****

**Цель работы:** научиться создавать приложения с графическим интерфейсом в среде разработки Android Studio: создавать цветовые и графические ресурсы, менять фон у приложения, обрабатывать нажатия кнопки и выводить текстовые сообщения.

Задание на лабораторную работу

1. Запустите Android Studio.

2. Создайте проекты Hello Friend с использованием языков Java и Kotlin. Используйте разные цвета фона экрана и разные картинки.

При запуске приложения проверьте оба случая – когда имя введено и когда нет. Коды и скриншоты с результатами занесите в отчет.

3. Создайте проекты Counter с использованием языков Java и Kotlin. При запуске приложения проверьте, что произойдет при нажатии на кнопку «Поздороваться» и при нажатии на кнопку «Считаем студентов». Коды и скриншоты с результатами занесите в отчет.

4. Сделайте одно из приложений Counter ещё более полезным – оно будет позволять подсчитывать не только студентов, но и преподавателей. Для этого добавьте в проект ещё одну кнопку, чтобы считать студентов и преподавателей по отдельности. При запуске приложения проверьте, что произойдет при нажатии на каждую из кнопок. Код и скриншоты с результатами занесите в отчет.

Ход работы:

1. Hello Friend
   1. Kotlin

В android studio создаем новый проект. На вкладке Phone and Tables выберем шаблон Empty Activity (рисунок 1).

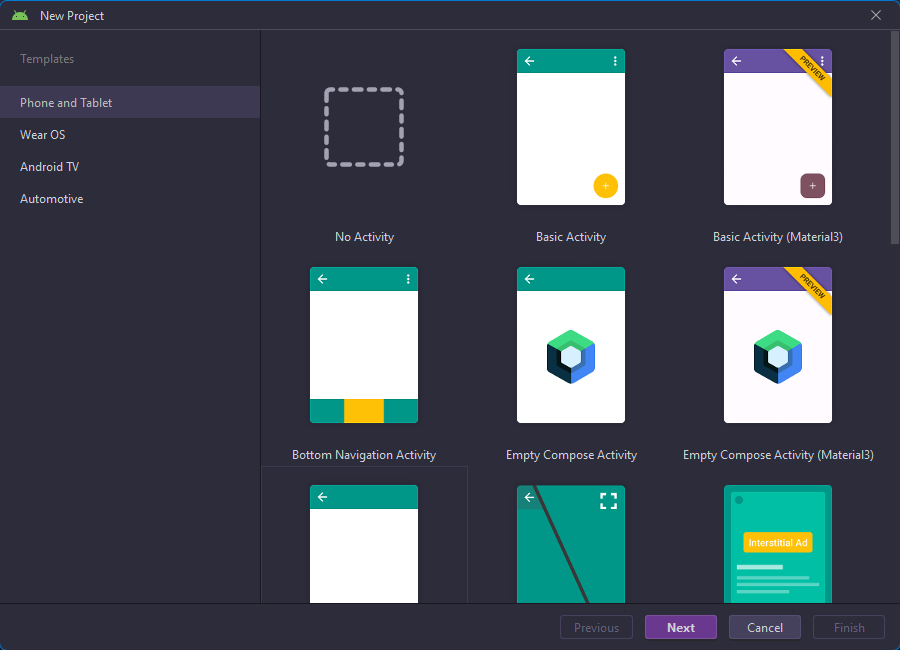


Рисунок 1 – выбор шаблона

После чего напишем название будущего проекта, в нашем случае “Hello Friends Ivan K” и выберем язык программирования kotlin (рисунок 2).

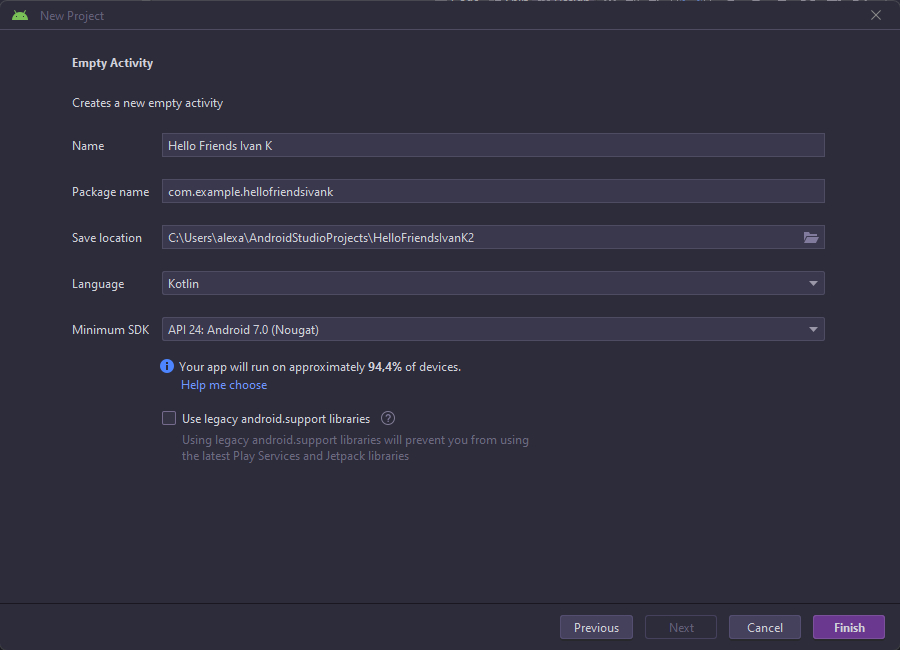


Рисунок 2 – новый проект в Android studio

После открытия проекта просмотрим файл activity\_main.xml в режиме дизайна (рисунок 3) и в режиме текста (рисунок 4).

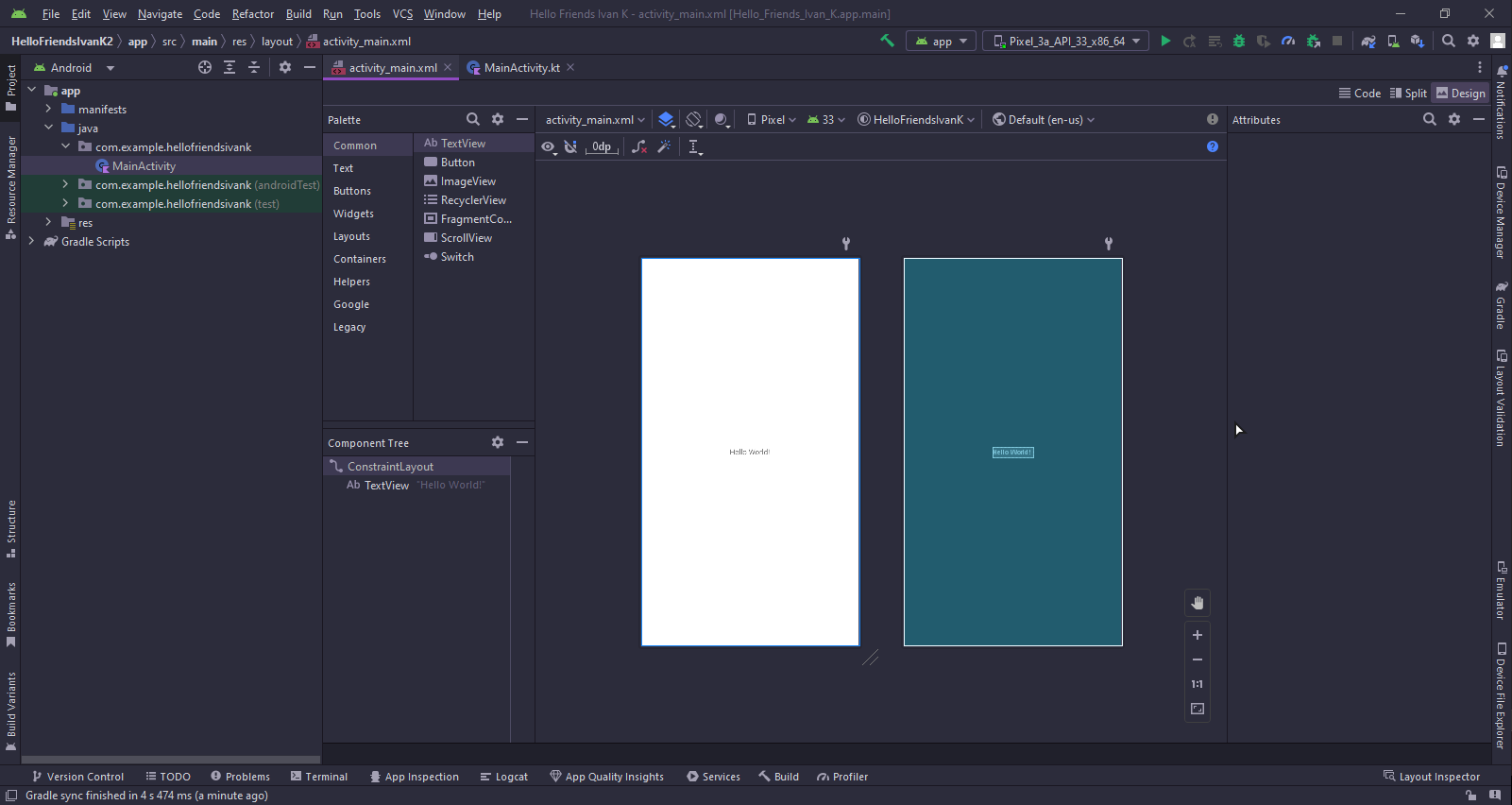


Рисунок 3 – XML-файл в режиме дизайна

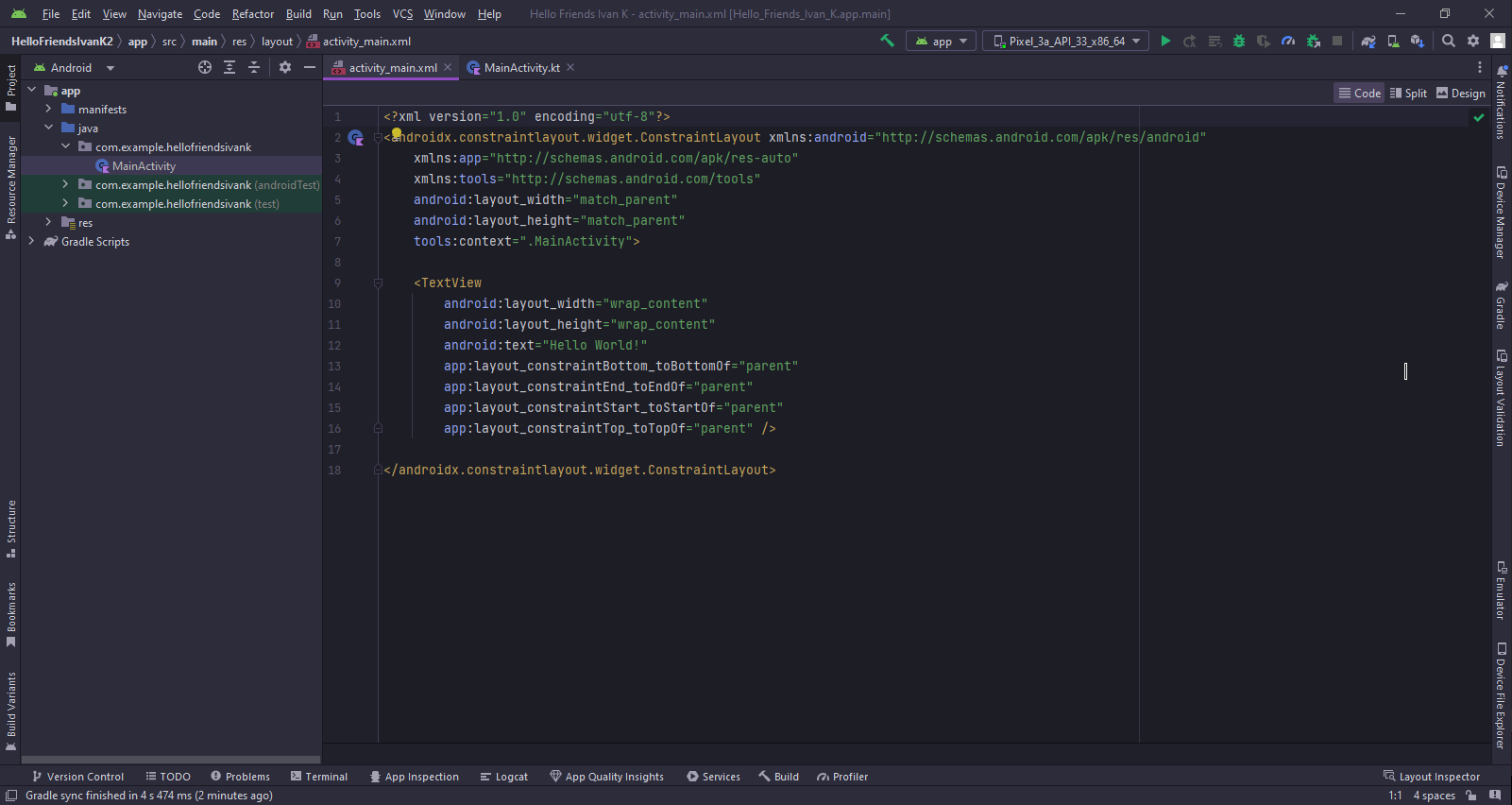


Рисунок 4 – XML-файл в режиме текста

В режиме дизайна добавим на форму элемент ImageButton из панели инструментов (рисунок 5).

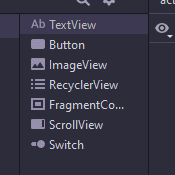


Рисунок 5 – элемент ImageButton на панели инструментов

При добавлении элемента у нас открывается диалоговое окно, на котором мы можем выбрать изображение для кнопки (рисунок 6).

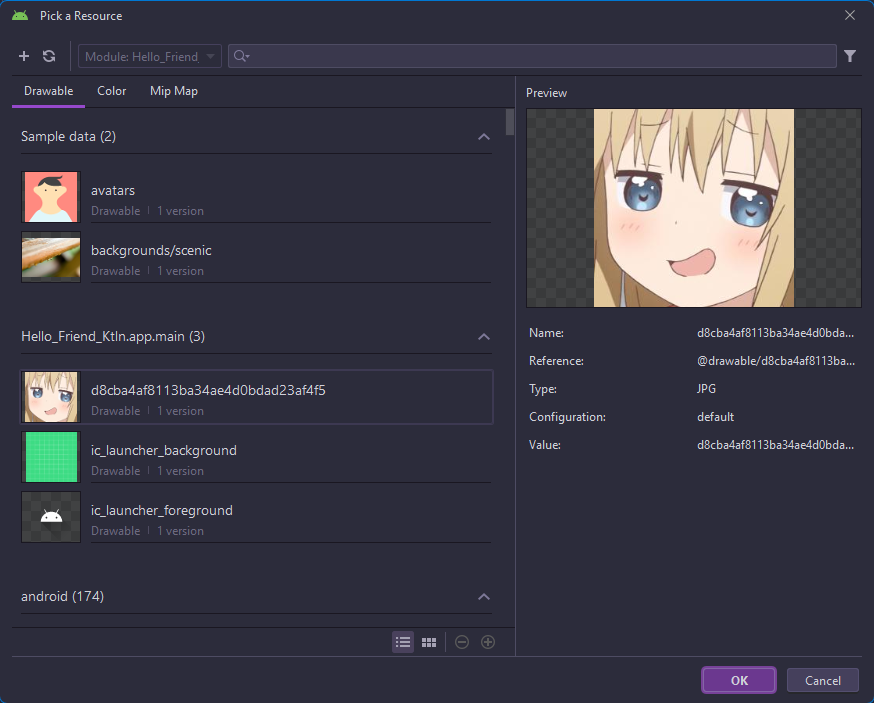


Рисунок 6 – выбор изображения для кнопки

После выбора изображение изменим задний фон проекта, для этого на панели Component Tree выберем элемент ConstraintLayout, затем увидим окно атрибутов, в котором можно изменять различные свойства объектов (рисунок 7), там же выберем свойство background и выберем цвет заднего фона (рисунок 8).

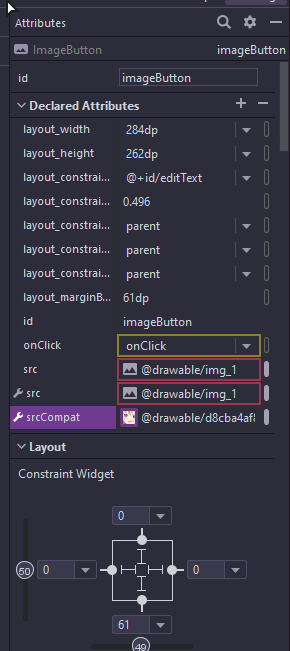


Рисунок 7 – окно атрибутов элементов

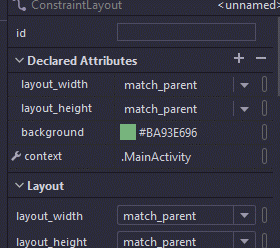


Рисунок 8 – выбор цвета заднего фона приложения

Затем мы выбрали изображение и загрузили его в Android Studio, после чего установим его в качестве изображения для созданной ранее кнопки. В свойствах кнопки найдем поле src и выберем добавленное нами изображение (рисунок 9).

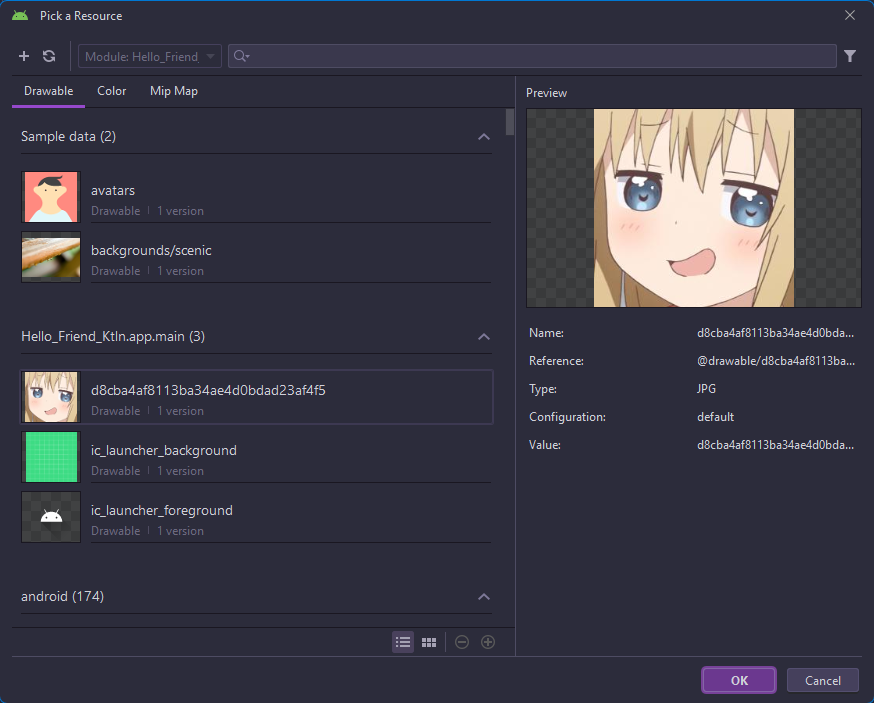


Рисунок 9 – выбор нового изображения для кнопки

Применим некоторые изменения в коде приложения, после чего запустим проект. В открывшемся окне эмулятора мы видим кнопку и надпись, после нажатия данной кнопки мы получаем новую надпись (рисунок 10).

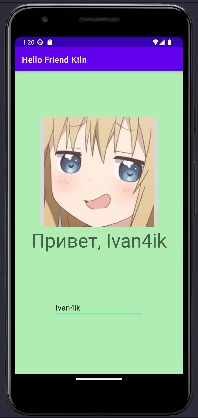
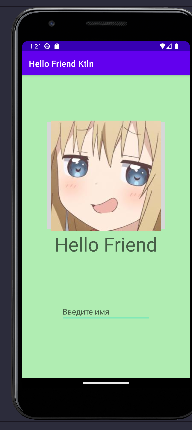


Рисунок 10 – окно эмулятора до и после нажатия кнопки

Добавим в программу дополнительный функционал, сделаем поле для ввода имени, чтобы при нажатии кнопки программа обращалась к этому имени. Для этого добавим на экран элемент Plain Text, в свойствах которого удалим текст и добавим подсказку «Введите имя», которая будет исчезать при вводе текста (рисунок 11).

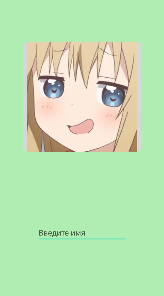


Рисунок 11 – элемент Plain Text на экране

После этого в коде программы добавим проверку по щелчку кнопки, если в добавленном нами поле пустой текст, то на экран выводится надпись «Hello Friend», при вводе имени, программа поприветствует, используя это имя. Запустим программу, при вводе имени, как и ожидалось, мы видим, что программа использует это имя в приветствии (рисунок 12). Если оставить это поле пустым, то приветствие не изменится (рисунок 13).

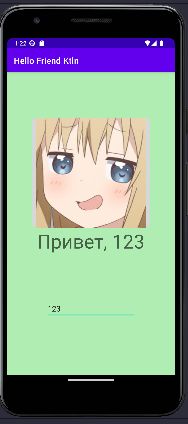
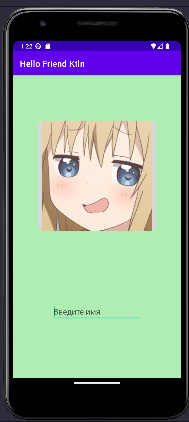


Рисунок 12 – окно эмулятора при вводе имени до и после нажатия кнопки

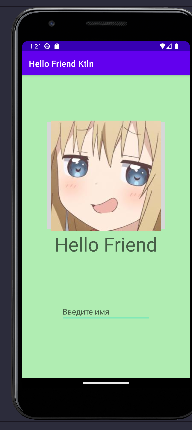


Рисунок 13 – окно эмулятора при отсутствии имени

* 1. Java

В android studio создаем новый проект. На вкладке Phone and Tables выберем шаблон Empty Activity (рисунок 14).

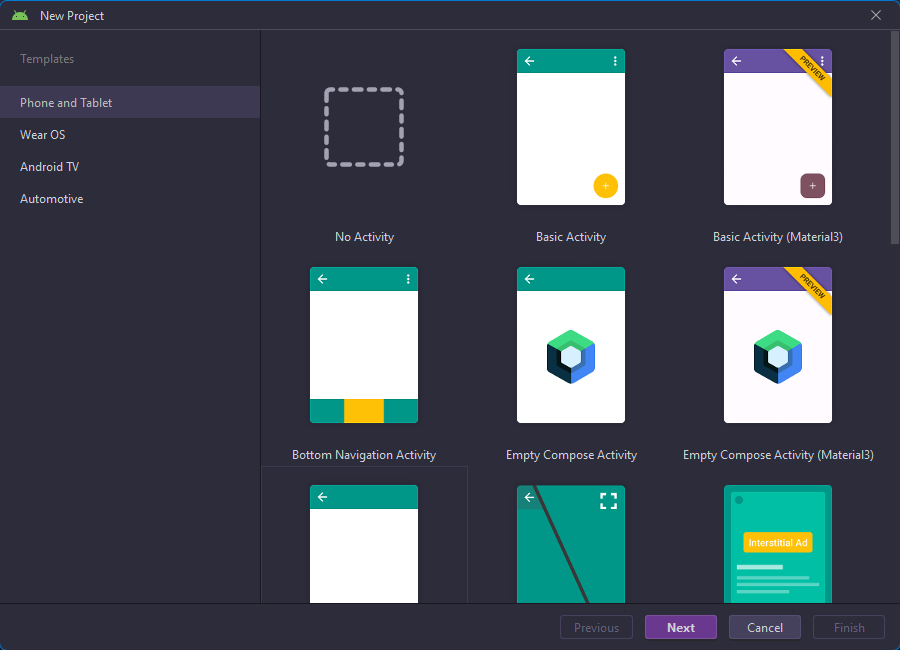


Рисунок 14 – выбор шаблона

После чего напишем название будущего проекта, в нашем случае “Hello Friend Ivan J” и выберем язык программирования java (рисунок 15).

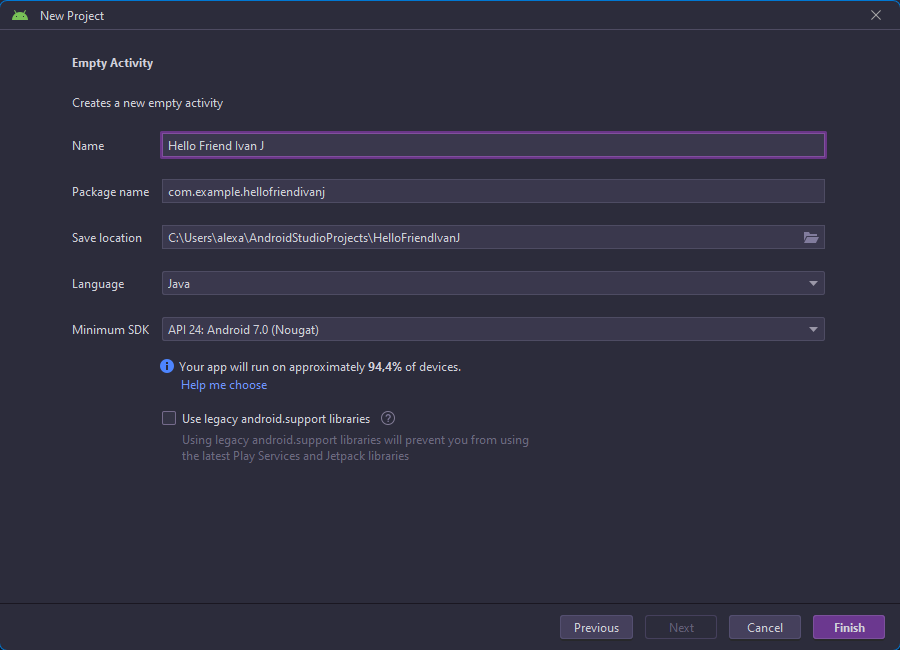


Рисунок 15 – новый проект в Android studio

После открытия проекта просмотрим файл activity\_main.xml в режиме дизайна (рисунок 16) и в режиме текста (рисунок 17).

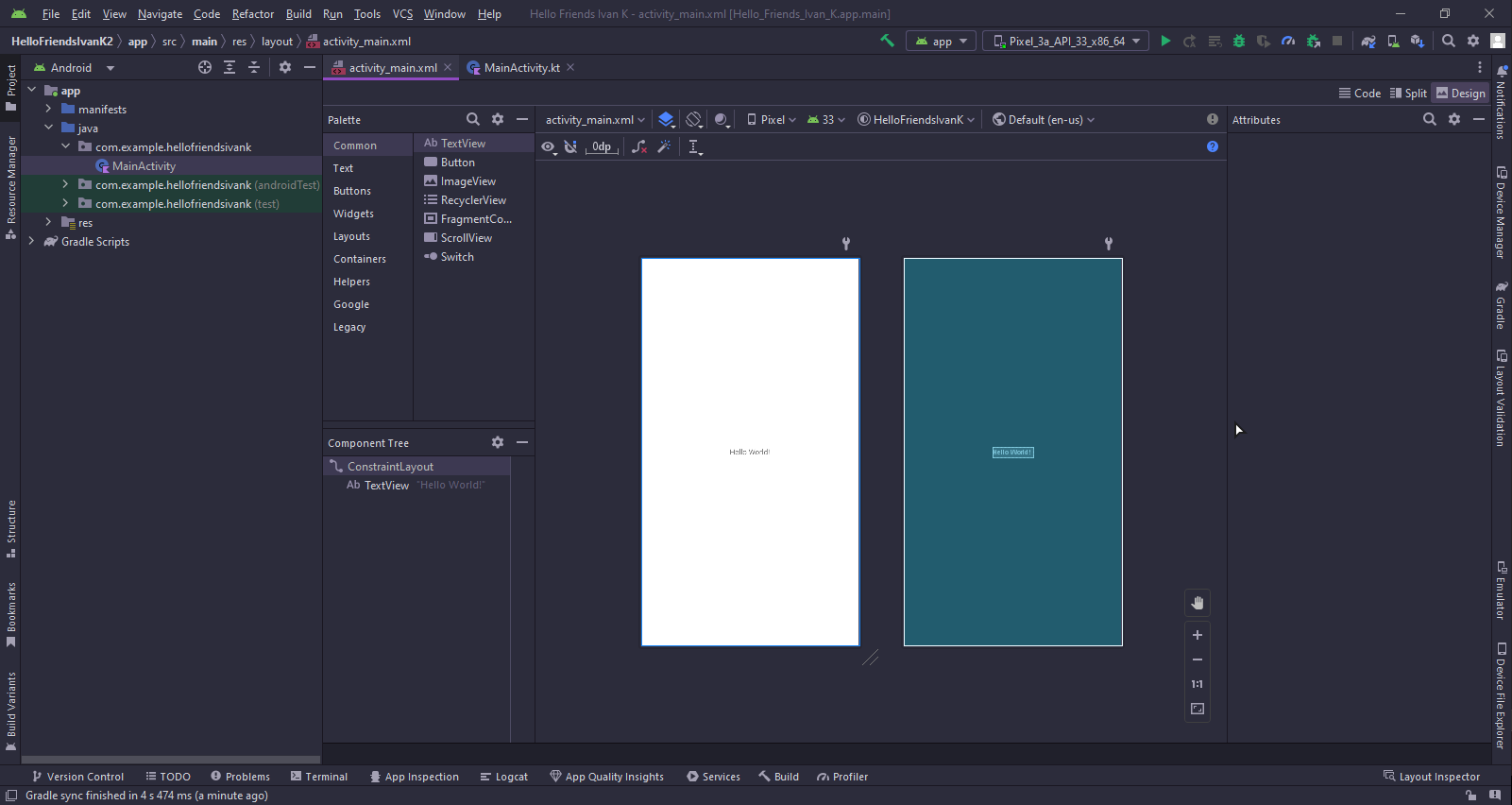


Рисунок 16 – XML-файл в режиме дизайна

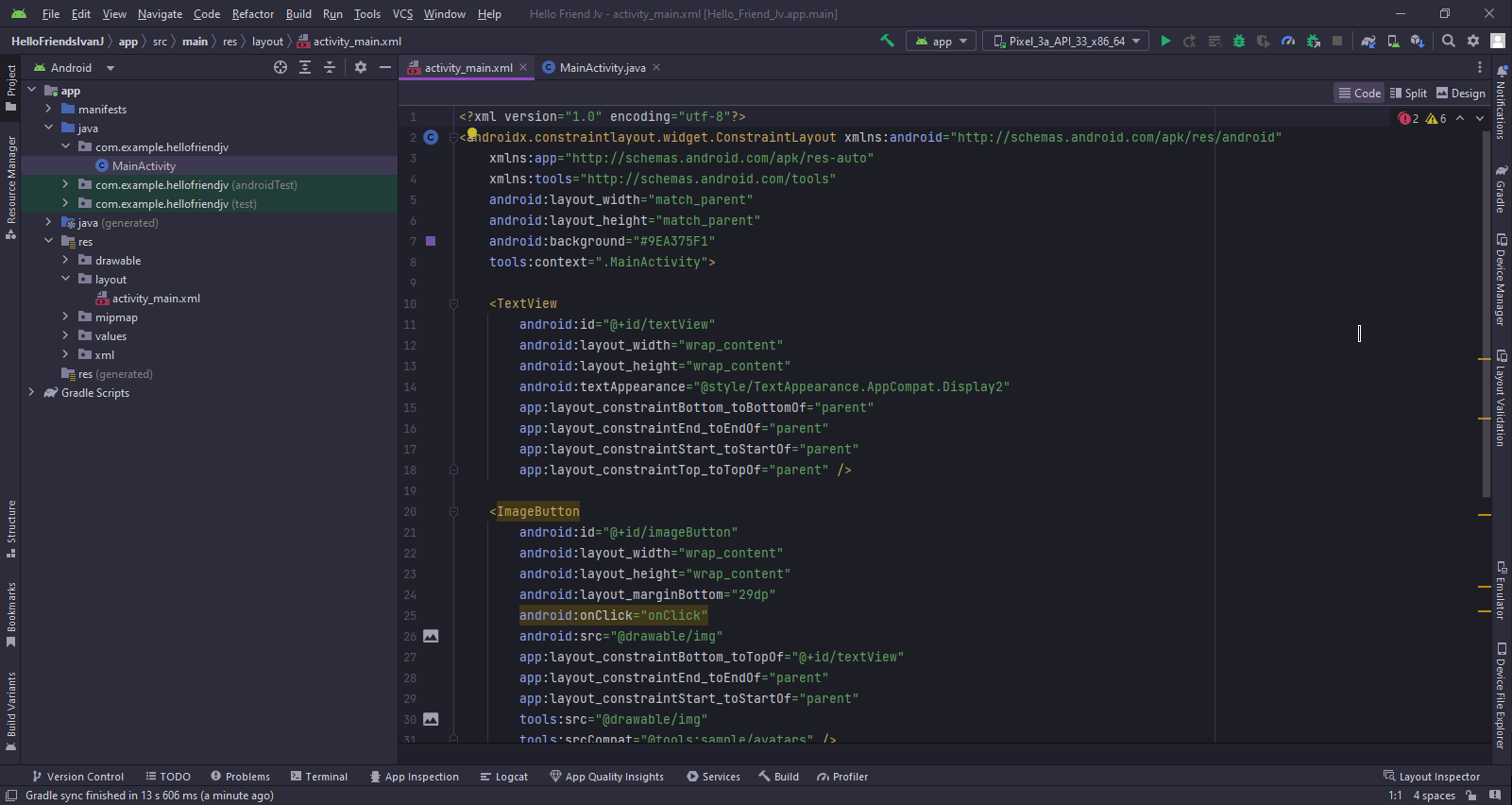


Рисунок 17 – XML-файл в режиме текста

Добавим кнопку с панели инструментов (рисунок 18). При добавлении выберем любое из представленных в списке изображений (рисунок 19).

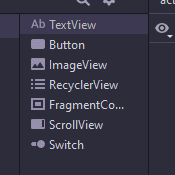


Рисунок 18 – элемент ImageButton на панели инструментов

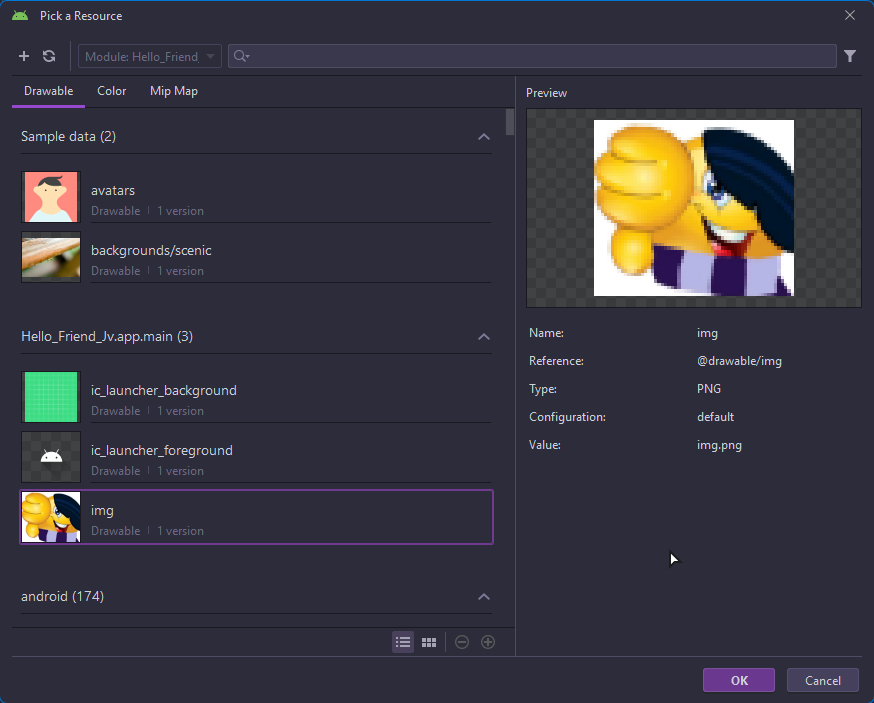


Рисунок 19 – выбор изображения для кнопки

Затем изменим цвет заднего фона приложения, зайдя в свойства объекта и изменив атрибут background (рисунок 20).

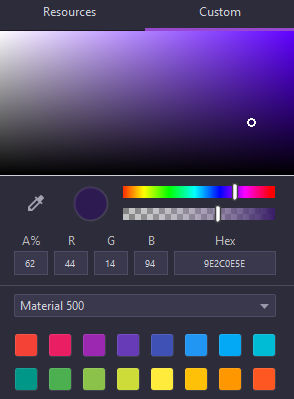


Рисунок 20 – выбор цвета фона приложения

Загрузим изображение в папку, после чего изменим рисунок кнопки в её свойствах (рисунок 21).

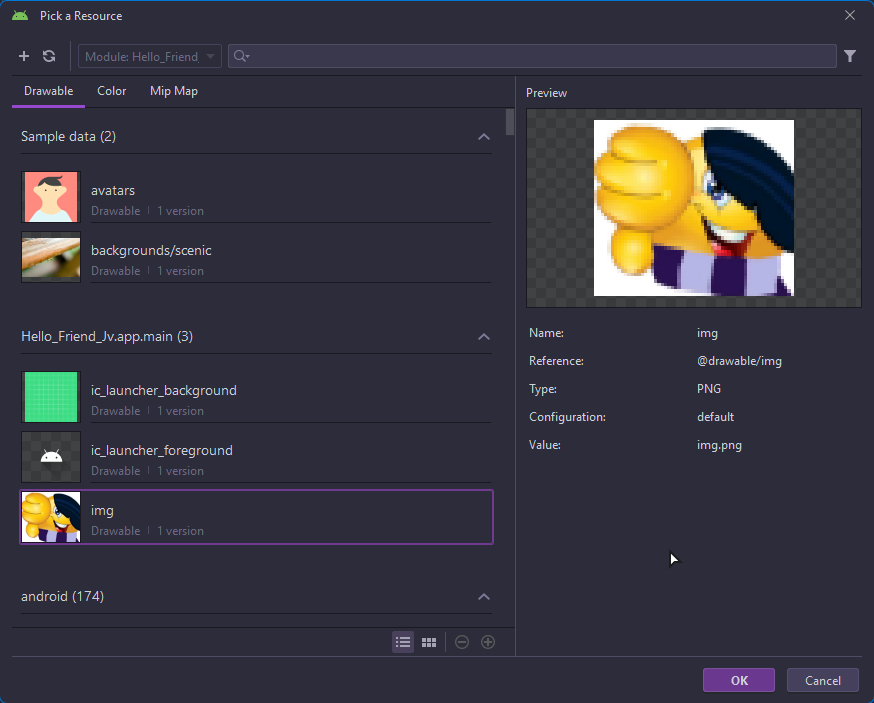


Рисунок 21 – выбор нового рисунка для кнопки

Изменим свойства у кнопки, для этого в свойстве onClick пропишем «onClick» (рисунок 22), после чего в форме кода нажмем на onClick и прожмем комбинацию Alt+Enter и выберем первый вариант (рисунок 23).

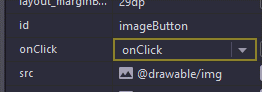


Рисунок 22 – свойство onClick кнопки

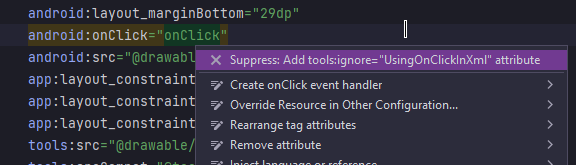


Рисунок 23 – изменение кода проекта

После этого изменим код в файле «MainActivity.java» и запустим проект. В открывшемся окне эмулятора нажмем на кнопку и получим следующий результат (рисунок 24).

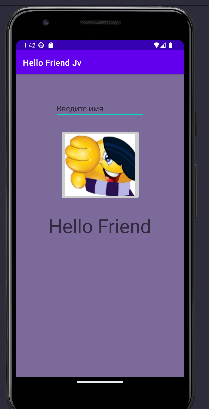
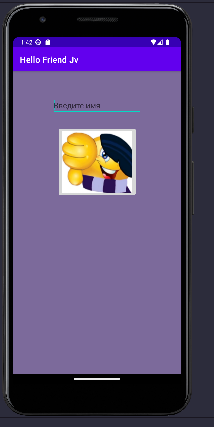


Рисунок 24 – окно эмулятора до и после нажатия кнопки

Добавим дополнительный функционал в нашу программу. Добавим поле для ввода имени, чтобы при нажатии кнопки отображалось приветствие с учетом данного имени. Изменим код, после чего запустим эмулятор. Введем имя в специальном поле и получим новое приветствие (рисунок 25), если оставить поле ввода имени пустым, то получим то же приветствие, как и до изменения программы (рисунок 26).

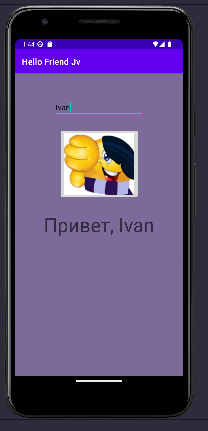
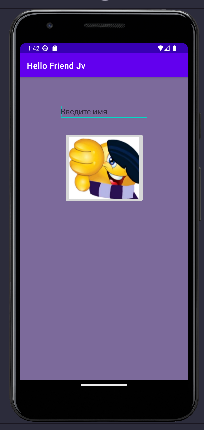


Рисунок 25 – программа после изменения при вводе имени

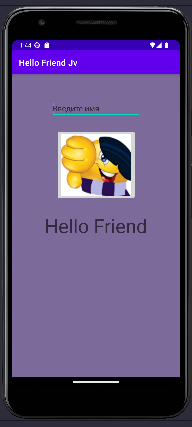


Рисунок 26 – программа после изменения при отстутствии введенного имени

1. Counter
   1. Kotlin

Создадим новый проект с шаблоном Empty Activity, выбираем язык Kotlin, назовем проект «CounterIvanK» (рисунок 27).

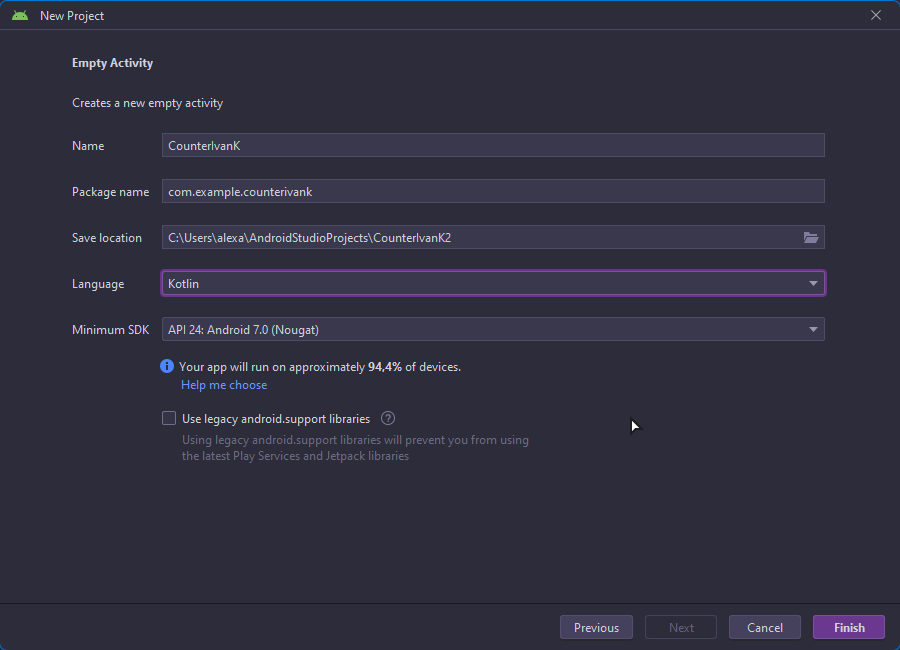


Рисунок 27 – создание нового проекта

Добавим на форму элемент Button, в свойствах которой изменим text на «Поздороваться». У компонента TextView удалим текст из поля text. В результате получим следующую форму (рисунок 28).

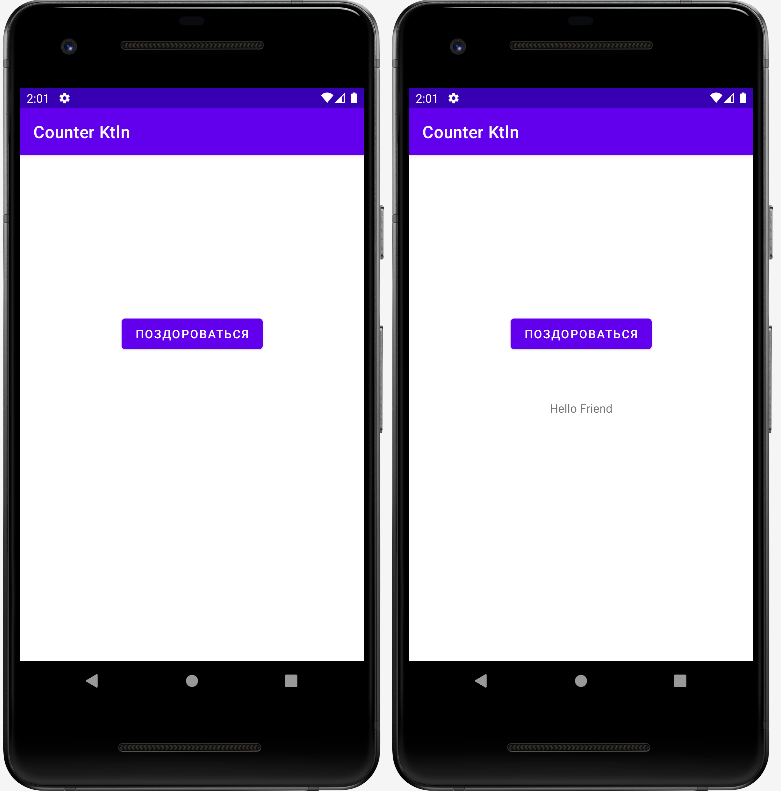


Рисунок 28 – окно эмулятора до и после нажатия кнопки

Усложним программу, добавим ещё одну кнопку на форму, разместим ее под первой кнопкой, после чего перейдем в режим чтения кода. Там изменим id этой кнопки с «id/button» на «id/button\_сounter», а также в поле text вместо «Button» напишем "Считаем студентов". Перейдем в режим дизайна и увидим изменения кнопки.

В файле MainActivity добавим функционал для новой кнопки: задаем новую переменную «counter», а также прописываем текст, чтобы при нажатии кнопки к переменной counter прибавлялось 1 с помощью операции инкремента ++ перед переменной и на экран выводилась надпись с указанием данного числа. Если поставить операцию после переменной, то при первом нажатии на кнопку сначала будет выводиться число 0, а не 1, как в первом случае. Запустим проект и получим следующий результат.

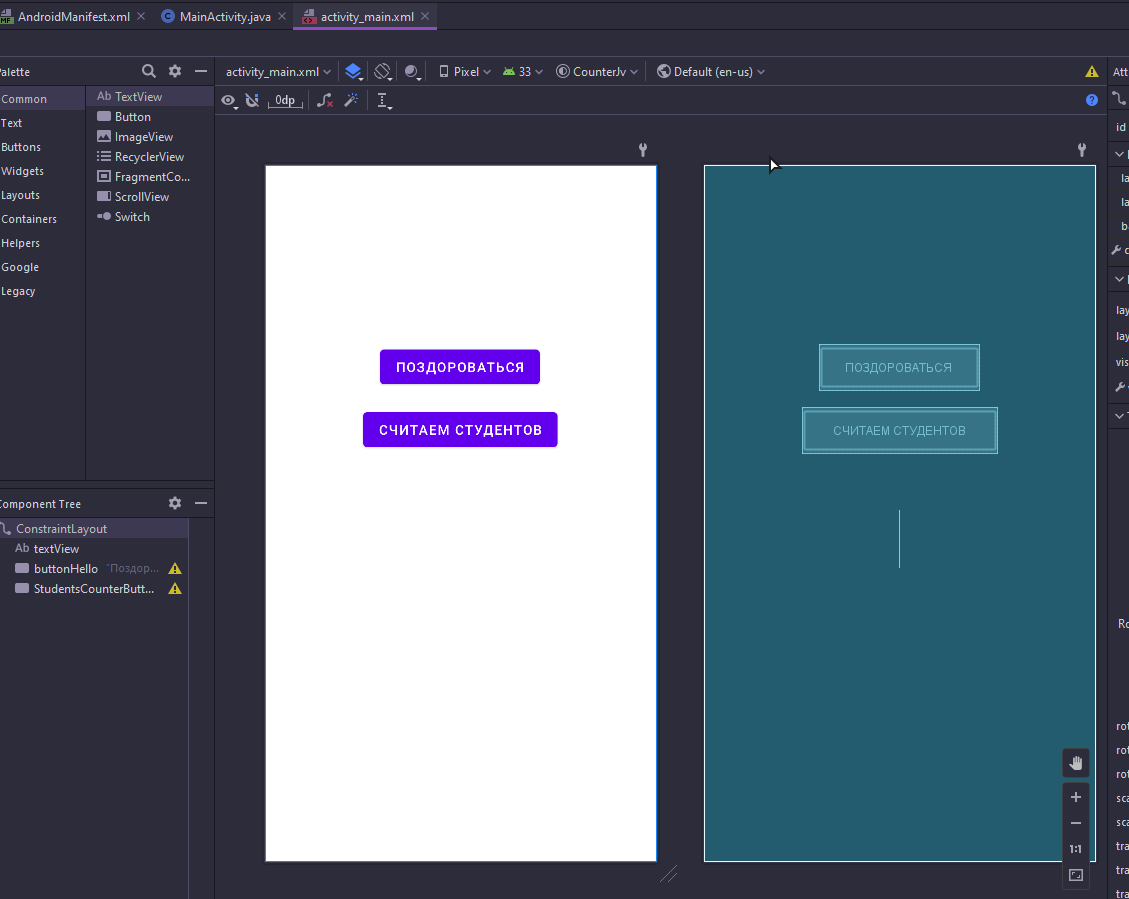
  
Рисунок 29 – окно эмулятора до и после нескольких нажатий на кнопку

* 1. Java

Создадим новый проект на языке Java, назовем его «Counter Ivan J»

Добавим на поле кнопку, после чего в ее свойствах изменим имя на «Поздороваться», а id изменим на «buttonHello». Уберем у элемента textView имя, после чего изменим цвет заднего фона всего приложения. Определим событие для нажатия кнопки и добавим обработку данного события. Изменим код, после чего запустим эмулятор и получим следующий результат (рисунок 30).

Усложним программу, добавив счетчик студентов. Добавим на форму новую кнопку и разместим под первой кнопкой и изменим свойства id на «StudentsCounterButton» и text на «Считаем студентов» (рисунок 30).

  
Рисунок 30 – измененная форма

Затем переходим в файл MainActivity, где пишем функционал для добавленной ранее кнопки. После изменения кода нажатие кнопки приводит к тому, что текстовое поле пишет «Я насчитал «число» студентов», где «число» - переменная, которая увеличивается каждый раз на 1 при нажатии на кнопку. Проверим работоспособность программы (рисунок 31).

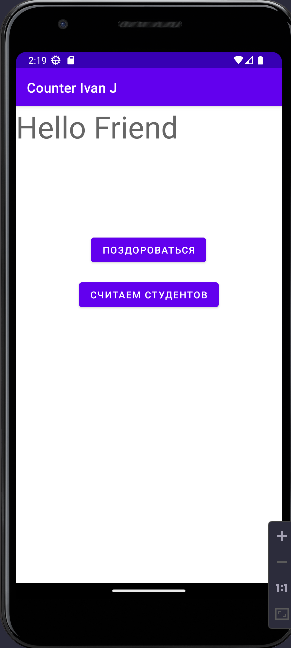
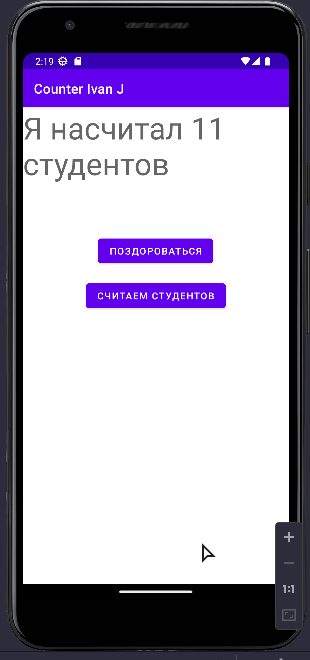
 

Рисунок 31 – окно эмулятора до и после нескольких нажатий на кнопку

1. Улучшенный counter

На языке Kotlin дополним созданный ранее counter, добавив дополнительную кнопку для подсчета преподавателей. Для этого на форму добавим дополнительные объекты button и textView. Пропишем им новые id для удобства и перейдем в файл MainActivity, где пропишем функционал для кнопки. В результате доработки кода по нажатию новой кнопки «Считаем преподавателей» в дополнительном текстовом поле срабатывает подсчет преподавателей (рисунок 32).

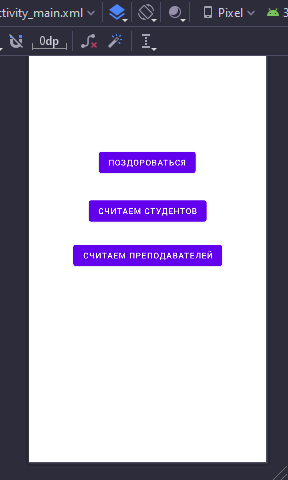
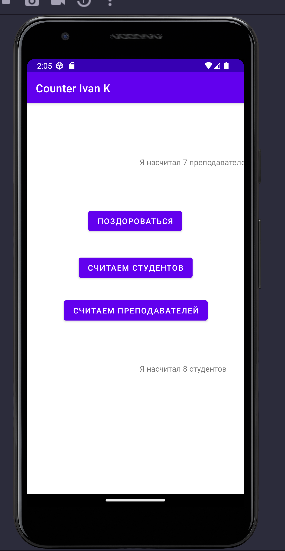
 

Рисунок 32 – окно эмулятора до и после нескольких нажатий кнопок