Научимся обрабатывать нажатие кнопки и узнаем, что такое обработчик.

Создадим проект:

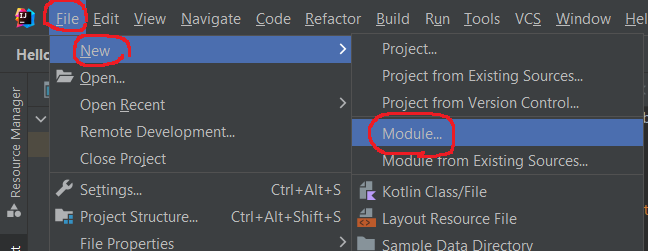


Рис. 1.

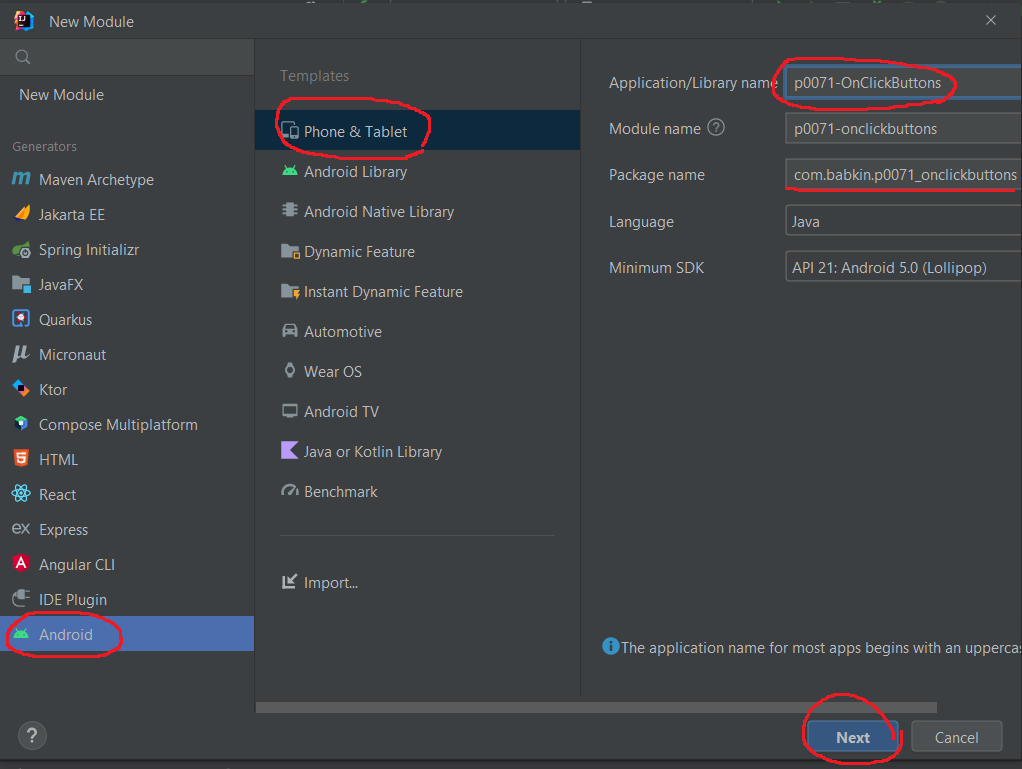


Рис. 2.

В layout файл activity\_main.xml напишем следующее и сохраним:

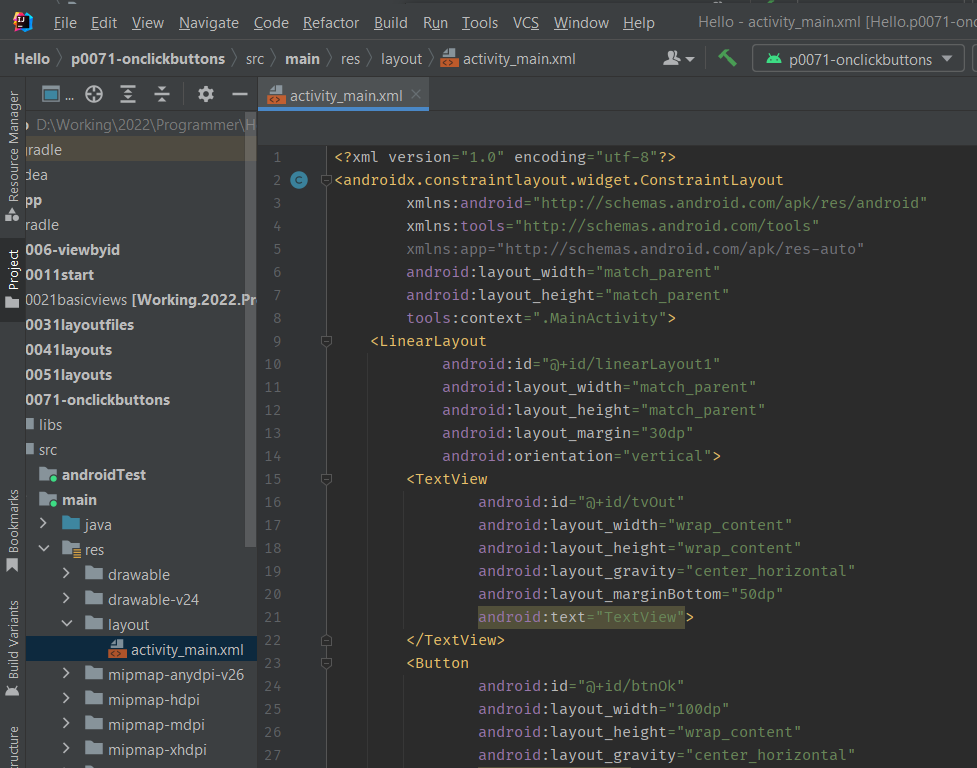


Рис. 3.

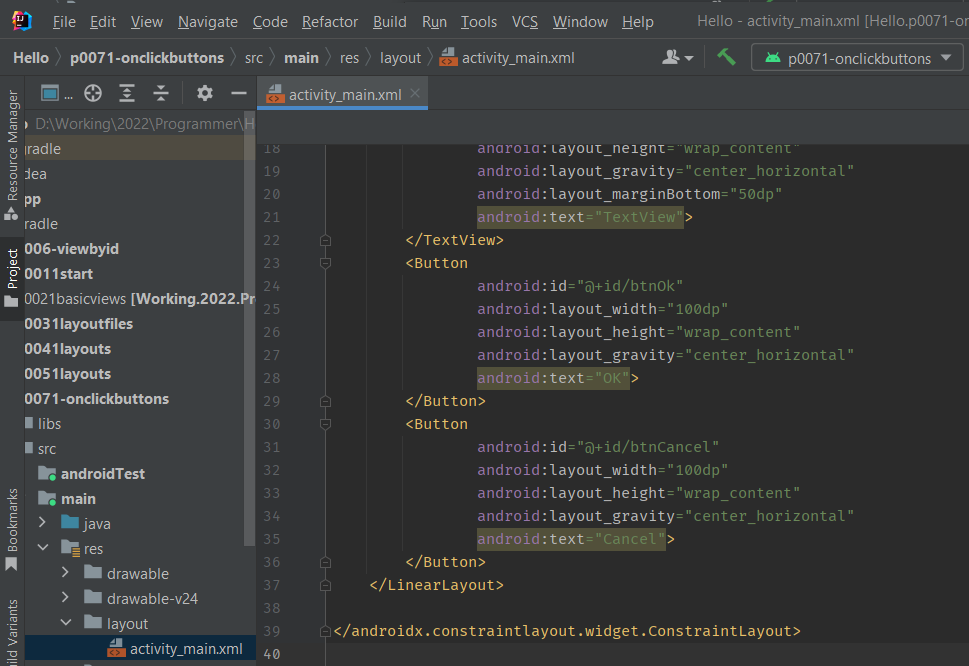


Рис. 4.

У нас есть TextView с текстом и две кнопки: OK и Cancel. Мы сделаем так, чтобы по нажатию кнопки менялось содержимое TextView. По нажатию кнопки OK – будем выводить текст: «Нажата кнопка ОК», по нажатию Cancel – «Нажата кнопка Cancel».

Открываем MainActivity.java. Описание объектов вынесем за пределы метода onCreate. Это сделано для того, чтобы мы могли из любого метода  
обращаться к ним. В onCreate мы эти объекты заполним с помощью уже пройденного нами метода findViewById. В итоге должен получиться такой код:

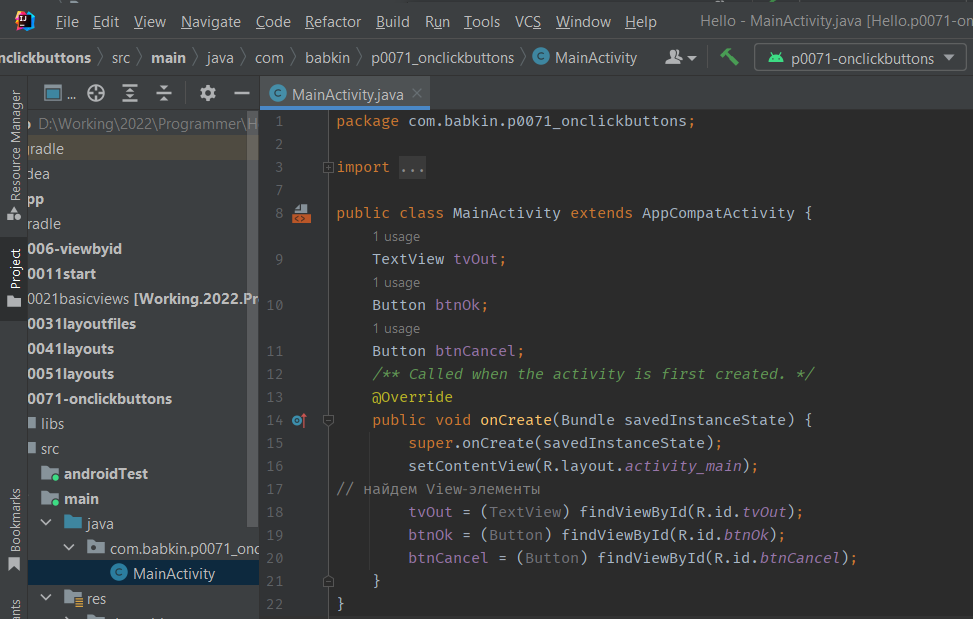


Рис. 5.

Обновляем секцию import (CTRL+SHIFT+O). Объекты tvOut, btnOk и btnCancel соответствуют View-элементам экрана и мы можем с ними работать. Нам надо научить кнопку реагировать на нажатие. Для этого у кнопки есть метод setOnClickListener. На вход подается объект с интерфейсом View.OnClickListener. Именно этому объекту кнопка поручит обрабатывать нажатия. Давайте создадим такой объект. Код продолжаем писать в onCreate:

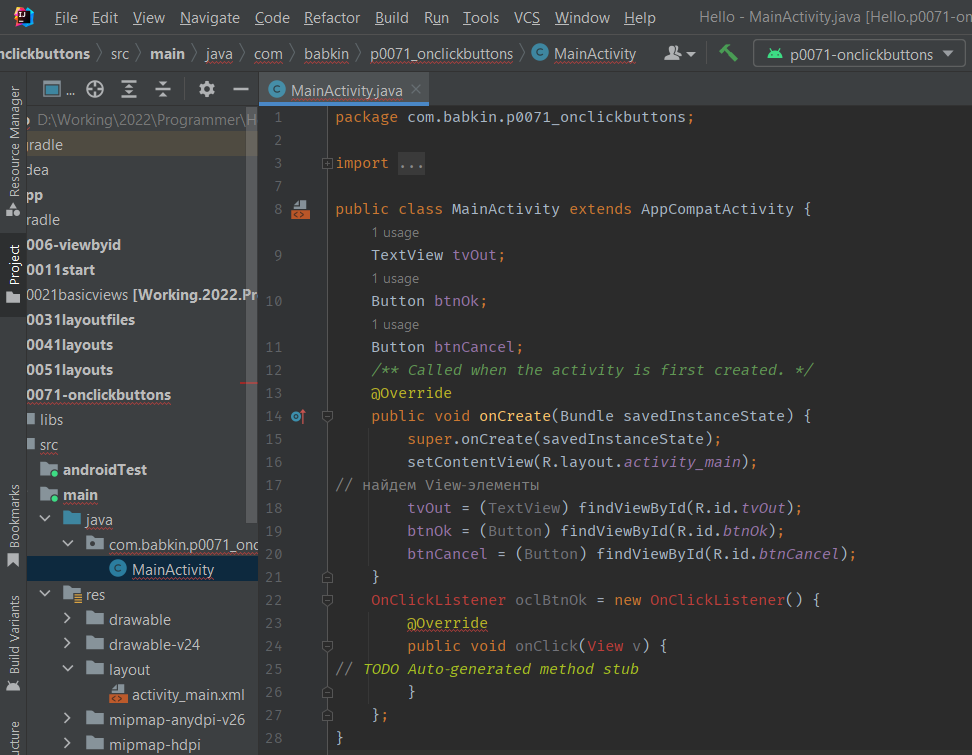


Рис. 6.

Intellij Idea подчеркивает OnClickListener красной линией т.к. пока не знает его. Необходимо обновить секцию import. Жмем ALT+ENTER, Intellij Idea показывает нам, что он знает два интерфейса с именем onClickListener и предлагает выбрать. Нам нужен View.OnClickListener, т.к. метод кнопки setOnClickListener принимает на вход именно его.

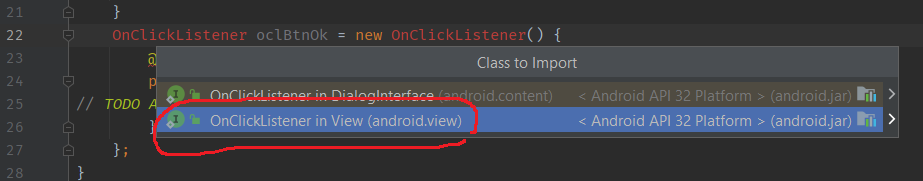


Рис. 7.

