**ЛАБОРАТОРНАЯ РАБОТА № 2**

**ПОЛЯ ВВОДА**

*Создание приложения с использованием элементов ImageButton, TextView, EditText и обработчика нажатия на кнопку.*

**Цель работы:** Знакомство с основными элементами в редакторе XML ANDROID STUDIO.

В данной лабораторной работе нам пригодится файл activity\_main.xml, в котором мы создадим поля для взаимодействия с программой. Файл имеет расширение XML, eXtensible Markup Language — «расширяемый язык разметки» и представляет собой текстовый файл, в котором хранится представление основного (main) окна проекта.

XML поддерживает обмен информацией между компьютерными системами, такими как веб-сайты, базы данных и сторонние приложения. Предопределенные правила упрощают передачу данных в виде XML-файлов по любой сети, поскольку получатель может использовать эти правила для точного и эффективного чтения данных.

Если проводить аналогию с Windows, то ANDROID приложение состоит из окон, называемых Activity. В конкретный момент времени обычно отображается одно Activity и занимает весь экран, а приложение переключается между ними. В качестве примера можно рассмотреть почтовое приложение. В нем одно Activity – список писем, другое – просмотр письма, третье – настройки ящика. При работе вы перемещаетесь по ним.

Содержимое Activity формируется из различных компонентов, называемых View. Самые распространенные View — это кнопка, поле ввода, чекбокс и т.д.

Вложенность компонентов view может быть сколь угодно глубокой. Обычно View размещают в контейнер ViewGroup. Самый распространенный пример ViewGroup – это Layout. Layout бывает различных типов и отвечает за позиционирование элементов, как будут расположены его дочерние View на экране (таблицей, строкой, столбцом и т.д.).

Итак, создадим приложение из одного окна (activity) в котором будут элементы ImageButton, TextView, EditText, кнопка и обработчик нажатий на кнопку. Задача состоит в том, чтобы по нажатию на кнопку на экране появлялся текст, введенный в поле ввода, и картинка.

Создадим новый проект на java, выбрав Empty view activity:

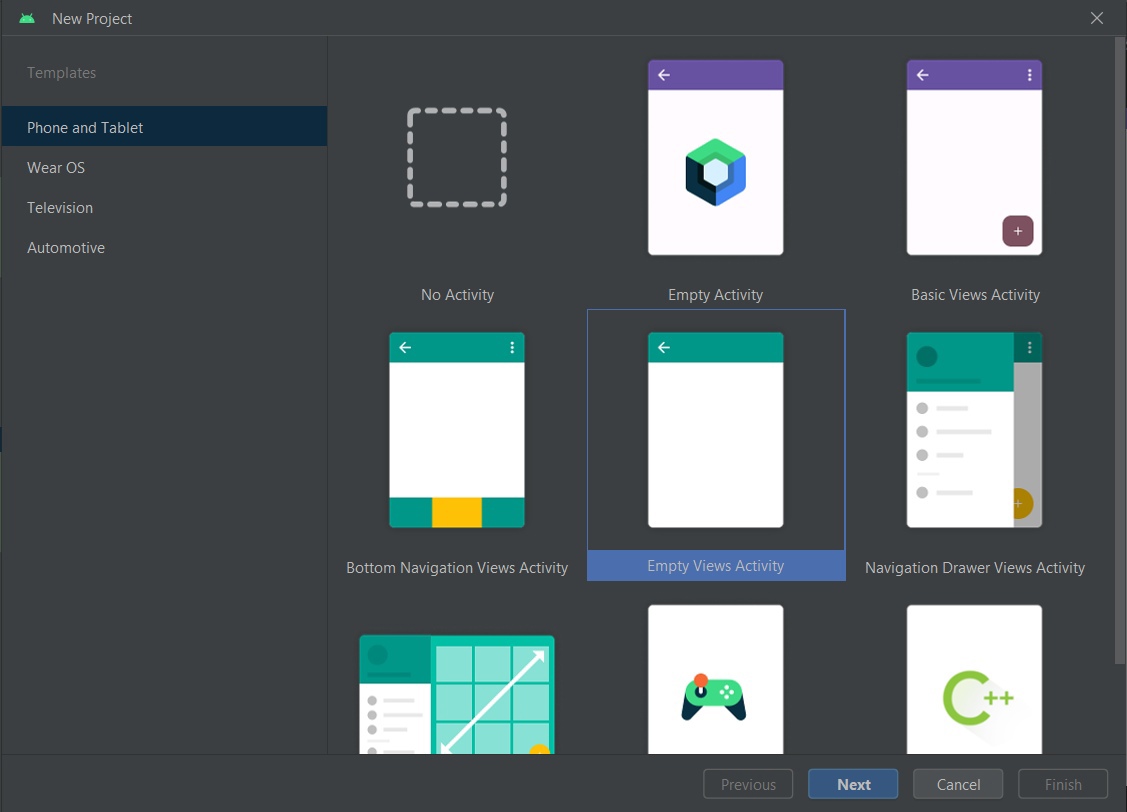


Рис. 2.1 - Empty view activity

Выберем файл activity\_main.xml. Его можно редактировать, как в редакторе кода, во вкладке code, так и графически, во вкладке design.

