## Кликер

### Цель работы

Познакомиться с основными элементами управления приложения и научиться их использовать согласно логике приложения.

### Задания для выполнения

1. Создайте простое приложение-кликер. Главное и единственное окно приложения должно демонстрировать кнопку с надписью “Нажми меня” и надпись “Вы нажали кнопку 0 раз”. При каждом нажатии на кнопку, надпись соответствующим образом изменяется.

### Методические указания

Создадим новый проект так же, как на прошлой работе. Назовем его clicker.

Откроем XML-файл с активностью. Добавим текстовое поле:

<TextView

android:id="@+id/mainTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кликов: 0"

Под ним добавим кнопку:

<Button

android:id="@+id/main\_btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/click\_me"/>

Перейдем в файл с кодом активности. Для начала введем переменные для наших элементов управления:

TextView mainText;

Button mainBtn;

Еще на понадобится переменная, содержащая значение количества тапов по кнопке. Она может быть целого или длинного типа.

private long score = 0;

Эти переменные будут являться членами класса.

Затем в методе при загрузке приложения найдем по идентификатору ссылки на соответствующие элементы управления:

mainText = (TextView) findViewById(R.id.*mainTxt*);

mainBtn = (Button) findViewById(R.id.*main\_btn*);

Теперь создадим обработчик события нажатия на кнопку. В этой работе мы попробуем создать новый класс для обработки события. Это более правильный подход, чтобы не загромождать главный класс активности всеми возможными событиями приложения. В Java можно создавать класс сразу при его объявлении по месту, объединив объявление класса с вызовом конструктора:

View.OnClickListener clickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

}

};

Обратите внимание на точку с запятой в конце объявления класса - это все-таки оператор присваивания.

Таким образом, мы создали анонимный класс-потомок класса View.OnClickListener() с одним переопределенным методом - onClick() и сразу же создали экземпляр этого класса в переменную clickListener. Это полезно, если Вы планируете использовать определение класса только один раз.

Метод onClick() должен будет вызываться при нажатии на кнопку. В определении этого метода нам нужно произвести следующие действия:

Увеличиваем значение счетчика нажатий:

score ++;

Составляем нужную строку:

String s = "Кликов: " + score;

Выводим эту строку в текстовое поле:

mainText.setText(s.toCharArray(),0, s.length());

Теперь нам осталось только назначить обработчик события нужному нам элементу управления:

mainBtn.setOnClickListener(clickListener);

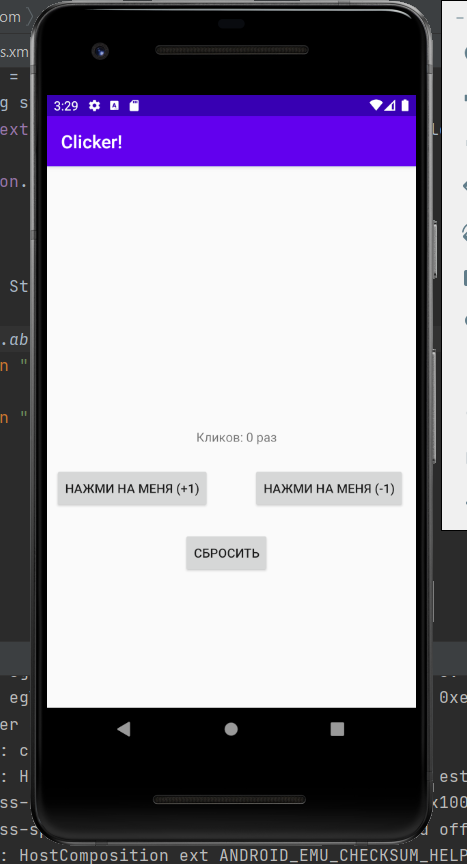
Запустим наше приложение. Мы должны увидеть работающую кнопку и счетчик нажатий:

|  |  |
| --- | --- |
|  |  |

При желании Вы можете изменить оформление или позиционирование этих элементов.

### Дополнительные задания

1. Добавьте в приложение кнопку, которая наоборот, уменьшает счетчик нажатий.
2. Добавьте кнопку, сбрасывающую счетчик.
3. Модифицируйте надпись так, чтобы она гласила “Кнопка нажата N раз”. Реализуйте правильное склонение слова “раз”.

