	android:orientation="vertical">						



Рассмотрим подробно экран.

rgGravity – это RadioGroup, с тремя RadioButton (rbLeft, rbCenter, rbRight). Этот компонент мы используем для выбора выравнивания создаваемого компонента etName – текстовое поле, здесь будем указывать текст, который будет отображаться на созданном компоненте

btnCreate – кнопка, запускающая процесс создания.

btnClear – кнопка, стирающая все, что создали

IlMain – вертикальный LinearLayout, в котором будут создаваться компоненты

#### Яндекс.Директ



## Аккумулятор для BP-6M

АКБ для любой модели. От 69 рублей. Узнай на сайте!

<u>Контакты</u> <u>Доставка</u>

<u>Новинки</u>

la-bora.ru Адрес и телефон



# <u>Хостинг</u> Будущего Евробайт

Невероятная скорость SSD. Непревзойденная стабильность. Душевная поддержка.

<u>Месяц</u>

бесплатно VPS/VDS хостинг Битрикс хостинг О компании Евробайт eurobyte.ru



<u>Закажи</u> платный хостинг за 50 руб.

Поддержка 24/7, домен RU или РФ в подарок, 30 дней бесплатно. Заказывайте!



Экран готов, давайте кодить реализацию. Открываем **MainActivity.java**. Начнем с того, что опишем и найдем все необходимые нам компоненты. Кстати, у нас есть пара кнопок, которые мы будем использовать, значит им нужен обработчик. В качестве **обработчика** назначим **Activity** (т.е. необходимо дописать: implements OnClickListener) и создадим пустой пока метод обработки **onClick**:

public class MainActivity extends Activity implements OnClickListener{

LinearLayout llMain; RadioGroup rgGravity; EditText etName; Button btnCreate; Button btnClear;

int wrapContent = LinearLayout.LayoutParams.WRAP\_CONTENT;

/\*\* Called when the activity is first created. \*/
@Override
public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.main);

llMain = (LinearLayout) findViewById(R.id.llMain);

#### 28.07.2015

<u>Домены RU,</u> <u>РФ 120 руб</u> <u>Мощные VDS</u> <u>- 500 руб</u> <u>VDS Windows</u> <u>Облачный диск</u> radiushost.ru Адрес и телефон

```
Урок 17. Создание View-компонент в рабочем приложении
```

```
rgGravity = (RadioGroup) findViewById(R.id.rgGravity);
etName = (EditText) findViewById(R.id.etName);
btnCreate = (Button) findViewById(R.id.btnCreate);
btnCreate.setOnClickListener(this);
btnClear = (Button) findViewById(R.id.btnClear);
btnClear.setOnClickListener(this);
}
@Override
public void onClick(View v) {
 // TODO Auto-generated method stub
}
```

Я также создал переменную **wrapContent** и буду хранить в ней значение LinearLayout.LayoutParams.WRAP\_CONTENT. Делаю это только для снижения громоздкости кода.

Теперь опишем процесс создания Button-компонента заполнив метод onClick:

```
@Override
 public void onClick(View v) {
    switch (v.getId()) {
    case R.id.btnCreate:
      // Создание LayoutParams с шириной и высотой по содержимому
      LinearLayout.LayoutParams lParams = new LinearLayout.LayoutParams(
          wrapContent, wrapContent);
      // переменная для хранения значения выравнивания
      // по умолчанию пусть будет LEFT
      int btnGravity = Gravity.LEFT;
      // определяем, какой RadioButton "чекнут" и
      // соответственно заполняем btnGravity
      switch (rgGravity.getCheckedRadioButtonId()) {
     case R.id.rbLeft:
       btnGravity = Gravity.LEFT;
       break;
      case R.id.rbCenter:
       btnGravity = Gravity.CENTER_HORIZONTAL;
       break;
      case R.id.rbRight:
       btnGravity = Gravity.RIGHT;
        break;
      }
      // переносим полученное значение выравнивания в LayoutParams
      lParams.gravity = btnGravity;
      // создаем Button, пишем текст и добавляем в LinearLayout
      Button btnNew = new Button(this);
      btnNew.setText(etName.getText().toString());
      llMain.addView(btnNew, lParams);
      break;
   }
```

}

Разберем написанное. Для начала мы проверяем, что была нажата кнопка **btnCreate** – т.е. кнопка создания. Затем создаем **LayoutParams** с высотой и шириной по содержанию. Здесь я использовал переменную, про которую писал выше – wrapContent. Иначе получилось бы довольно громоздко.

Далее создаем переменную **btnGravity**, в которую по умолчанию запишем значение выравнивания LEFT. Для определения, какой RadioButton выделен в данный момент, используем метод **getCheckedRadioButtonId** – он для RadoiGroup возвращает ID «чекнутого» RadioButton-а. Мы его сравниваем с нашими тремя ID и заносим соответствующее значение в переменную btnGravity. Скидываем это значение в gravity y LayoutParams.

Далее создаем кнопку и присваиваем ей текст из **etName**. Обратите внимание, что недостаточно написать getText, т.к. это не даст текста. Необходимо еще вызвать метод toString. Ну и в конце добавляем созданный Button в наш LinearLayout.