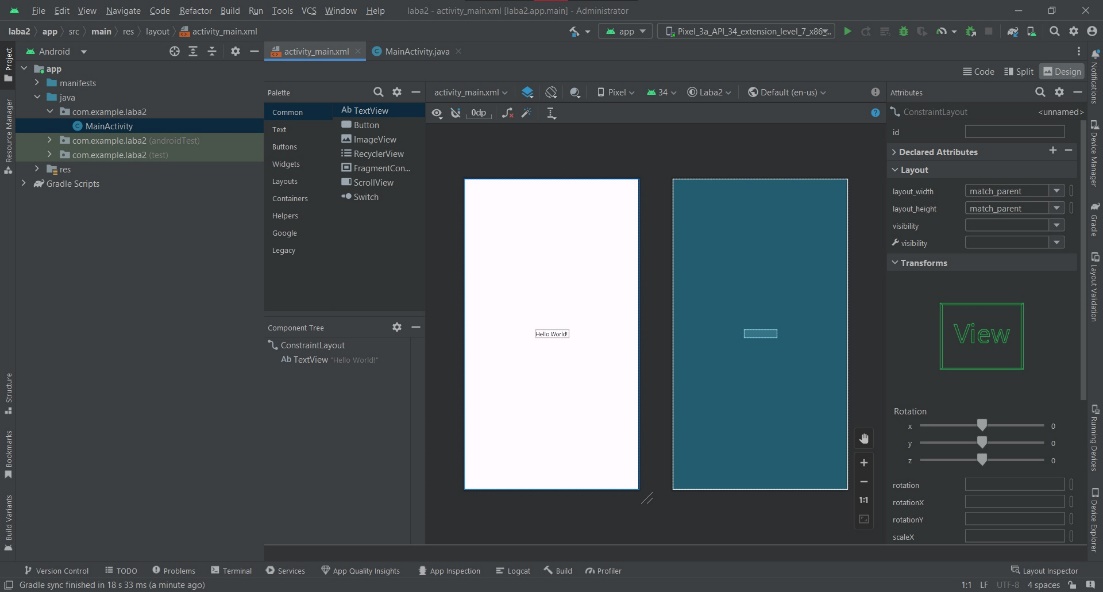


Рис 2.2 – пустой activity\_main.xml

Вытащим элементы ImageButton, TextView и Button в окно редактора, как показано на рисунке 2.3

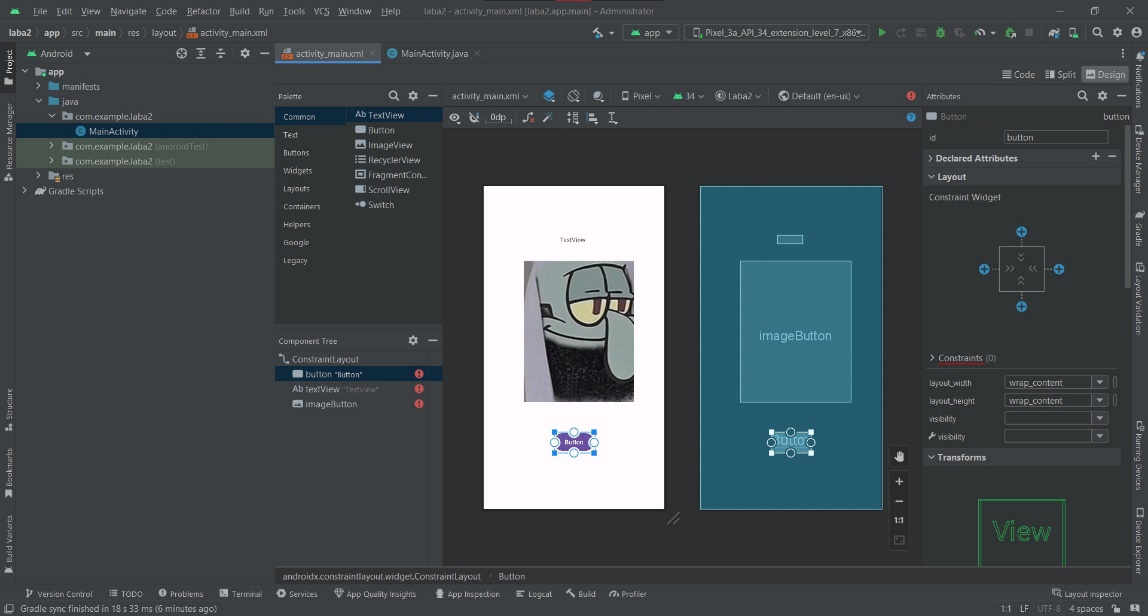


Рис. 2.3 - activity\_main.xml с тремя view компонентами

Запустим проект и увидим, что все компоненты находятся вовсе не там, где мы их оставили при редактировании.

