**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Чернышева Ксения Юрьевна

Программирование Мобильных и Встраиваемых Систем

Отчет по лабораторной работе №5

«Разработка мобильных

приложений с определением местоположения»

студентки 3 курса 12 группы

**Преподаватель**

**Давидовская Мария Ивановна**

**Минск 2024**

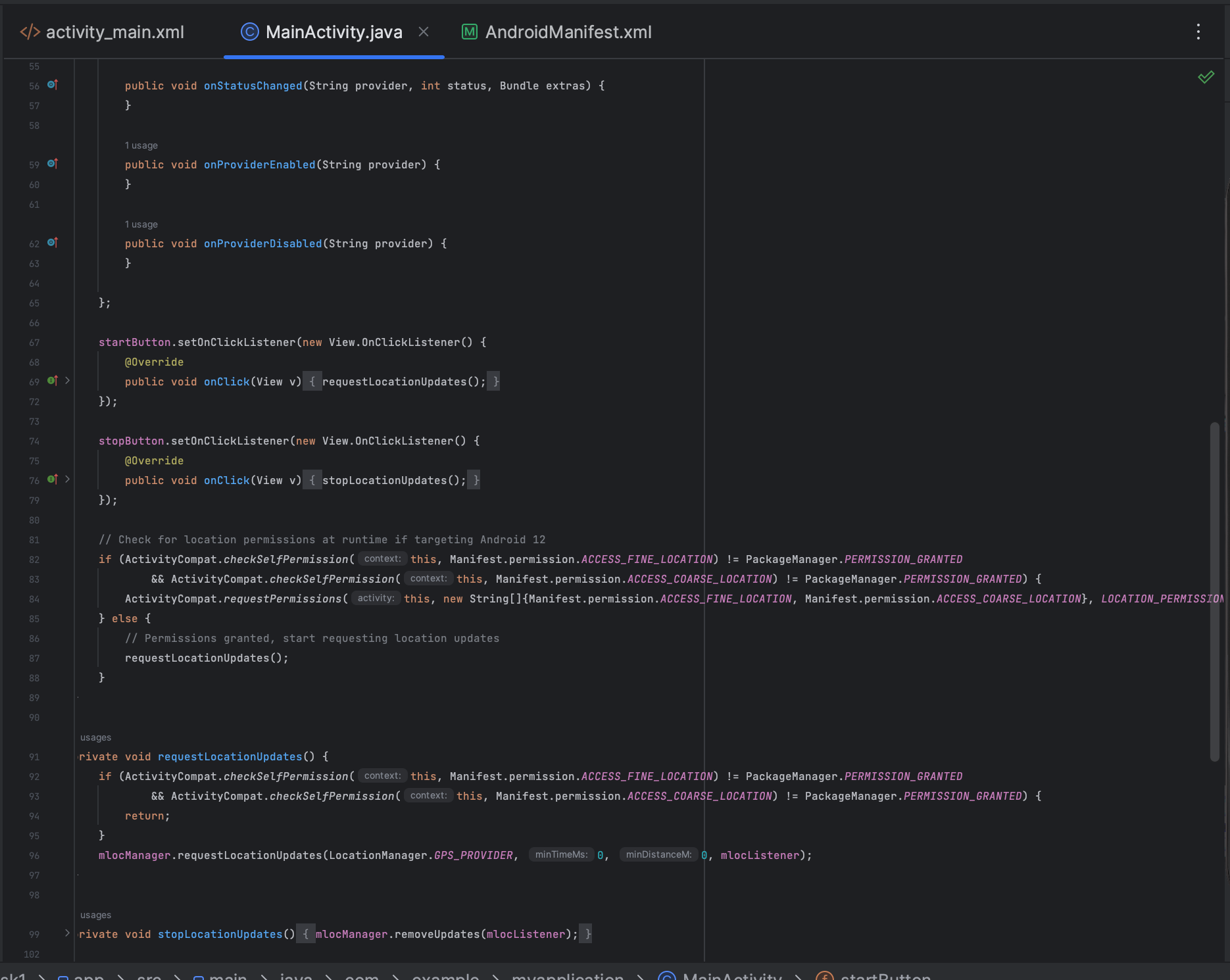
**Цели работы:**

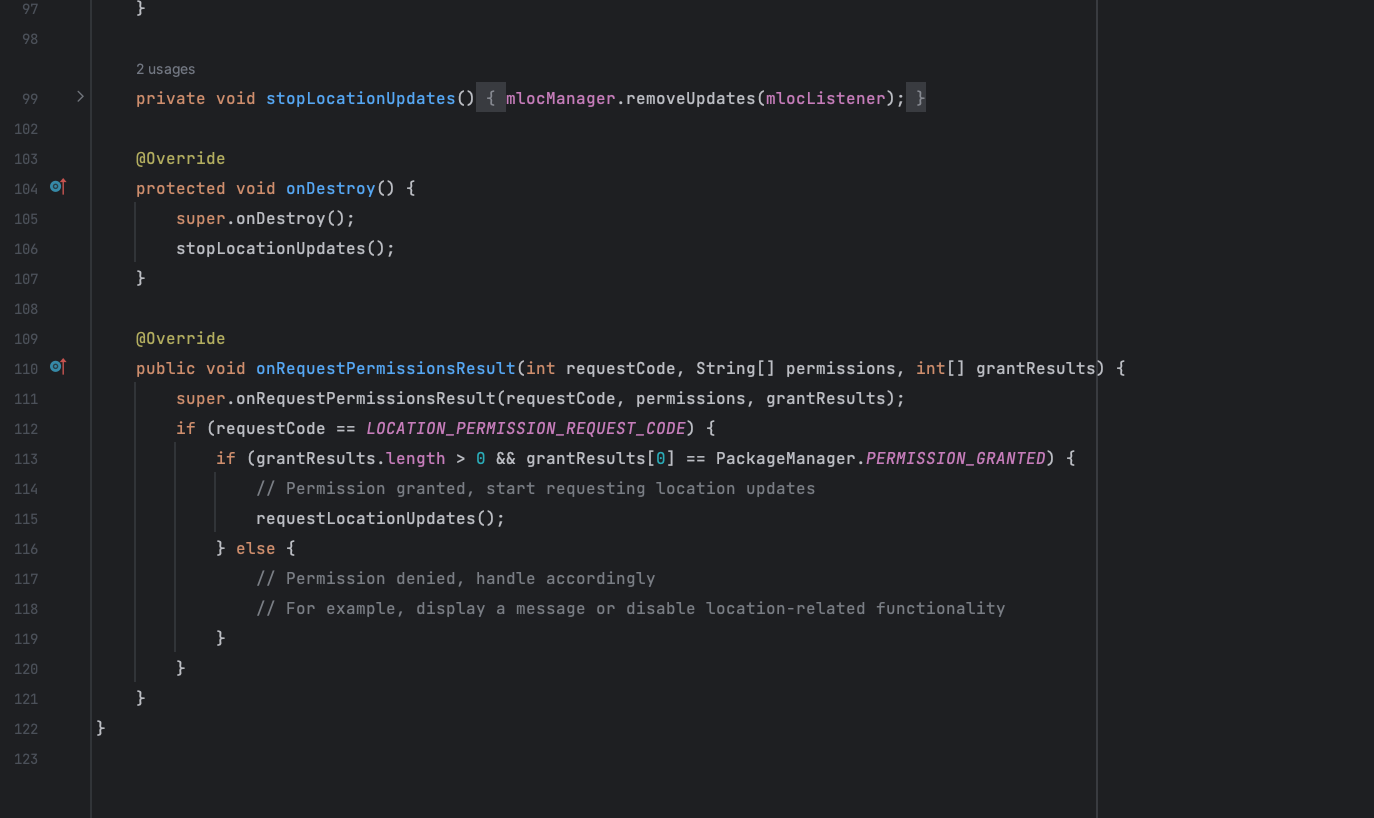
* разработка приложения, демонстрирующего возможности определения местоположения.

