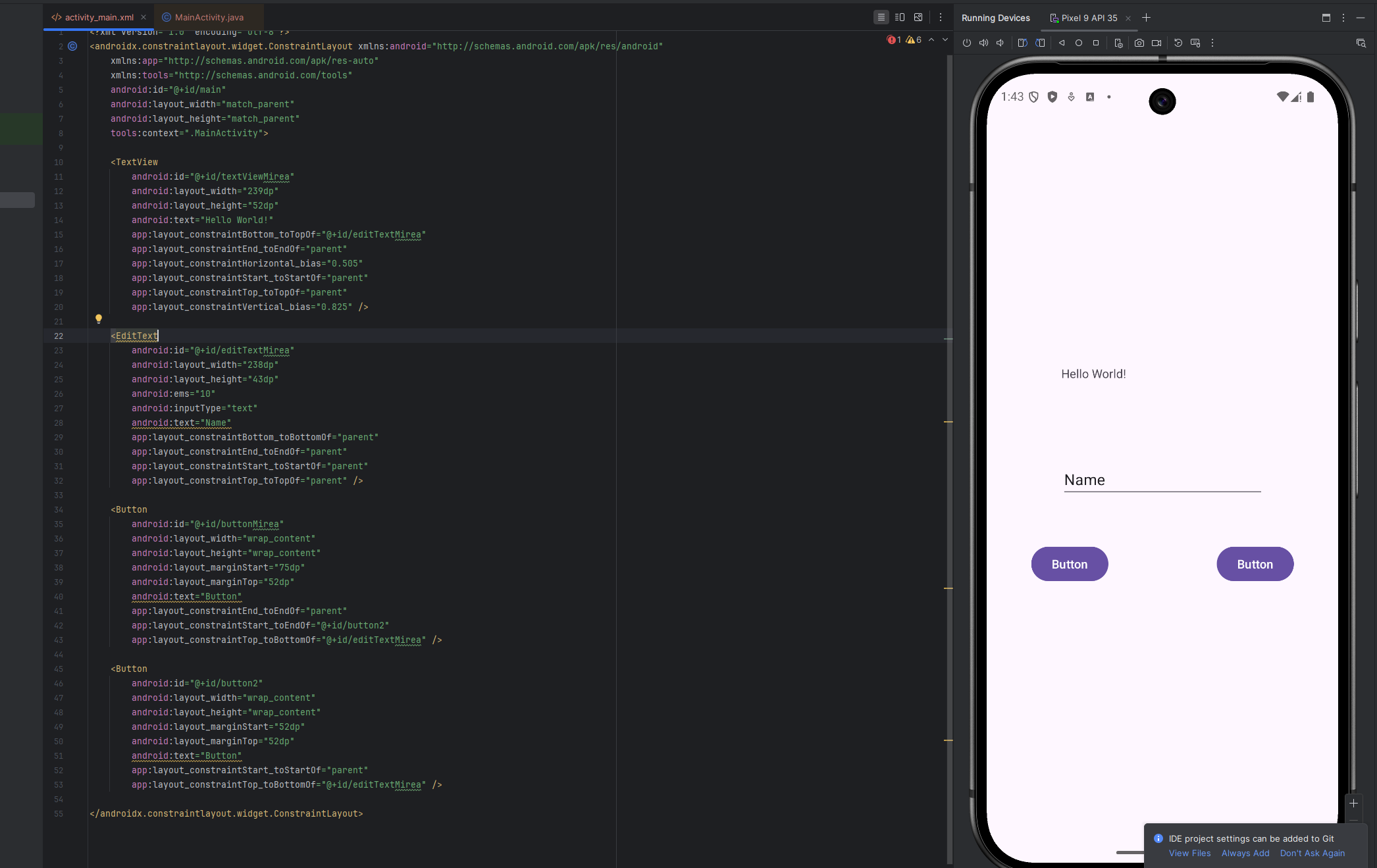
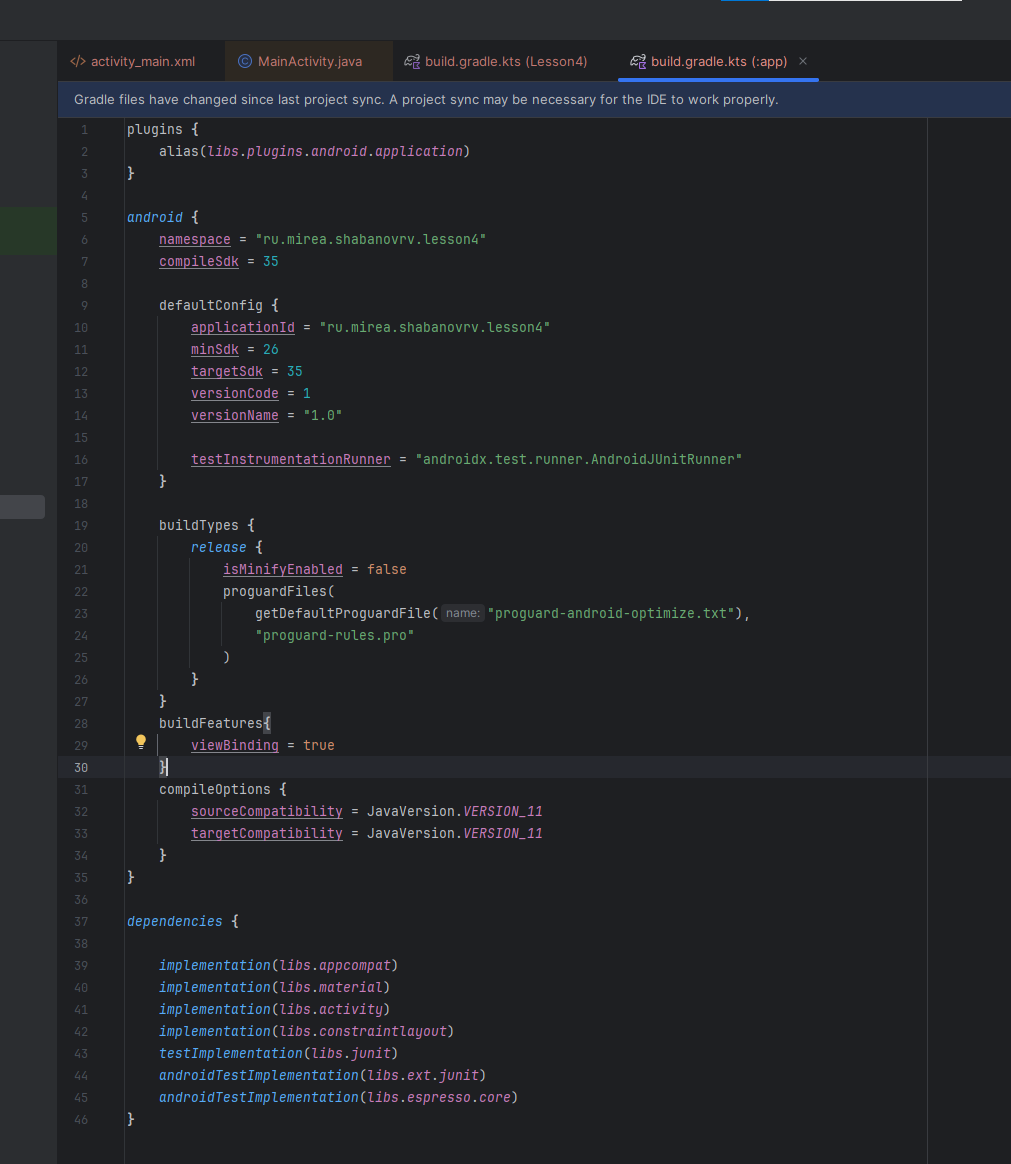
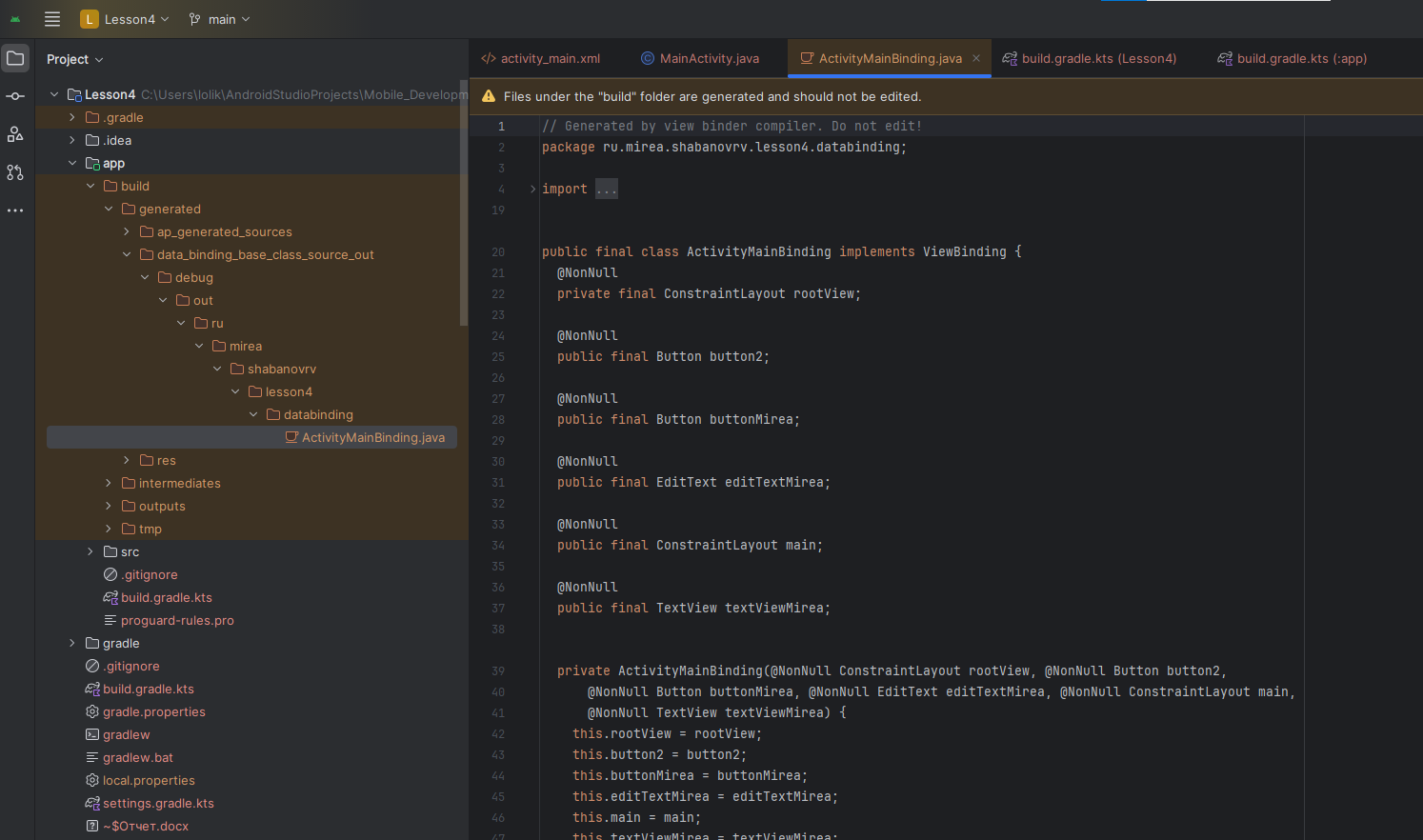
В файл разметки «activity\_main» требуется добавить несколько элементов графического интерфейса («button», «textView» и т.д.) и установить соответствующий идентификатор.