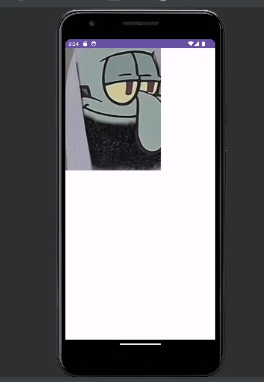


Рис 2.4 - Окно эмулятора с тремя непривязанными view компонентами

Для того, чтобы компоненты оставались в той позиции, в которой мы их оставили при редактировании, необходимо в дизайнере сцены выделить все элементы и нажать кнопку ―infer constraints‖. Она ―привяжет‖ все элементы друг к другу, а какие-то из элементов сделает ―якорями‖ – за них будут цепляться все остальные, а сам якорь, в свою очередь, за сцену(окно) activity или за контейнер, если он есть. В нашем случае, элементы не вложены в контейнер.

В данном случае, Button связана с ImageButton и с окном, а TextView связан с окном отдельно.

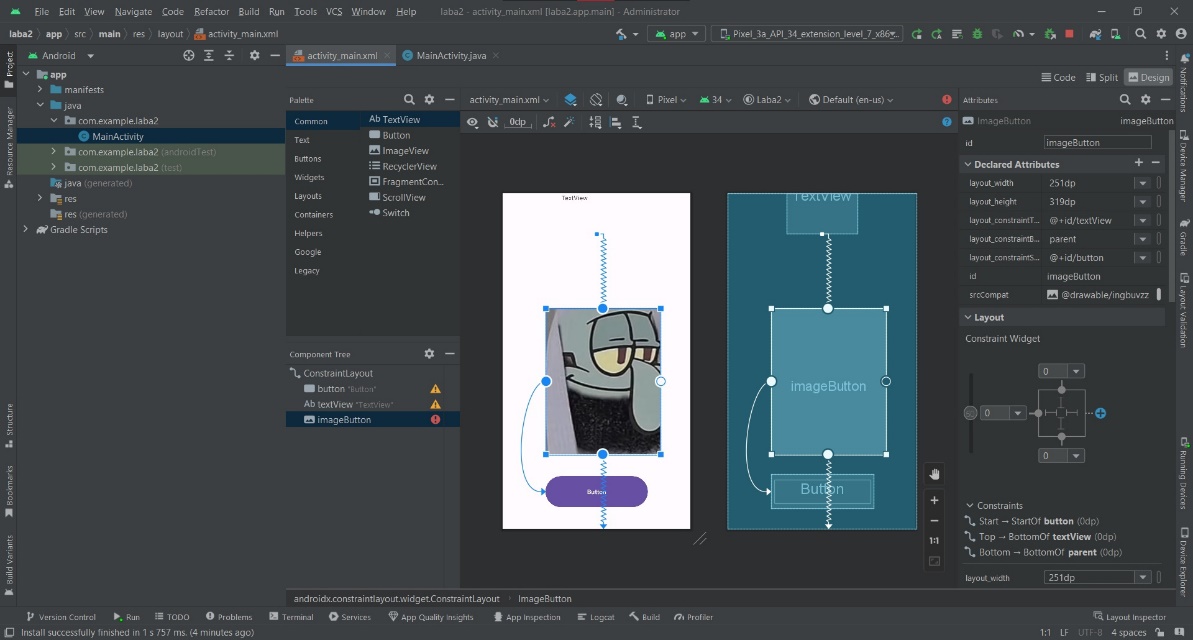


Рис 2.5 - Окно дизайнера с привязанными компонентами

Обратите внимание, что привязанные друг к другу компоненты двигаются нераздельно. Привязку можно удалить, как отдельно для каждого компонента, так и удалить сразу все привязки.

Теперь, добавим в скрипт MainActivity.java следующий код:

Для начала, создадим три переменные ссылочного типа внутри класса MainActivity:

TextView myTextView;

Button myButton;

ImageButton imageButton;

Переменные ссылочного типа хранят ссылку на объекты, так как сами объекты могут быть очень больших размеров и их размер может меняться во время выполнения программы. Объект – сложная переменная, состоящая из более простых типов данных и созданная по шаблону, называемому классом.

В эти переменные мы запишем ссылки на объекты из файла activity\_main.xml. Ссылку на объект можно получить через функцию findViewById().

По умолчанию в скрипте есть функция(метод) onCreate – она выполняется один раз во время старта программы. В ней присутствует следующий код:

super.onCreate(savedInstanceState); // обращается к методу onCreate родительского класса

AppCompatActivity setContentView(R.layout.activity\_main); // функция setContentView связывается с окном(активностью) activity\_main.xml

Дополним метод следующим кодом:

myTextView = findViewById(R.id.textView2);

myButton = findViewById(R.id.button3);

imageButton = findViewById(R.id.imageButton2);

Теперь запишем в файл activity\_main.xml в ImageButton следующую строку android:visibility="invisible". Это сделает данное поле невидимым при старте программы. Запустим программу, чтобы убедиться в этом.

Наконец, допишем скрипт MainActivity.java так, чтобы по нажатию кнопки в текстовом поле изменялся текст, а поле imageButton становилось видимым.

Для этого создадим функцию onClick() внутри следующей конструкции:

View.OnClickListener oMyButton = new View.OnClickListener() { @Override // ключевое слово для переопределения метода

public void onClick(View v) {

}

};

Метод OnClickListener класса View запустится и будет работать все время, пока запущена программа.

Теперь, внутри функции onCreate допишем следующую строку:

myButton.setOnClickListener(oMyButton); // свяжет кнопку myButton c объектом oMyButton класса OnClickListener

Допишем внутри метода onClick() следующие строки кода:

myTextView.setText("Не многие вспомнят");

imageButton.setVisibility(View.VISIBLE);

В итоге, должно получится следующее:

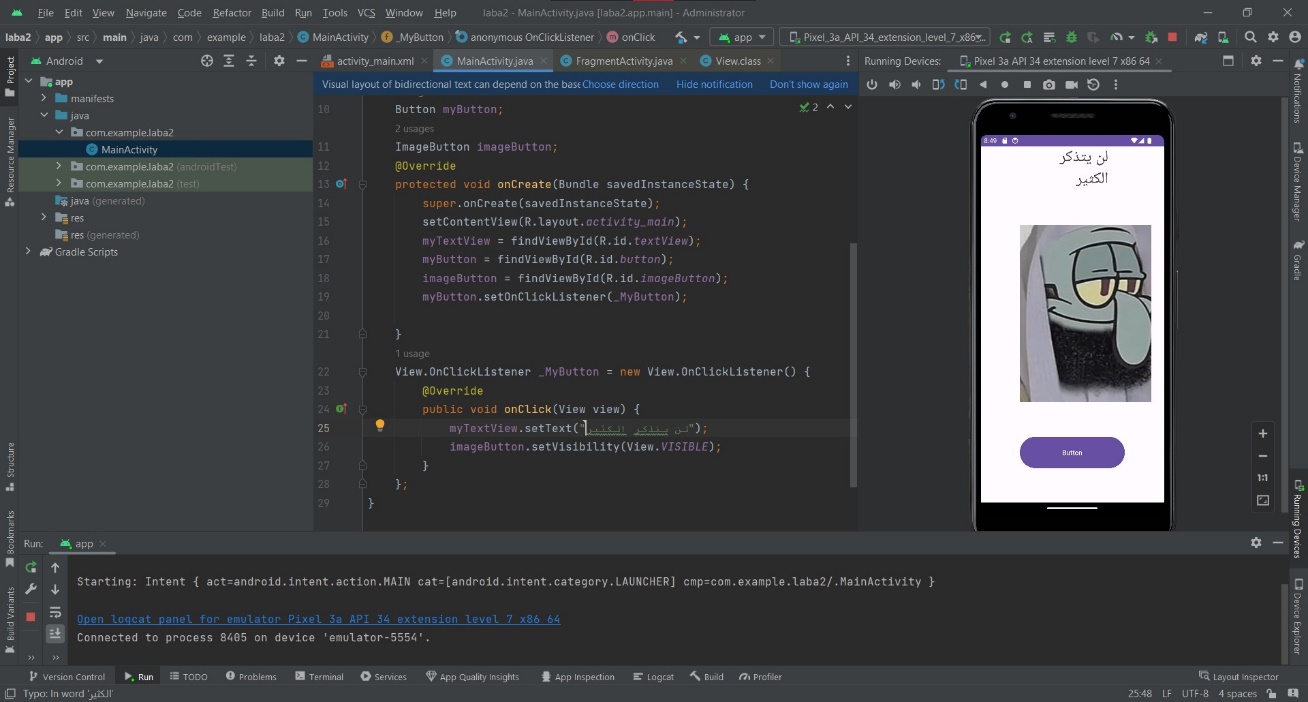


Рис. 2.6 - Итоговый скрипт лабораторной №2

**Вывод**: в результате выполнения лабораторной работы познакомились с основными элементами в редакторе XML ANDROID STUDIO.