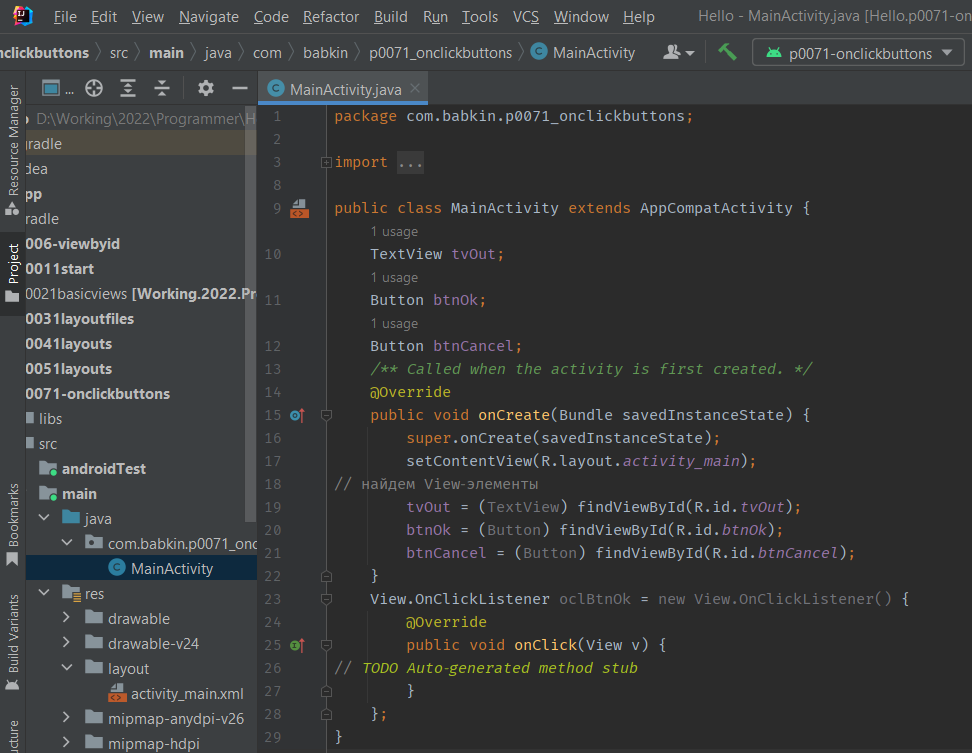


Рис. 8.

Итак, мы создали объект oclBtnOk, который реализует интерфейс View.OnClickListener. Объект содержит метод onClick – это как раз то, что нам нужно.

Именно этот метод будет вызван при нажатии кнопки. Мы решили, что по нажатию будем выводить текст: «Нажата кнопка ОК» в TextView (tvOut). Реализуем это.

В методе onClick пишем:

tvOut.setText("Нажата кнопка ОК");

Обработчик нажатия готов. Осталось «скормить» его кнопке с помощью метода setOnClickListener.

btnOk.setOnClickListener(oclBtnOk);

В итоге должен получится такой код:

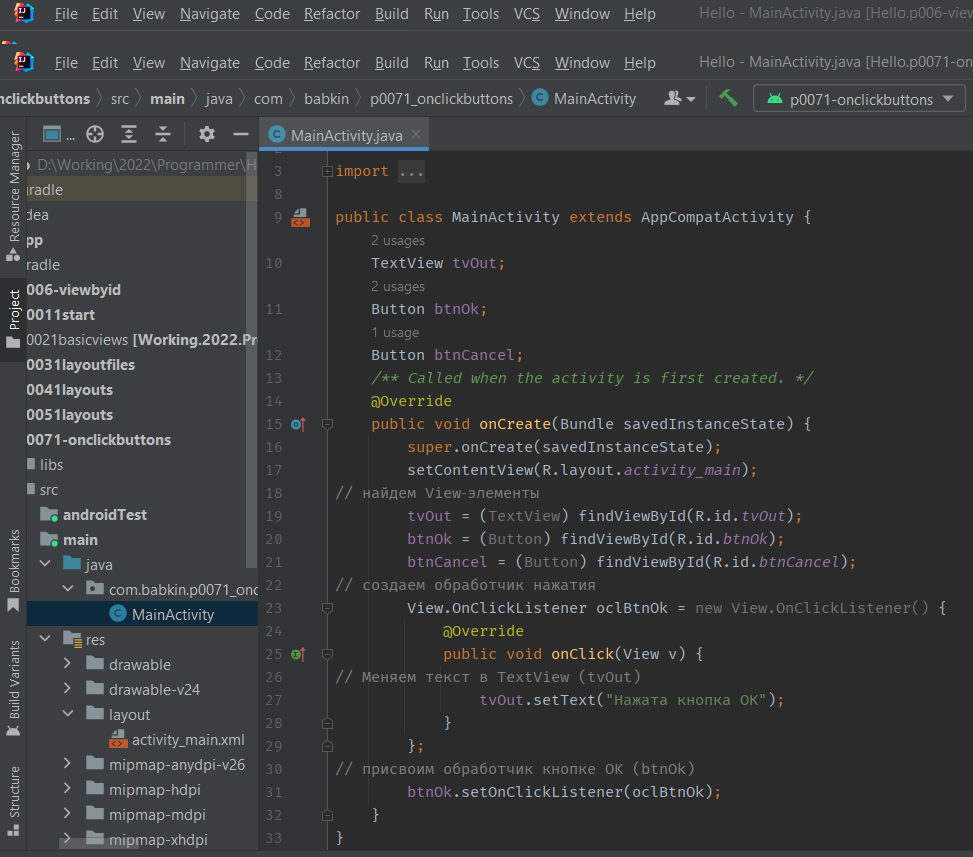


Рис. 9.