Включение viewBinding

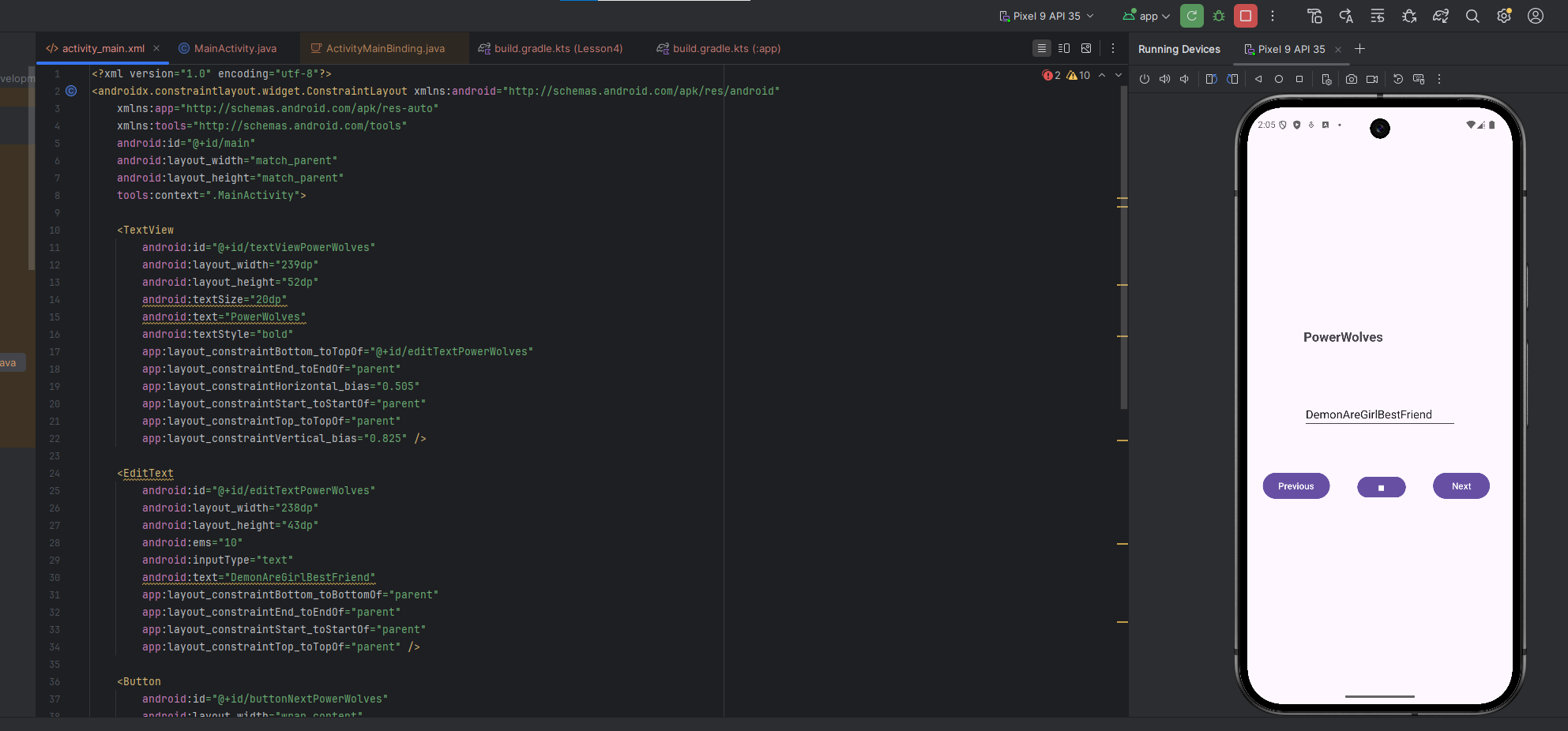


Путь до ActivityMainBulding

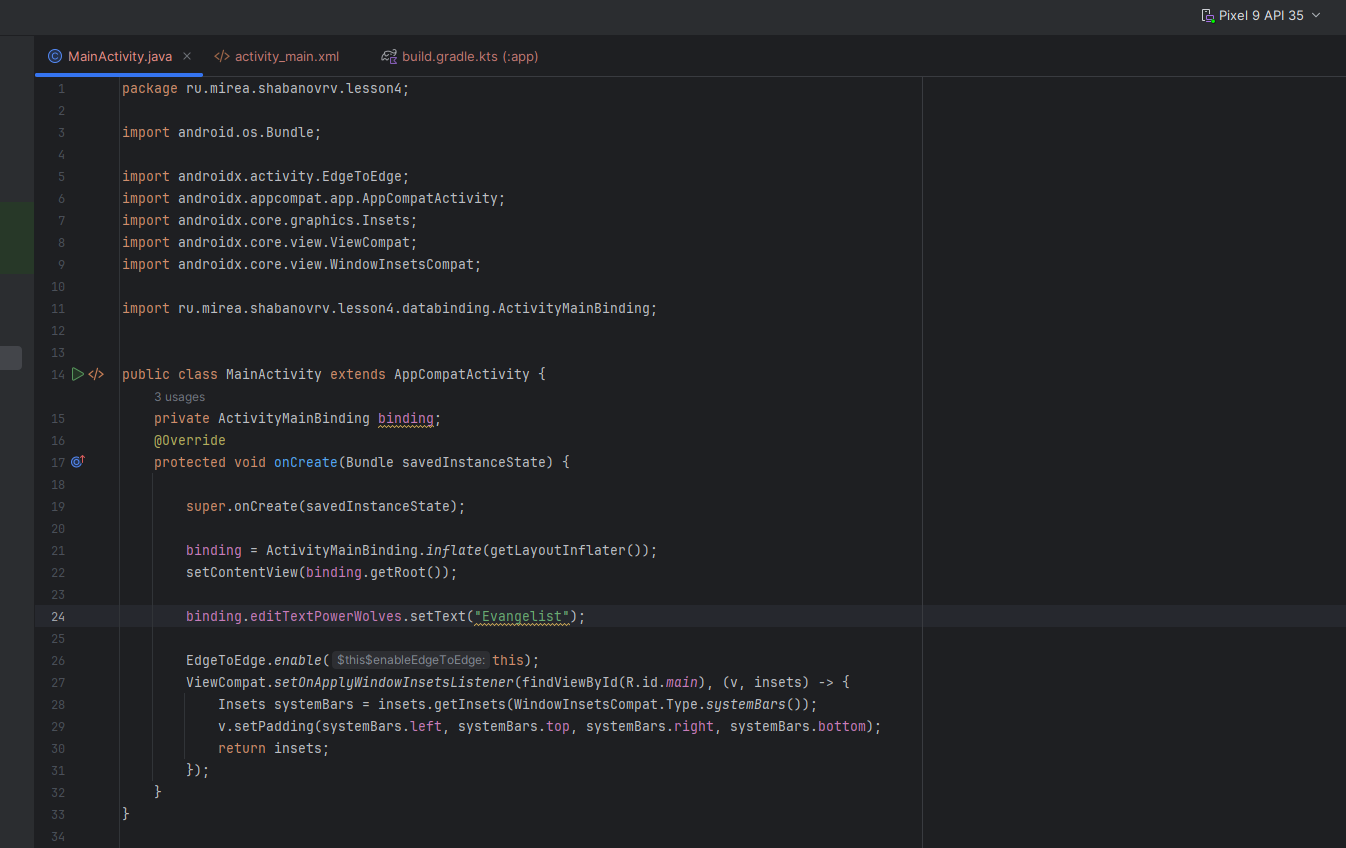


**Задание 1.**

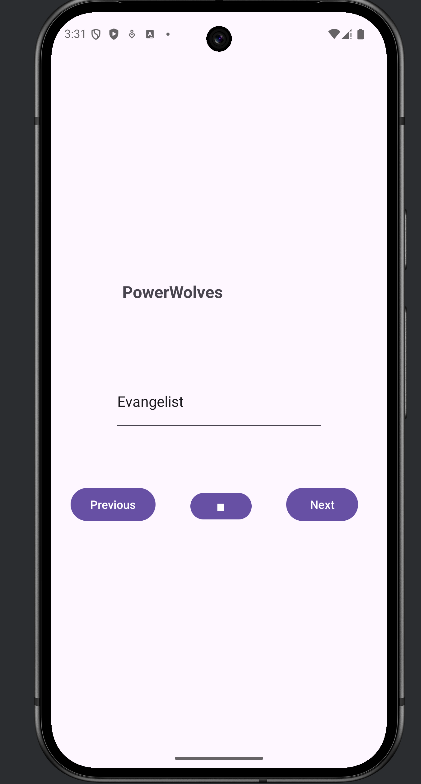
Требуется создать экран музыкального плеера с использованием «binding» для горизонтальной и портретной ориентации.