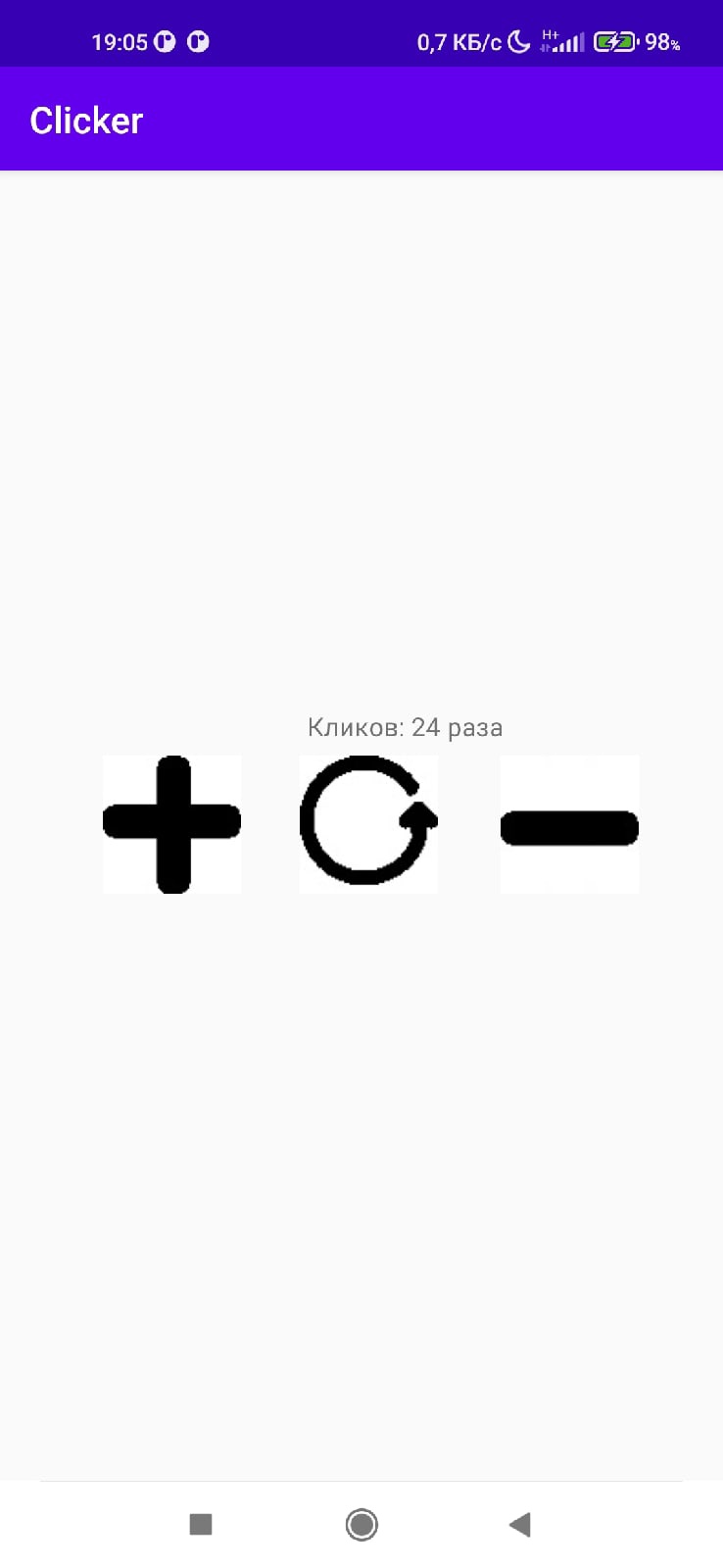
Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

1. Используйте вместо кнопки произвольную картинку.



1. (\*) Создайте приложение-калькулятор. Приложение должно предоставлять пользователю возможность вводить числа тапами по кнопкам и с клавиатуры.

Activity\_mainИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Strings

Изображение выглядит как текст

Автоматически созданное описание

MainActivity

fgИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание