TextView

TextView – элемент, который используется для как ни странно показа текста. Вероятно, самый распространенный элемент интерфейса. Иногда встречаются TextView, которые используются в качестве плоских кнопок, без видимой рамки. Часто используемые атрибуты (с приставкой **android:**)

- **text** – сам текст
- **textSize** – размер шрифта
- **textColor** – цвет текста
- **textStyle** – (normal, bold, italic) – обычный, полужирный, курсив
- **textAlignment** (API 17) – выравнивание текста внутри TextView
- **gravity** – (API 1) выравнивание текста внутри TextView

Button

Button – кнопка, мотиватор к какому-либо действию.
По сути, Button – это тот же TextVIew, но с особенным стилем отображения, и это выражается в том, что в Android класс Button – наследник класса TextView. Естественно, можно использовать все атрибуты из класса TextView.

Для того, чтобы при нажатии на кнопку что то происходило, на нее нужно навесить слушателя. Возможностей две – в коде и в xml.

В коде мы вызываем метод кнопки setOnClickListener и передаем в качестве аргумента View.OnClickListener и переопределяем его метод onClick:

```
mButton.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    //do something
  }
});
```

В xml мы добавляем к кнопке атрибут onClick и передаем туда название метода, который должен вызываться при нажатии:

```
<Button
  android:id="@+id/button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:onClick="doSomething"
  android:text="Button" />
```

Сам метод должен в качестве аргумента принимать View и иметь модификатор доступа public:

```
public void doSomething(View view) {
```

```
  //do something
}
```

Также существует вариант кнопки с изображением вместо текста. Он называется ImageButton, и в свою очередь, наследуется от ImageView. Для ImageButton можно задать изображение через атрибут src. При этом можно убрать сам фон кнопки, оставив только изображение, задав значение background как @null

CheckBox, RadioButton

Checkbox и RadioButton – элементы интерфейса, которые используются для выбора тех или иных опций, причем CheckBox'ы используются для множественного выбора, а RadioButton – для единственного, в пределах своей RadioGroup. RadioGroup – контейнер, в который следует обернуть RadioButton'ы, которые планируется выделять взаимоисключающе.

В принципе, и на CheckBox и на RadioButton можно повесить обработчик нажатия onClickListener. Однако он не дает никакой информации о состоянии view, выделена она или нет. Это нужно будет выяснять отдельно, через метод isChecked().

```
checkBox.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
    Toast.makeText(MainActivity.this, "checkbox clicked", Toast.LENGTH_SHORT)
        .show();
  }
});
```

Альтернативно, можно повесить CompoundButton.OnCheckedChangeListener() и переопределить метод
onCheckedChanged(CompoundButton compoundButton, boolean isChecked)
в котором compoundButton – нажатая view, а isChecked – состояние.

```
checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
  @Override
  public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked) {
    Toast.makeText(MainActivity.this,
                compoundButton.getText() + " " + (isChecked ? "checked" : "unchecked"),
                Toast.LENGTH_SHORT)
        .show();
  }
});
```

Этот способ хорош для CheckBox, но плохо подходит для RadioButton, так как onCheckedChanged вызовется 2 раза – для новой выделенной RadioButton и для

старой, потерявшей выделение, не говоря уже о дублировании кода.

Чтобы избежать этого, можно повесить RadioGroup.OnCheckedChangeListener() на радиогруппу. Метод для переопределения onCheckedChanged(RadioGroup radioGroup, int id), в котором id – это id выделенной радиокнопки.

```
radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
  @Override
  public void onCheckedChanged(RadioGroup radioGroup, int id) {
    RadioButton rb = radioGroup.findViewById(id);
    Toast.makeText(MainActivity.this,
                   rb.getText() +" "+ rb.isChecked(),
                   Toast.LENGTH_SHORT)
         .show();
  }
});
```