Создадим макет видео плеера 

Обновляем файл gradle, пересобираем приложение, и прописываем binding c изменением названия музыки



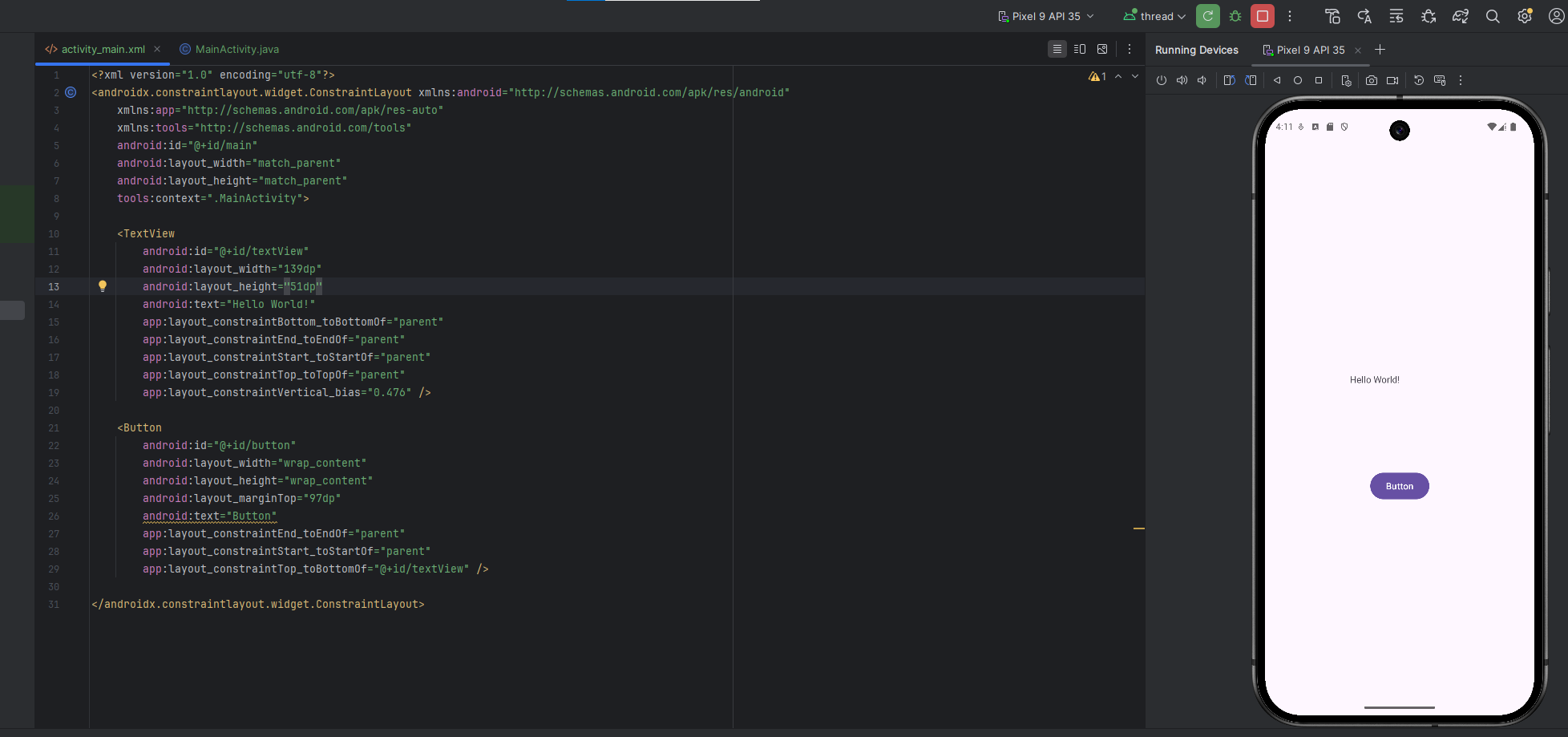
Получаем



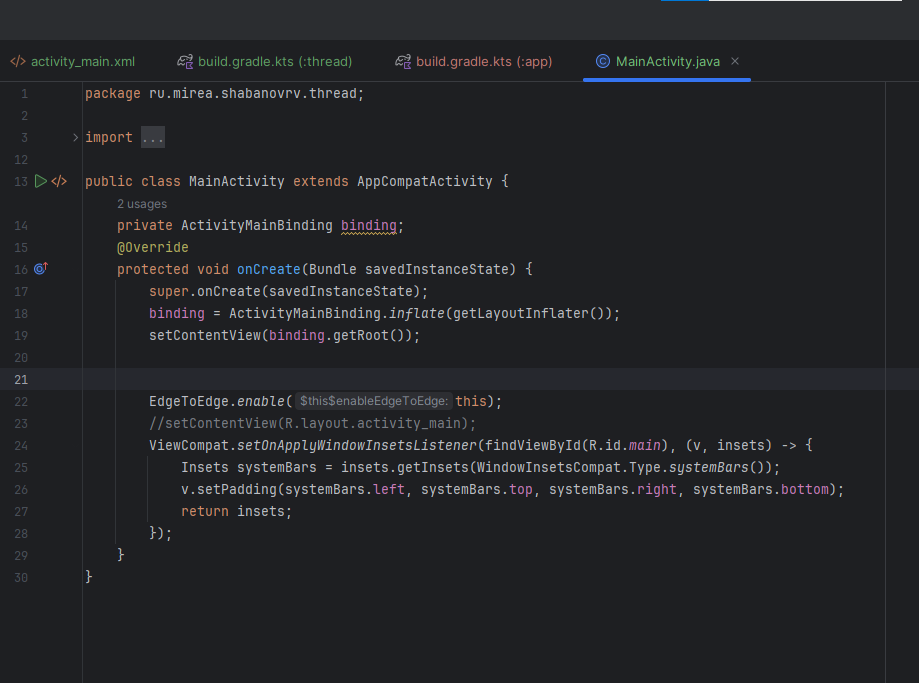
**Задание 2**

Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля «thread». На экране требуется разместить элементы «Button» и «TextView». Инициализацию графических компонентов осуществить с помощью «Binding». Посчитать в фоновом потоке среднее количество пар в день за период одного месяца. Общее количество пар и учебных дней вводятся в главном экране. Отобразить результат в «TextView».

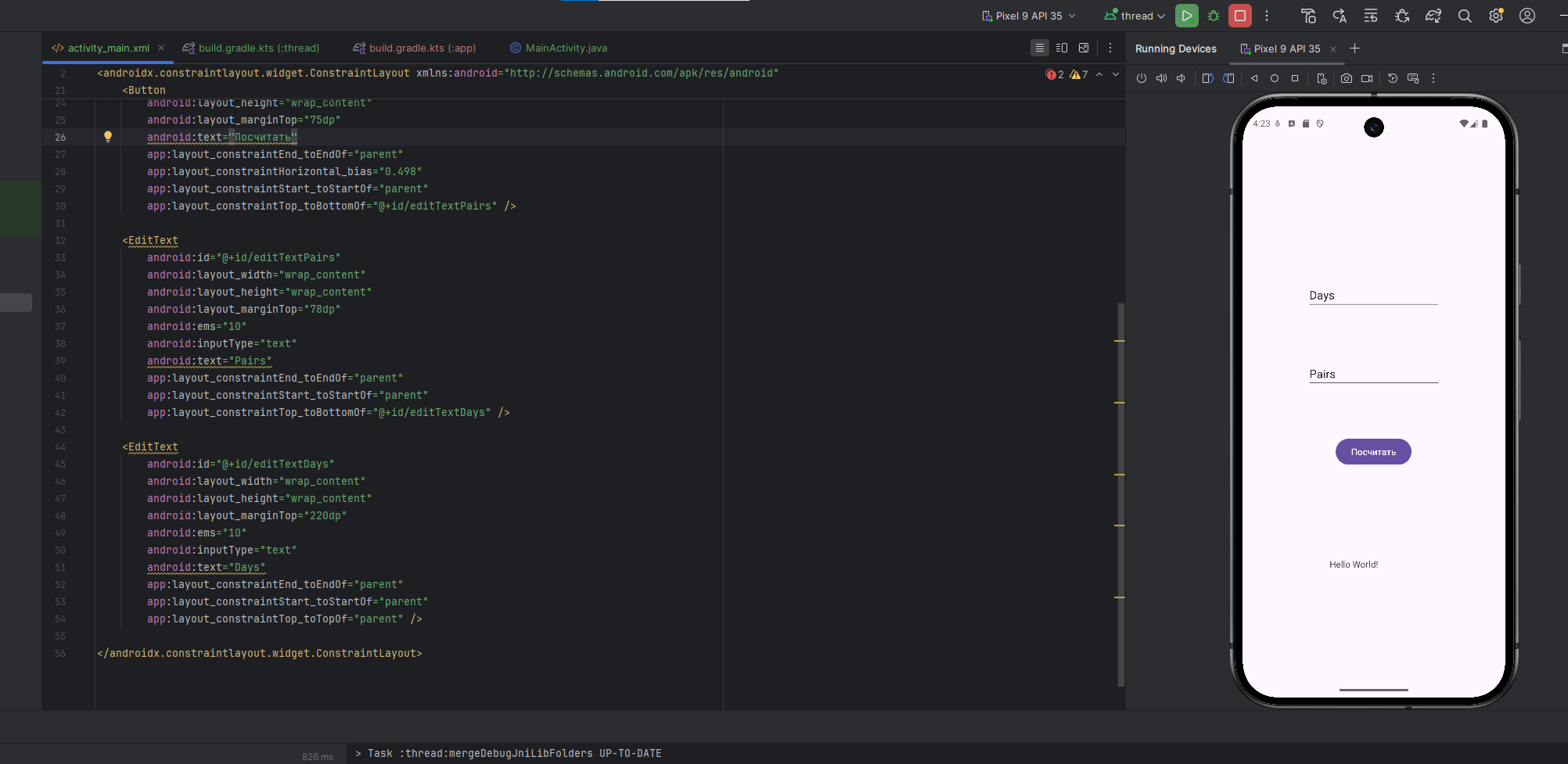
Создадим текст и кнопку:



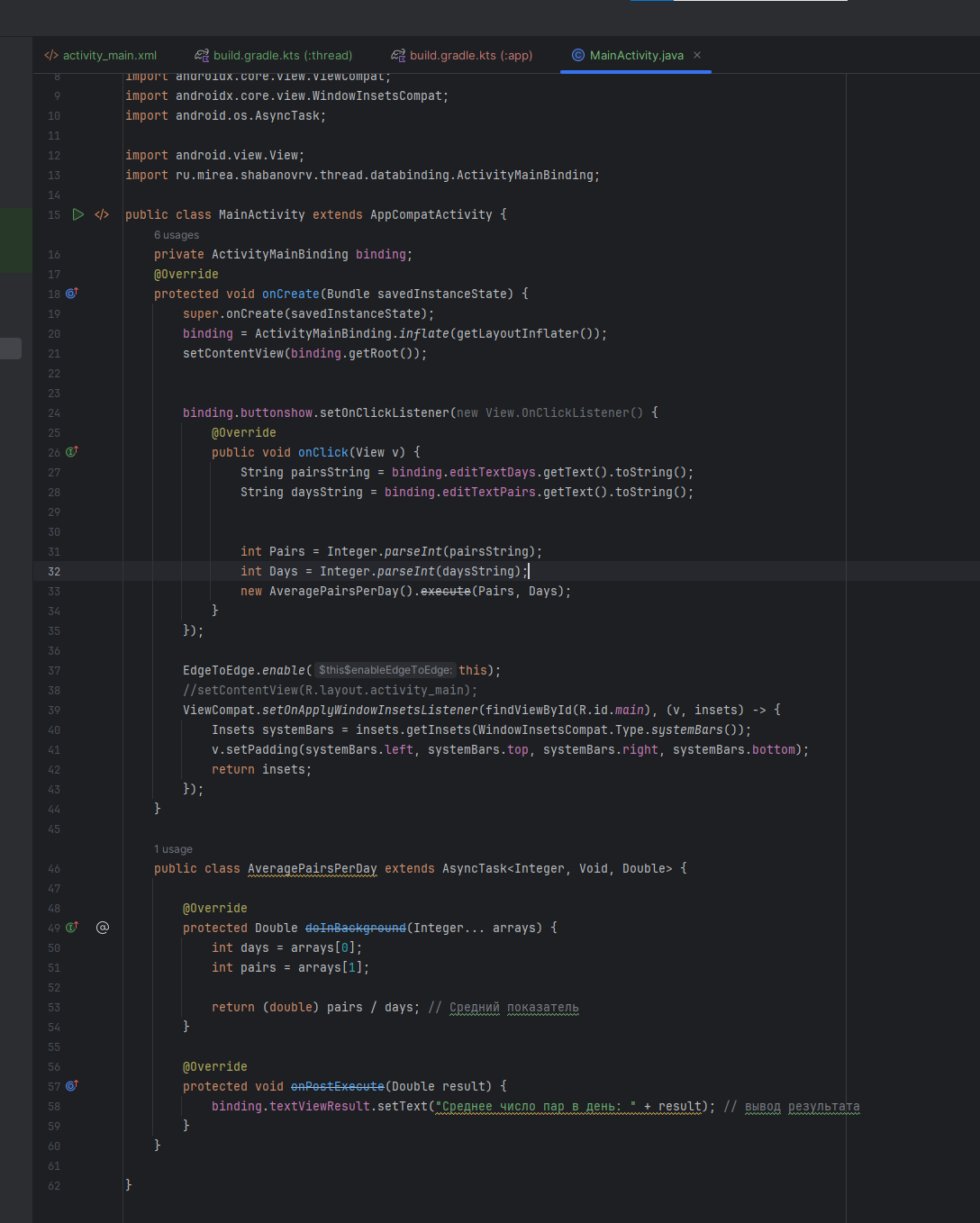
Создадим инициализацию через binding

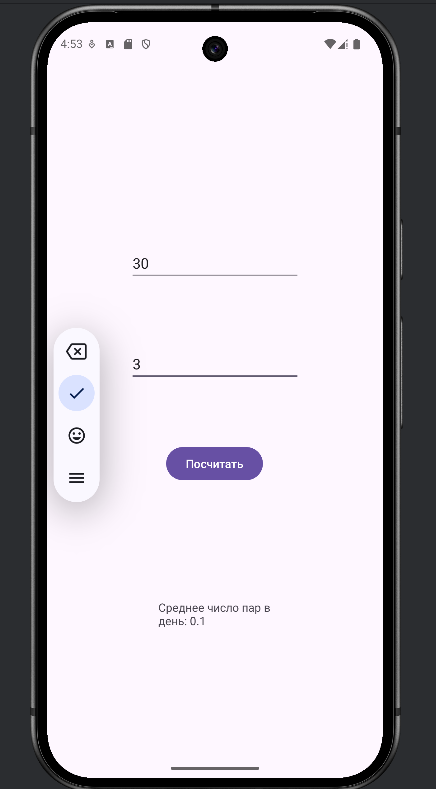


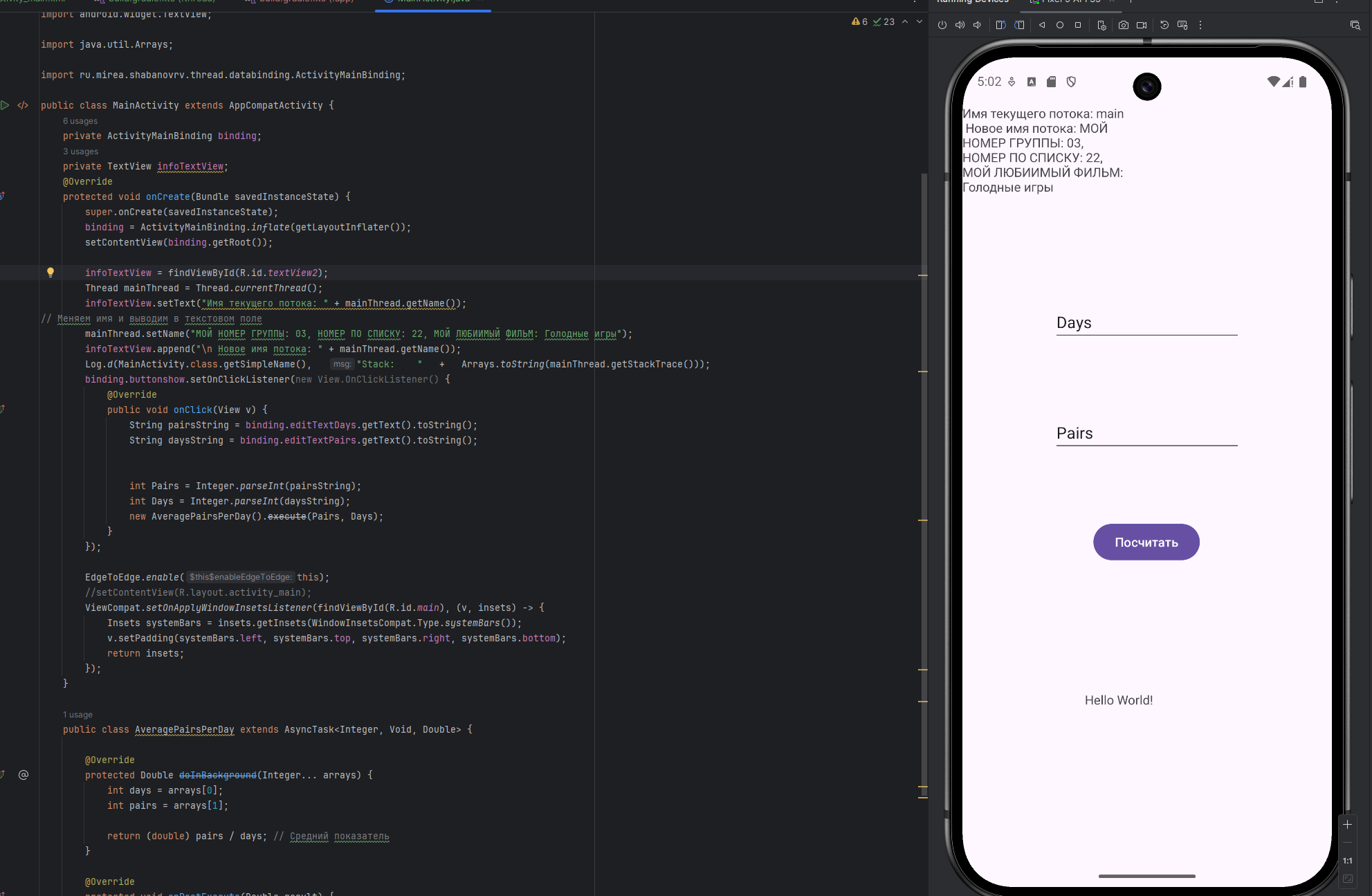
Добавим еще два поля: дни и пары



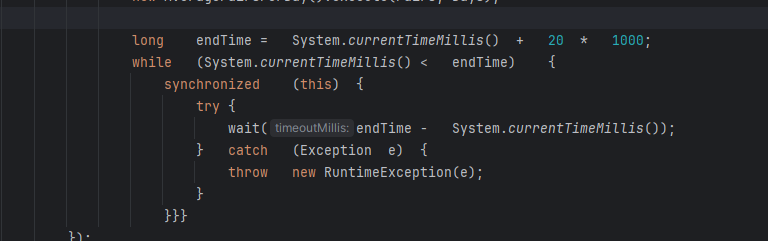
Реализуем логику расчета и вывода на экран по нажатию кнопки.



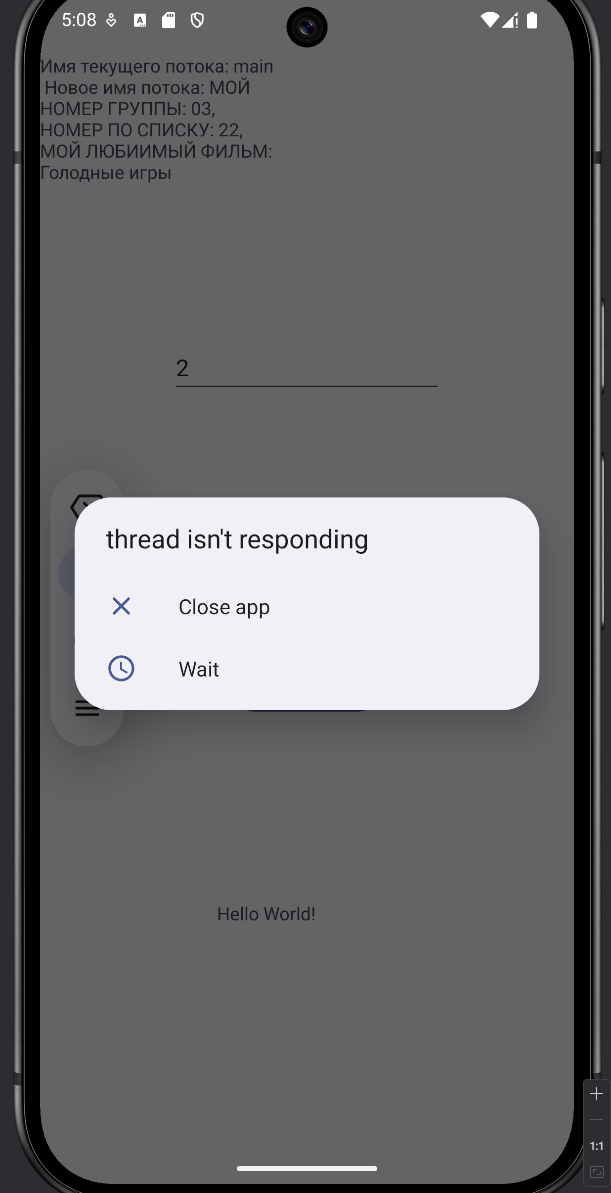


Выведем имя текущего потока и изменим его название 

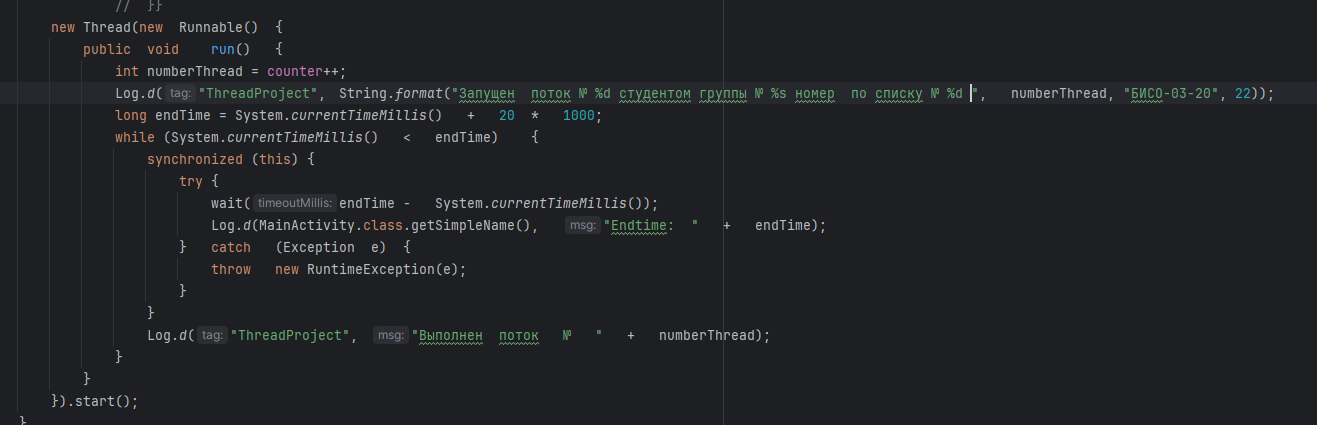
Для имитации вычислений в главном потоке требуется добавить следующий код в метод onClick, реагирующий на нажатие кнопки:



После многочисленных нажатий на кнопку приложение сломалось

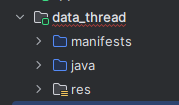


Требуется перенести функционал, замедляющий работу приложения, в отдельный поток. Для этого создаётся экземпляр класса «Runnable», у которого имеется метод «run». Далее создаётся объект «Thread», в конструкторе у которого указывается созданный «Runnable». После этого возможно запускать новый поток с помощью метода «start».



Задание 3

Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля: «data\_thread».



**Требуется определить в какой последовательности происходит запуск процессов.**

Сначала идет задержка в 2 секунды, потом выводится runn1, затем идет задержка в 1 секунду и одновременно выводится runn2 и запускается задержка в 2000 миллисекунд для вывода runn3

**Изучите методы «runOnUiThread», «postDelayed», «post».**

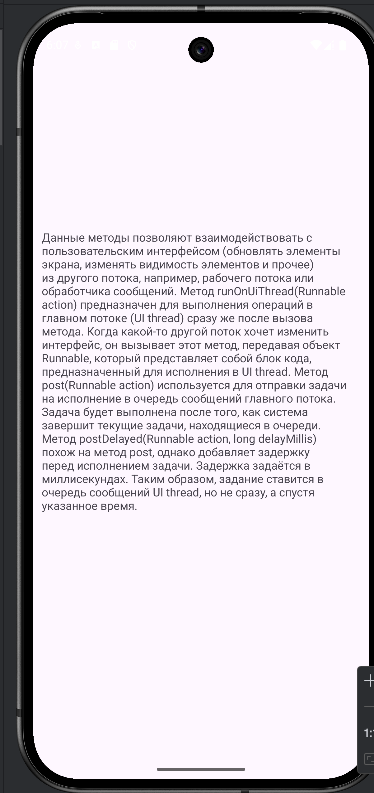
Данные методы позволяют взаимодействовать с пользовательским интерфейсом (обновлять элементы экрана, изменять видимость элементов и прочее) из другого потока, например, рабочего потока или обработчика сообщений.

Метод runOnUiThread(Runnable action) предназначен для выполнения операций в главном потоке (UI thread) сразу же после вызова метода. Когда какой-то другой поток хочет изменить интерфейс, он вызывает этот метод, передавая объект Runnable, который представляет собой блок кода, предназначенный для исполнения в UI thread.

Метод post(Runnable action) используется для отправки задачи на исполнение в очередь сообщений главного потока. Задача будет выполнена после того, как система завершит текущие задачи, находящиеся в очереди.

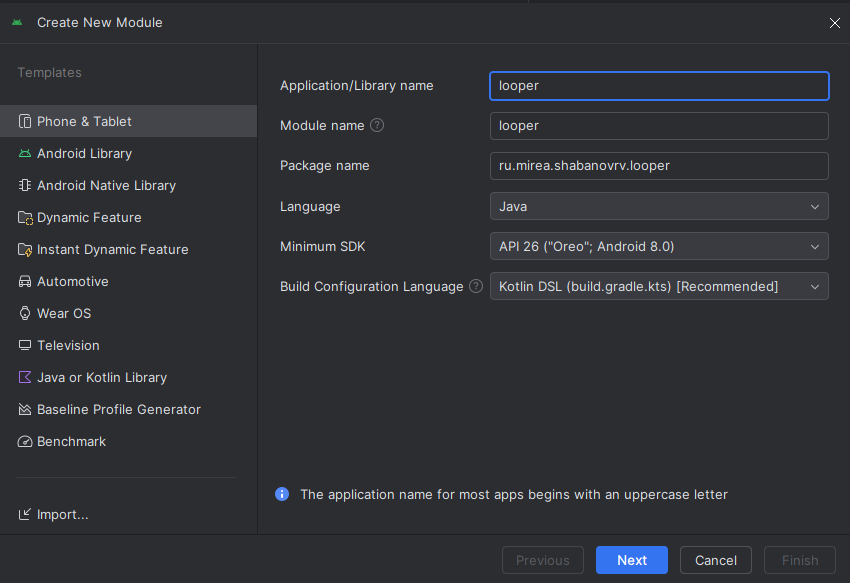
Метод postDelayed(Runnable action, long delayMillis) похож на метод post, однако добавляет задержку перед исполнением задачи. Задержка задаётся в миллисекундах. Таким образом, задание ставится в очередь сообщений UI thread, но не сразу, а спустя указанное время.

В «TextViwe» описать в чём различия между элементами и последовательность запуска. У элемента «TextView» имеется возможность установки значений строк: android:maxLines="10" android:lines="10"

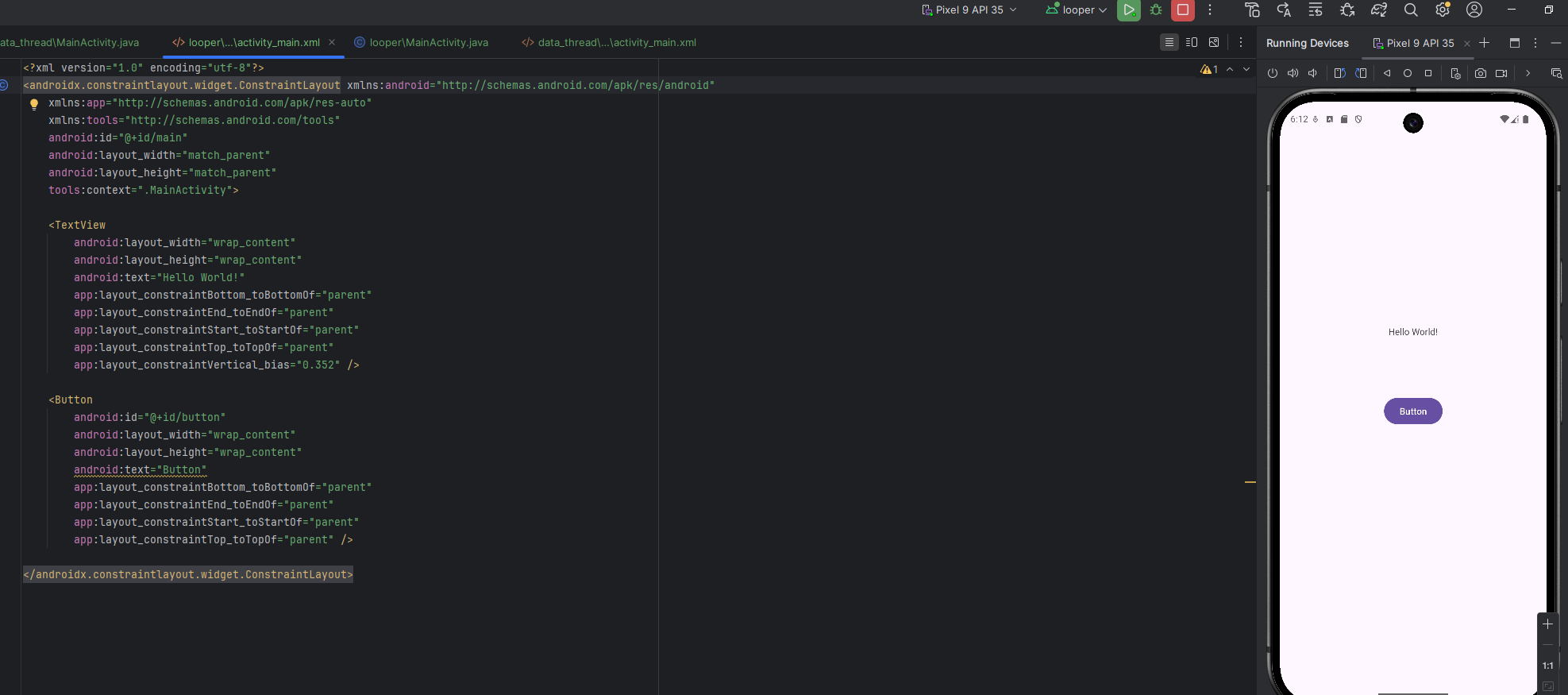


Задание 4

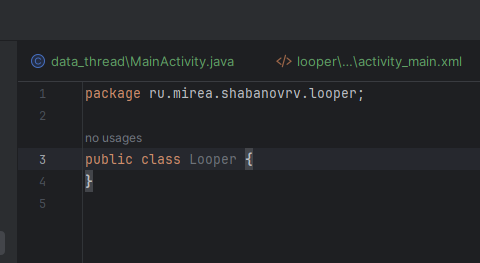
Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля: «looper». В main\_activity.xml требуется добавить «button» и реализовать обработку нажатия.



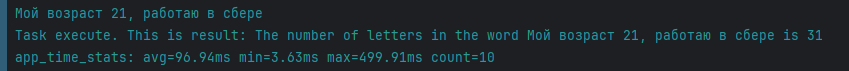
Создание кнопки:



Для создания класса требуется вызвать контекстное меню, нажав на директорию с кодом в папке «java» : «New | Java Class

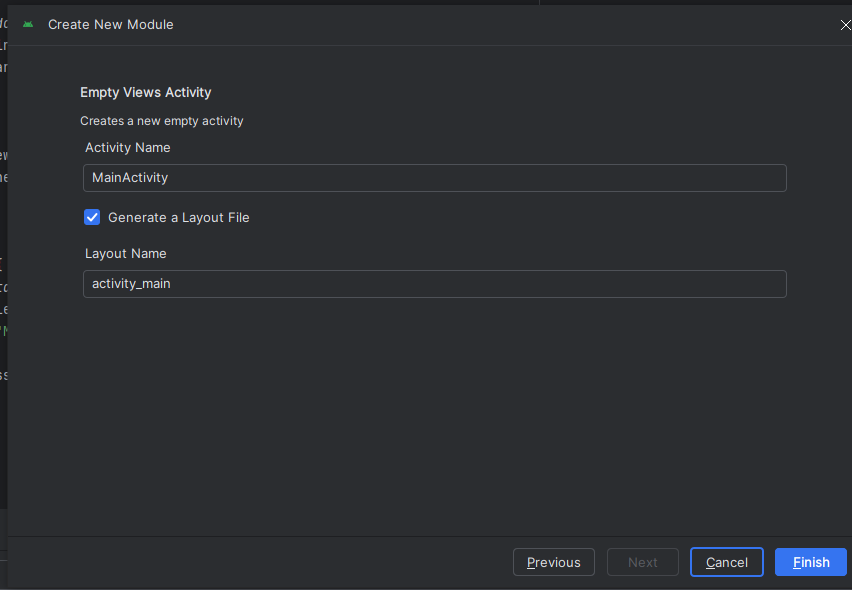


**Реализуйте пример, в котором вводится Ваш возраст и кем Вы работаете. Количество лет соответствует времени задержки. Результат вычисления осуществлять через Log.d.**

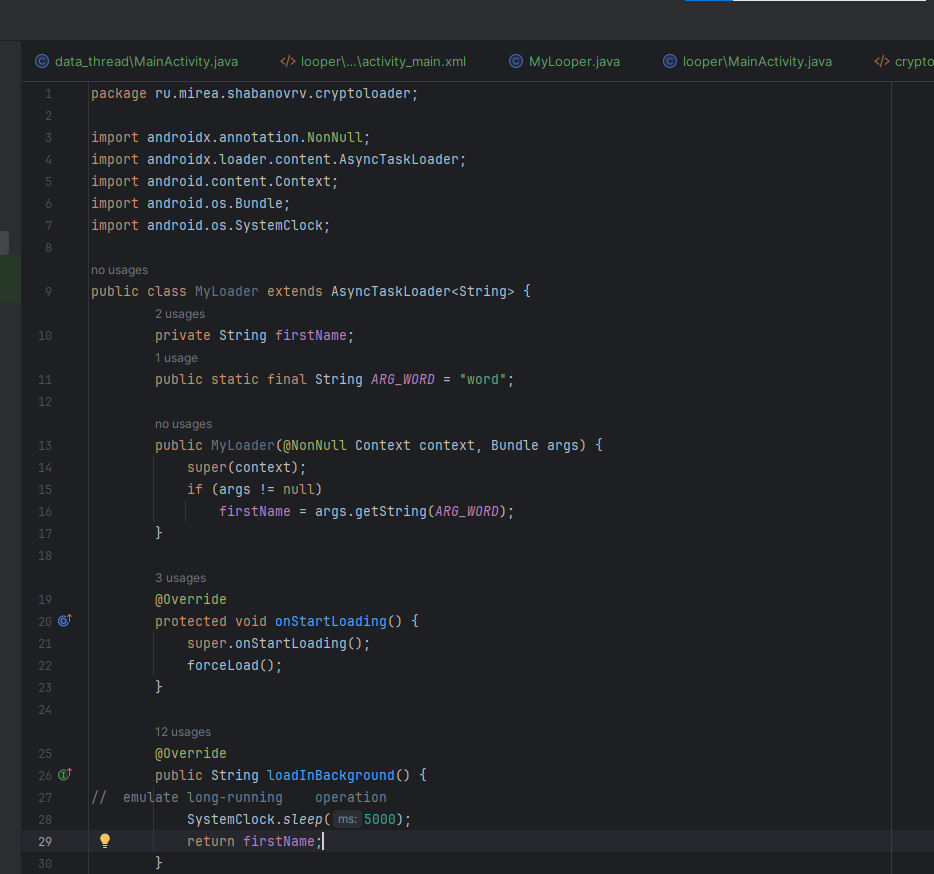


Задание 5

Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля: «CryptoLoader».

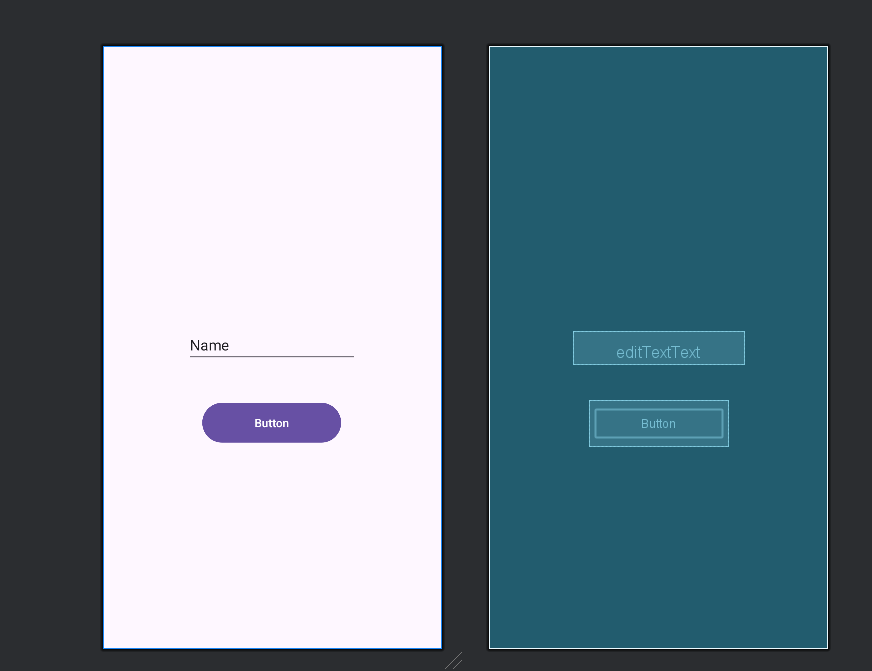


Для реализации собственного загрузчика, требуется создание класса и переопределение методов «loadInBackground» и «onStartLoading». Имя класса MyLoader («New->JavaClass->…»):

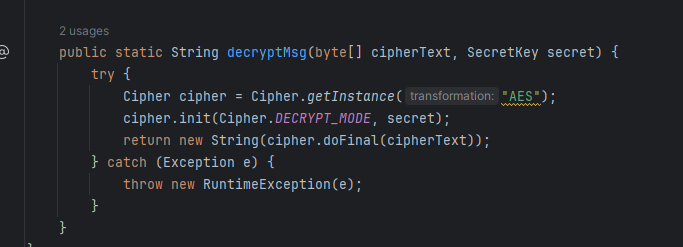


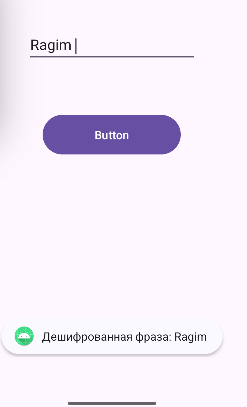
Задание 6

В созданном модуле требуется добавить элементы «EditText» и «Button». Пользователь вводит фразу в «EditText», далее она шифруется с помощью алгоритма AES и передается вместе с ключом в «Loader». В «Loader» происходит дешифровка фразы и последующая передача в «MainActivity». Дешифрованная фраза отображается с помощью «toast» или «snackBar».





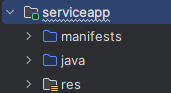




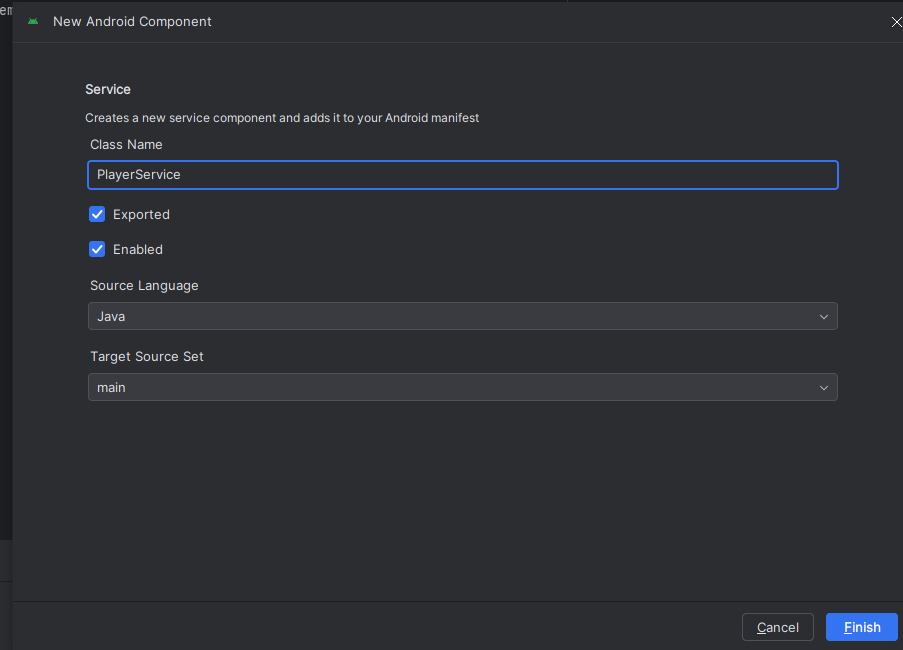
**Задание 7**

Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля: «ServiceApp».

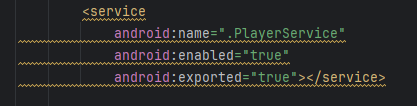
Создание модуля



Создание сервиса



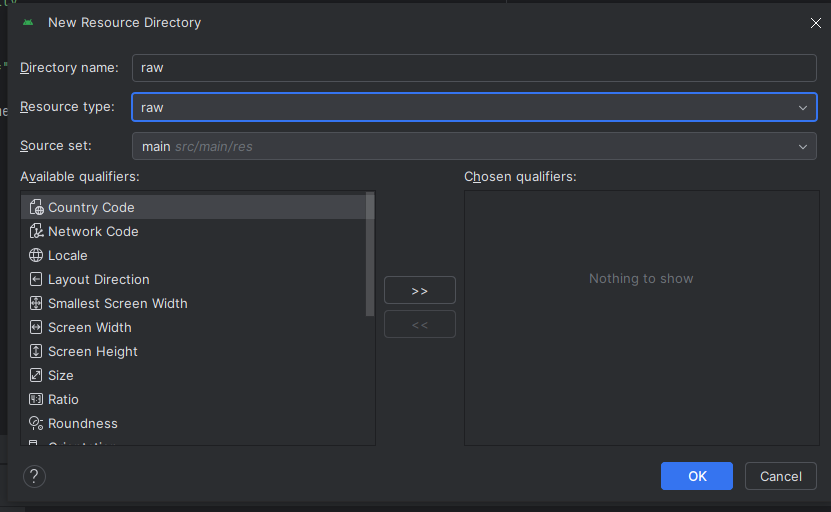
Проверка манифест файла



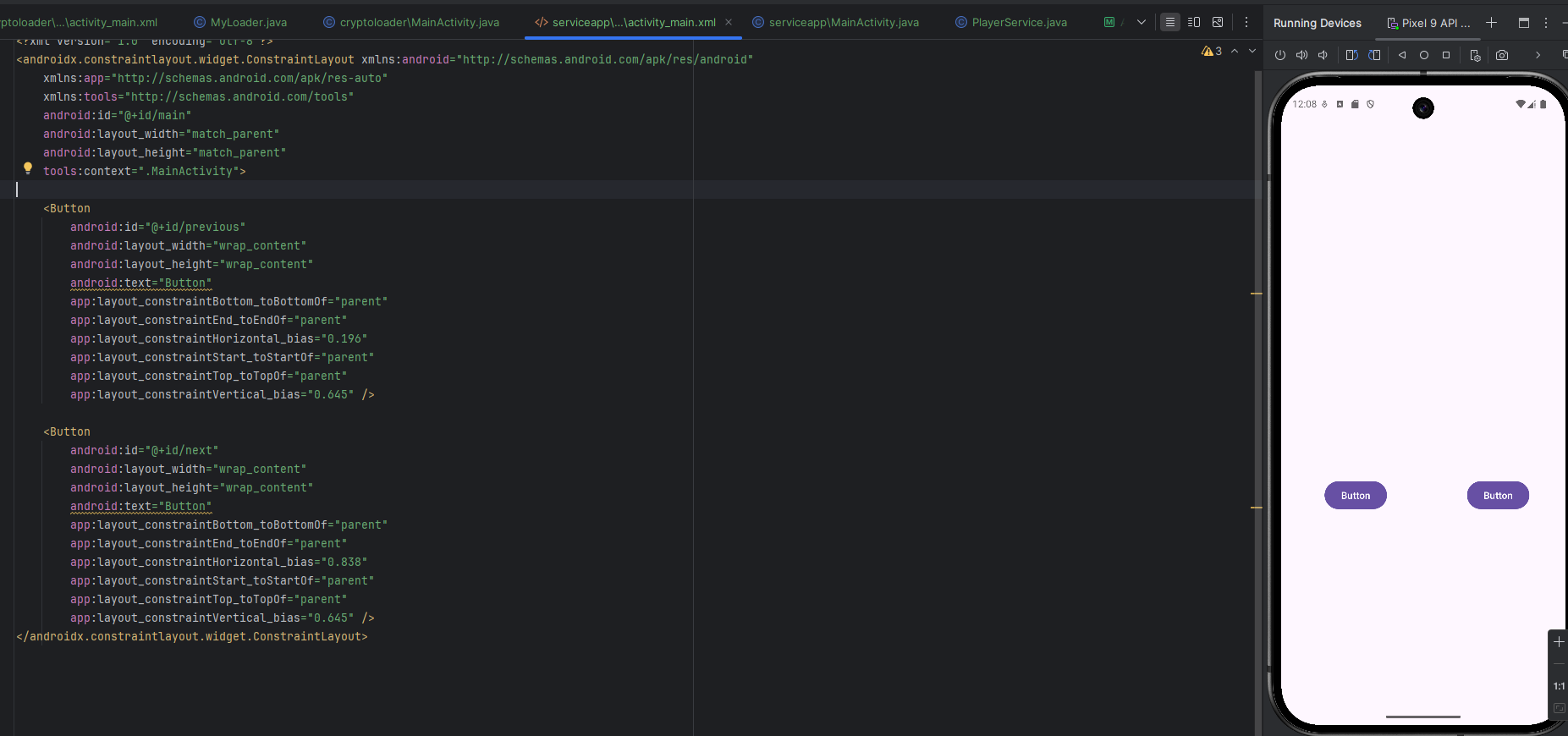
**Задание 8**

**Требуется добавить функционал воспроизведения аудиофайла. В первую очередь необходимо добавить медиа файл (типа «.mp3» и т.д.) в ресурсы. В активности добавить две кнопки «button» для воспроизведения и остановки музыкальных композиций/композиции. Придумать собственный дизайн данного проигрывателя.**

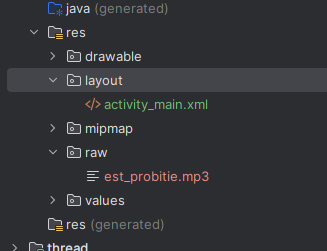
**Создание директории**



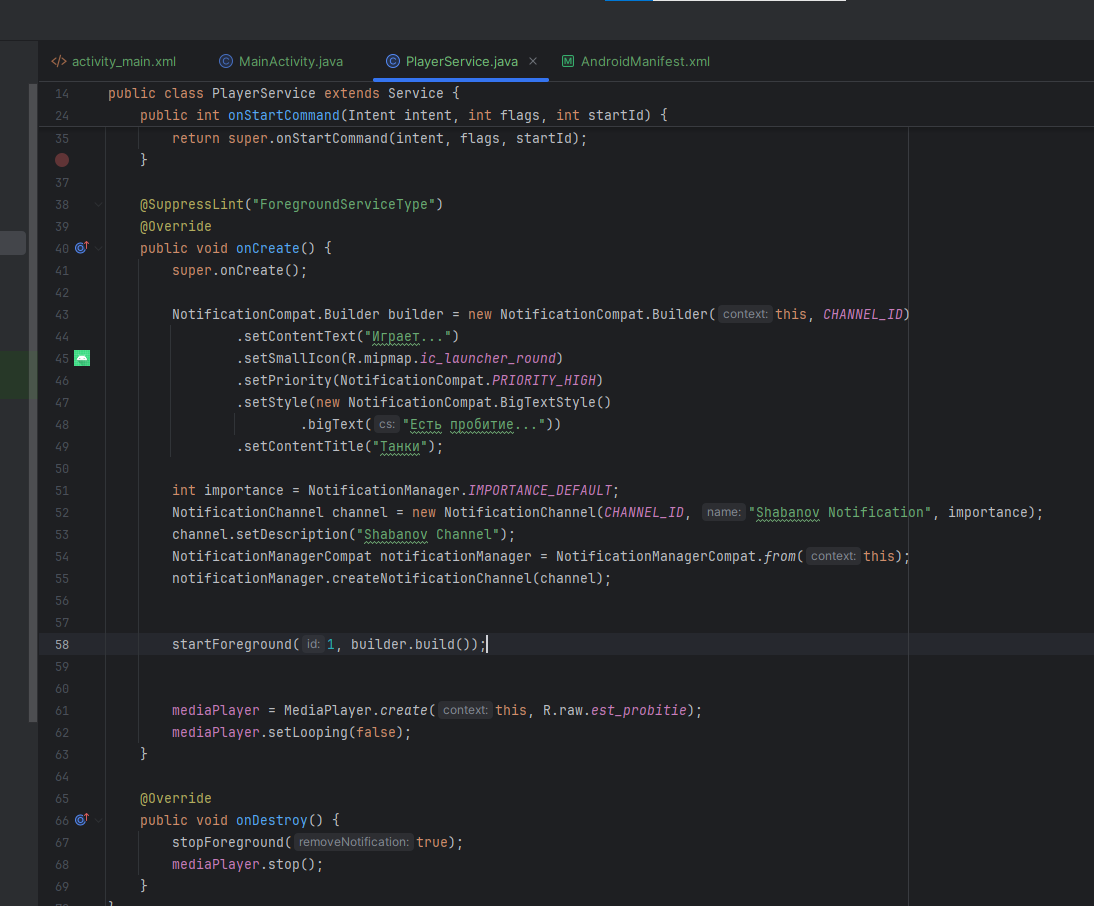
Создадим кнопки

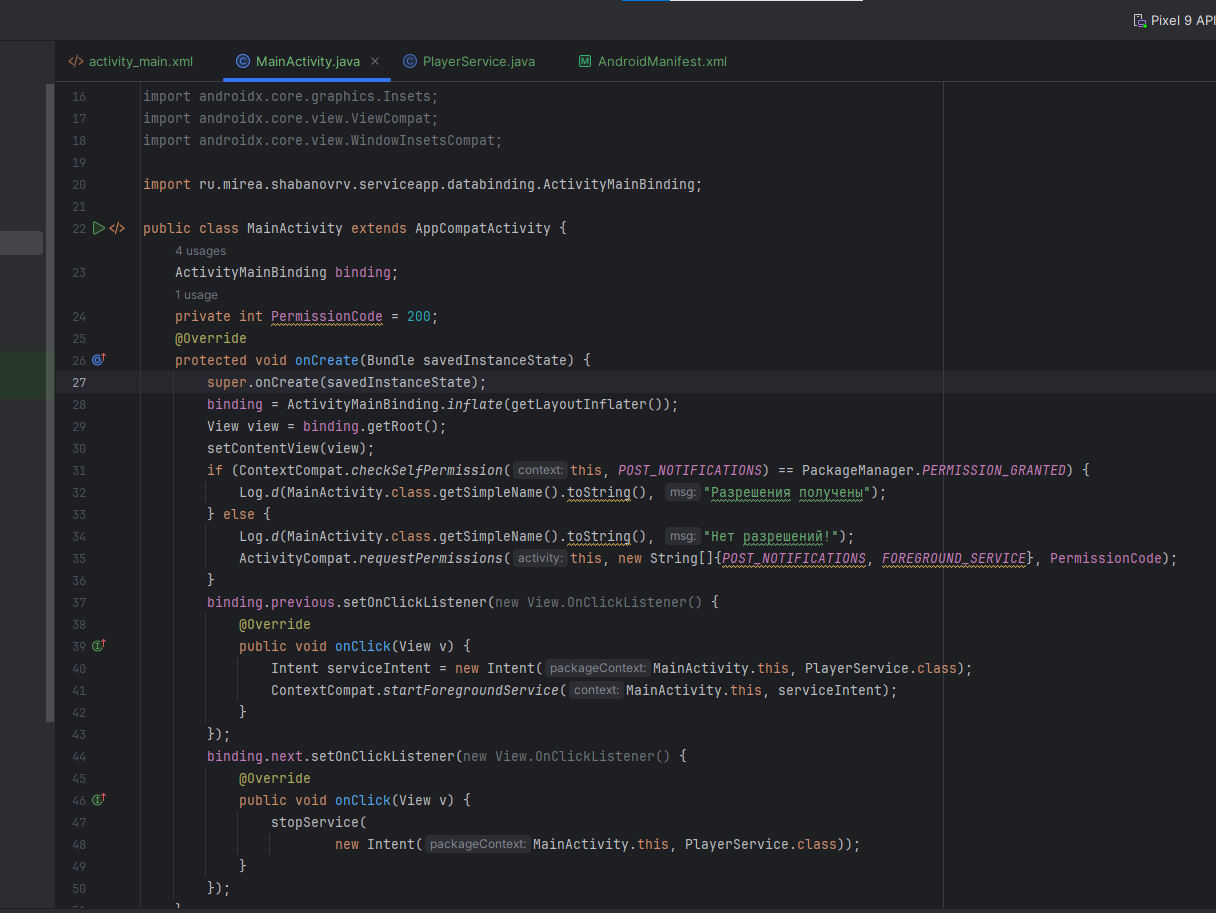


И перекинем mp3 файл в директорию

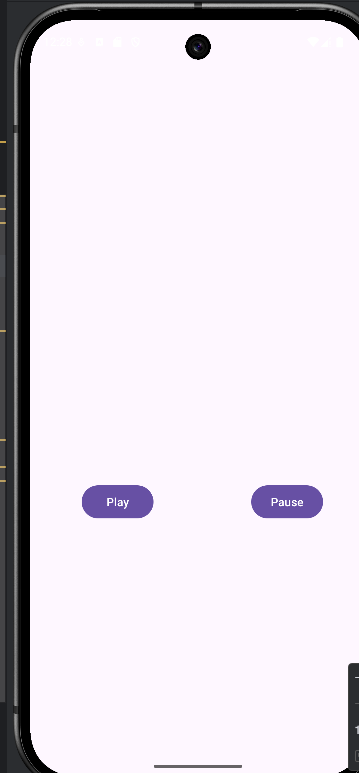


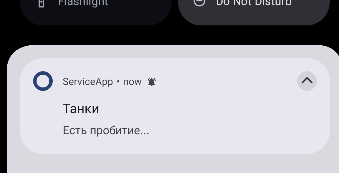
Пропишем логику





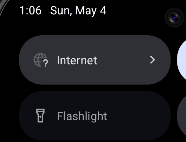
Получим



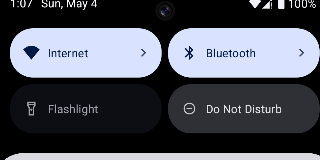


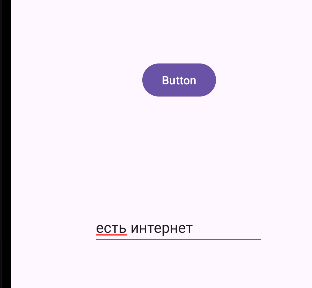
**Задание 9**

Создать новый модуль. В меню «File | New | New Module | Phone & Tablet Module | Empty Views Activity». Имя модуля: «WorkManager». Добавить в пример критерии запуска: напр. наличие интернета.









**КОНТРОЛЬНОЕ ЗАДАНИЕ**

**В проекте «MireaProject» создать отдельный фрагмент выполнения фоновой задачи и реализовать её выполнение с помощью применения механизма «Worker», либо одной из разновидностей сервисов.**

Реализуем проверку интернета с помощью создания отдельного фрагмента, с использованием Worker