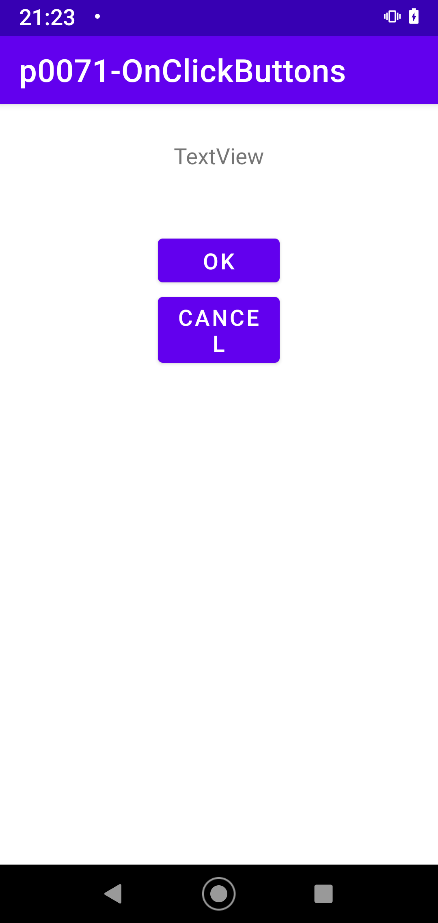


Рис. 10.

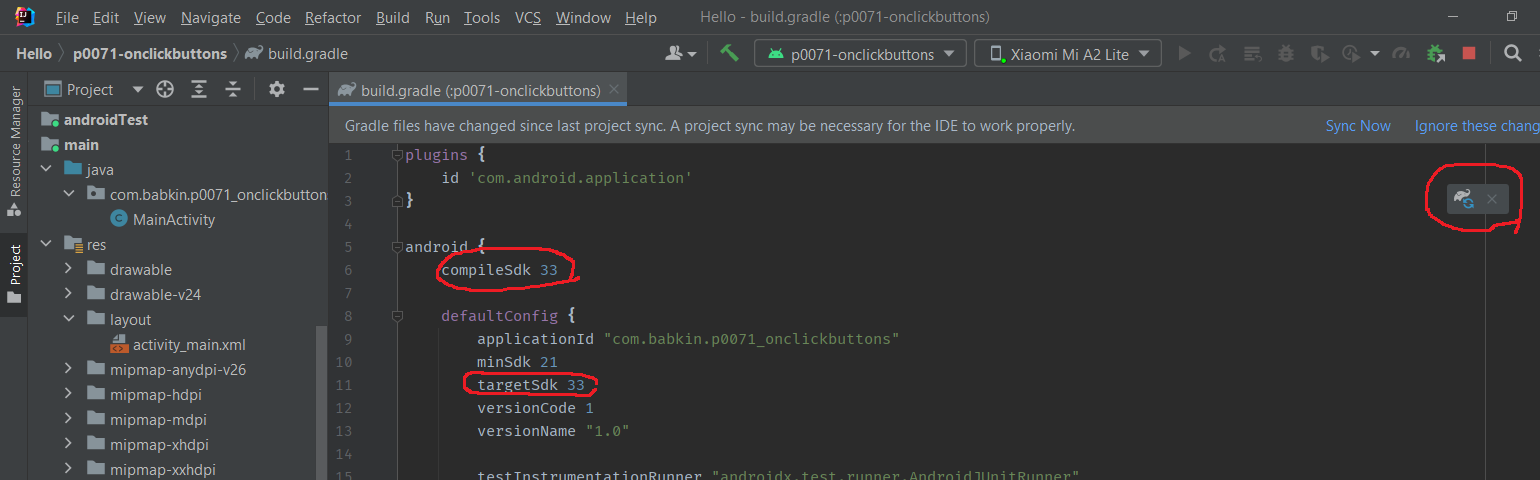


Рис. 11.

Все сохраняем и запускаем. Жмем на кнопку ОК и видим. Что текст изменился

Рис. 12.

Рис. 13.

Нажатие на Cancel пока ни к чему не приводит, т.к. для нее мы обработчик не создали и не присвоили. Давайте сделаем это аналогично, как для кнопки OK. Сначала мы создаем обработчик:

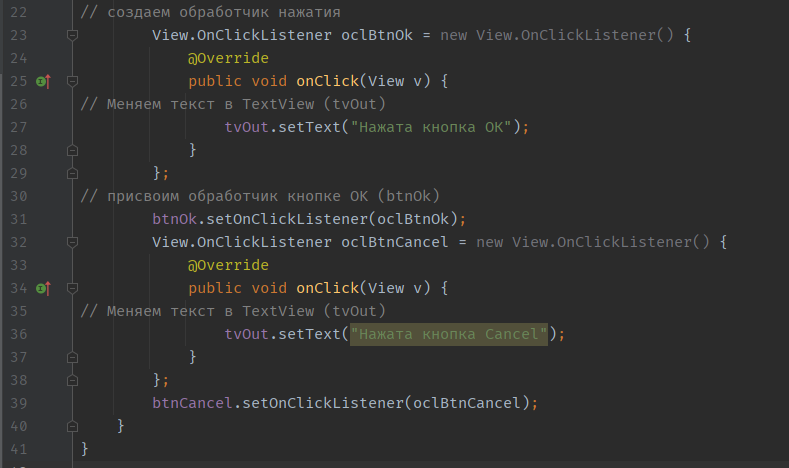


Рис. 14.

Сохраняем, запускаем, проверяем. Обе кнопки теперь умеют обрабатывать нажатия. Давайте еще раз проговорим механизм обработки событий на примере нажатия кнопки. Сама кнопка обрабатывать нажатия не умеет, ей нужен обработчик (его также называют слушателем - listener), который присваивается с помощью метода setOnClickListener. Когда на кнопку нажимают, обработчик реагирует и выполняет код из метода onClick. Это можно изобразить так:

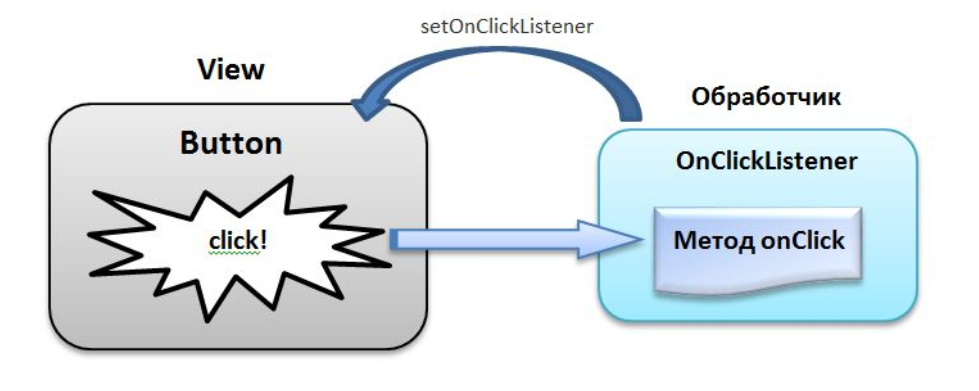


Рис. 15.

Соответственно для реализации необходимо выполнить следующие шаги:

- создаем обработчик

- заполняем метод onClick

- присваиваем обработчик кнопке

и система обработки событий готова.