	5554:AVD_233	and a			5		da da s				
I	DynamicLayout2		ii al	3:33							
I	Left Center Right										6
1	btn3		Create	Clear						Σ	Z
1	btn1							•		7	~
1		btn2								IENU	
1				btn3							
1						1 !	2 <sup>@</sup>	3 #	4 \$	5 <sup>%</sup>	6
						Q	w~	Ε″	R	т {	Y
						А	s`	D	F	G]	H
						슘	Ζ	Х	С	V	В
						ALT	SYM	0			_

Сохраним все и запустим приложение. Добавим несколько кнопок.

Кнопки должны появляться с указанным выравниванием и текстом.

Когда вводите текст, снизу появляется клавиатура и закрывает обзор. Чтобы она исчезла, надо нажать кнопку Back (Назад) на эмуляторе или ESC на обычной клавиатуре. Если клавиатура появляется японская с иероглифами, вызовите контекстное меню для поля ввода (долгое нажатие левой кнопкой мыши), нажмите Input method и выберите из списка Android Keyboard.

Осталось нереализованной кнопка **Clear**, которая призвана удалять все созданное. Для этого нам необходимо дополнить метод **onClick**, добавим в **switch** (v.getId()) еще один case:

```
case R.id.btnClear:
    llMain.removeAllViews();
    Toast.makeText(this, "Удалено", Toast.LENGTH_SHORT).show();
    break;
```

Метод **removeAllViews** удаляет все дочерние View-компоненты с нашего LinearLayout. С помощью **Toast** выводим на экран **сообщение** об успехе. Сохраним, запустим и проверим. Добавляем несколько кнопок, жмем кнопку **Clear** и наблюдаем результат:

5554:AVD_233						
🗯 📶 👂 3:35 DynamicLayout2						
Left Center Right			6			6
btn3 Create Clear			6		Z	3
						1
					MENU	
	1	2 <sup>@</sup>	3 #	4 \$	5 <sup>%</sup>	6
Venerue	Q	WĨ	Ε ″	R	т {	Y
Удалено	А	s`	D	F	G ]	Н
	슣	Ζ	Х	С	V	В
	ALT	SYM	@			_

В итоге у нас получилось очень даже динамическое приложение, которое умеет менять само себя.

На форуме задают вопрос: как потом получить доступ к этим созданным компонентам. Тут есть пара простых вариантов.

1) При создании вы можете сами присваивать компонентам ID. Это делается методом <u>setId</u>. И потом по этим ID просто вызываете findViewById.

2) Вы можете сохранять созданные компоненты в свой массив или список. Либо

можете воспользоваться методом <u>getChildAt</u>. Вызов этого метода для IIMain позволит получить его дочерние компоненты по индексу. Получить кол-во дочерних элементов позволит метод <u>getChildCount</u>.

Полный код урока:

```
public class MainActivity extends Activity implements OnClickListener {
 LinearLayout llMain;
 RadioGroup rgGravity;
 EditText etName;
 Button btnCreate;
 Button btnClear;
 int wrapContent = LinearLayout.LayoutParams.WRAP_CONTENT;
 /** Called when the activity is first created. */
 @Override
 public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
   llMain = (LinearLayout) findViewById(R.id.llMain);
   rgGravity = (RadioGroup) findViewById(R.id.rgGravity);
   etName = (EditText) findViewById(R.id.etName);
   btnCreate = (Button) findViewById(R.id.btnCreate);
   btnCreate.setOnClickListener(this);
   btnClear = (Button) findViewById(R.id.btnClear);
   btnClear.setOnClickListener(this);
 }
 @Override
 public void onClick(View v) {
   switch (v.getId()) {
    case R.id.btnCreate:
      // Создание LayoutParams с шириной и высотой по содержимому
      LinearLayout.LayoutParams lParams = new LinearLayout.LayoutParams(
          wrapContent, wrapContent);
      // переменная для хранения значения выравнивания
      // по умолчанию пусть будет LEFT
      int btnGravity = Gravity.LEFT;
      // определяем, какой RadioButton "чекнут" и
      // соответственно заполняем btnGravity
      switch (rgGravity.getCheckedRadioButtonId()) {
      case R.id.rbLeft:
       btnGravity = Gravity.LEFT;
       break;
      case R.id.rbCenter:
       btnGravity = Gravity.CENTER_HORIZONTAL;
       break;
      case R.id.rbRight:
       btnGravity = Gravity.RIGHT;
       break;
      }
      // переносим полученное значение выравнивания в LayoutParams
      lParams.gravity = btnGravity;
```

```
// создаем Button, пишем текст и добавляем в LinearLayout
Button btnNew = new Button(this);
btnNew.setText(etName.getText().toString());
llMain.addView(btnNew, lParams);
break;
case R.id.btnClear:
llMain.removeAllViews();
Toast.makeText(this, "Удалено", Toast.LENGTH_SHORT).show();
break;
}
}
```

На следующем уроке:

- изменяем layout-параметры для уже существующих компонентов экрана

• Обсудить на форуме [130 replies]

<u>< Назад</u> Вперёд >





При использовании материалов сайта ссылка на <u>startandroid.ru</u> обязательна.