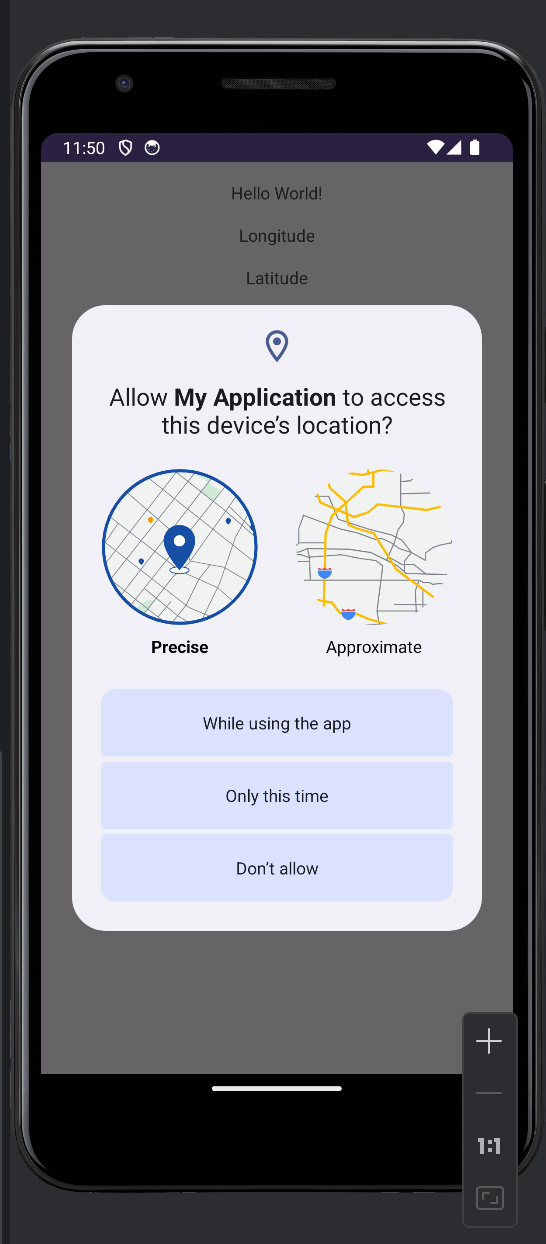
**Задание 1**

Реализовать на Java приложение RunTracker, которое позволяет отслеживать текущее местоположение и имеет две управляющие кнопки в интерфейсе. Фрагменты коды приложения представлены в разделе 5.1 текущей лабораторной работы. Пример интерфейса представлен на рисунке на странице 3 раздела 5.1. Пользовательский интерфейс отвечает за вывод простых данных о текущем местоположении. В нем имеются кнопки для запуска и остановки текущей серии. В качестве разметки использовать ConstraintLayout.









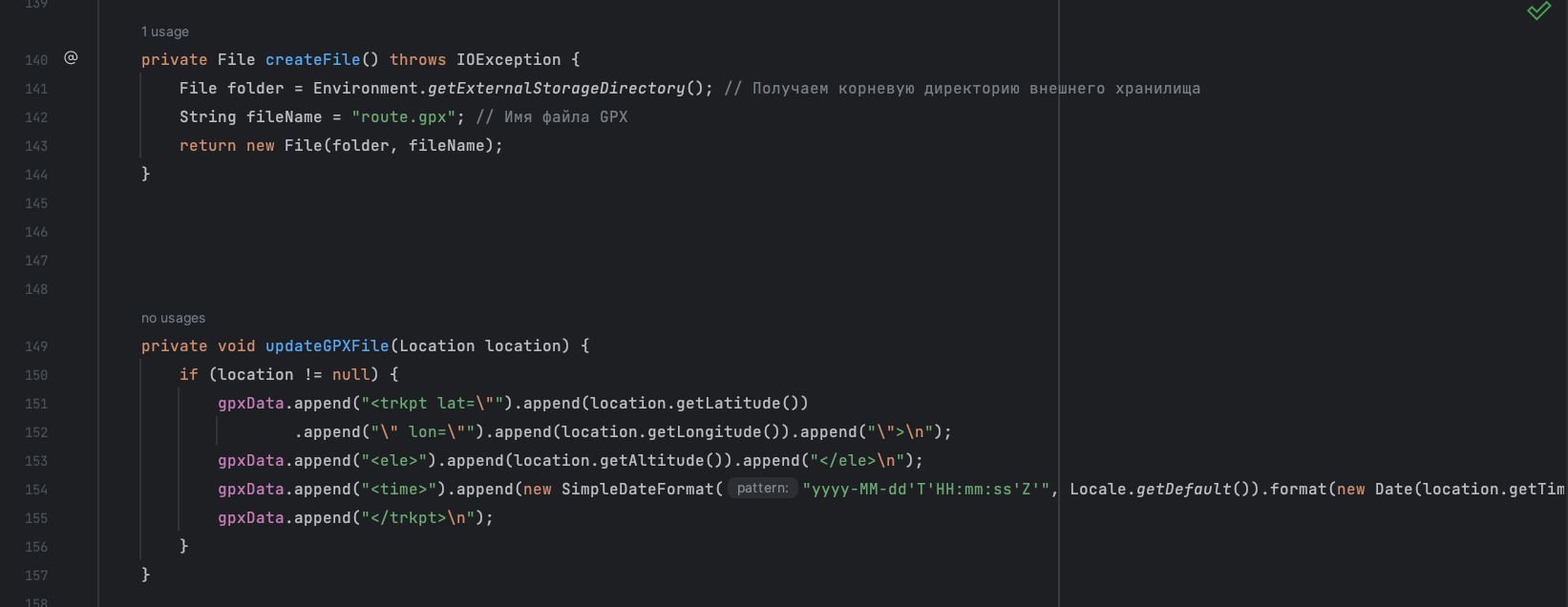
**Задание 2**

Внести изменения в приложение, реализованное в задании 1, используя один из двух вариантов хранения данных маршрута для передачи на виртуальное устройство:

а) GPX-файл

б) KML-файл

Код приложения аналогичен заданию 1 + метод генерации файла:



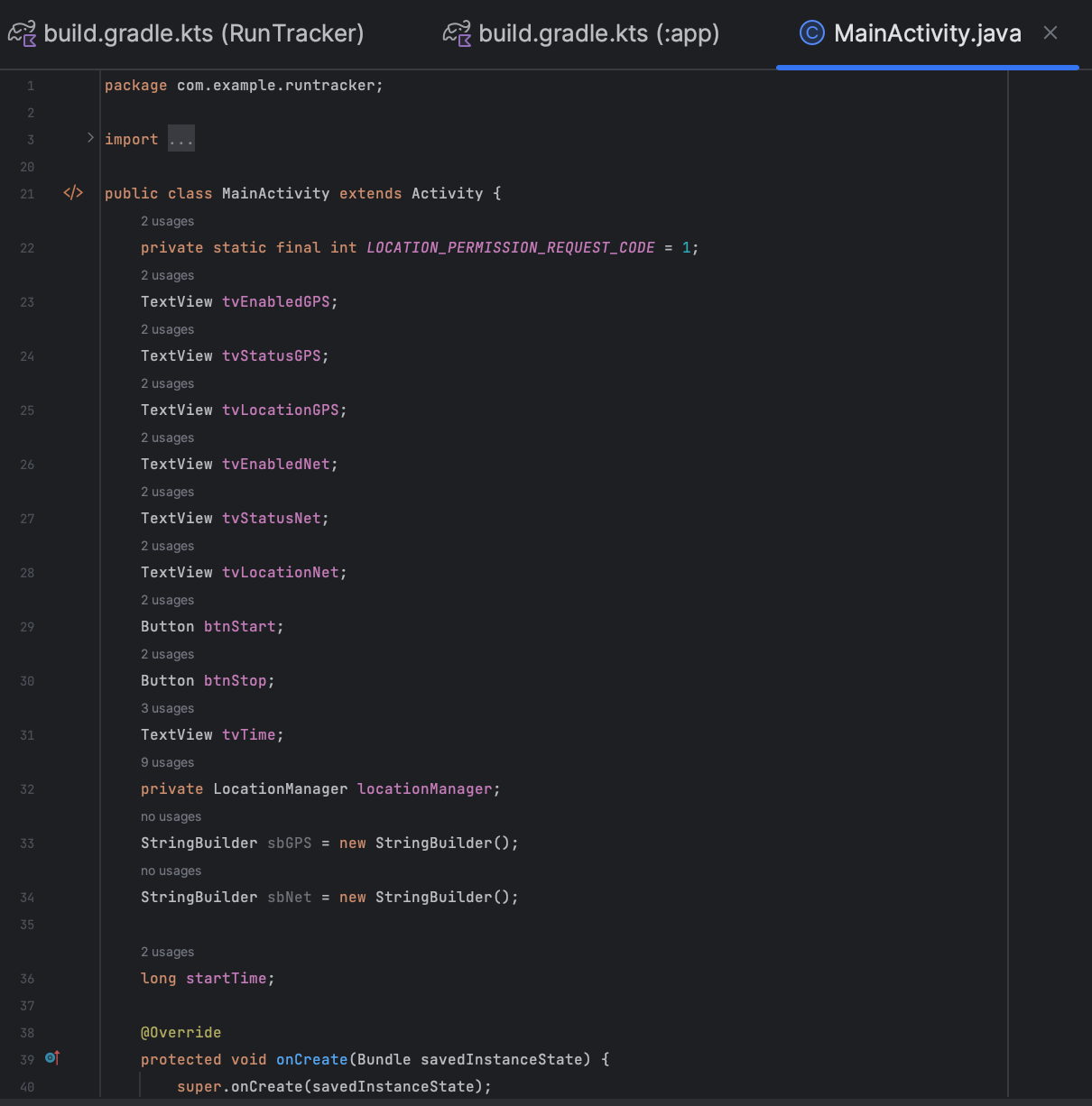
**Задание 3**

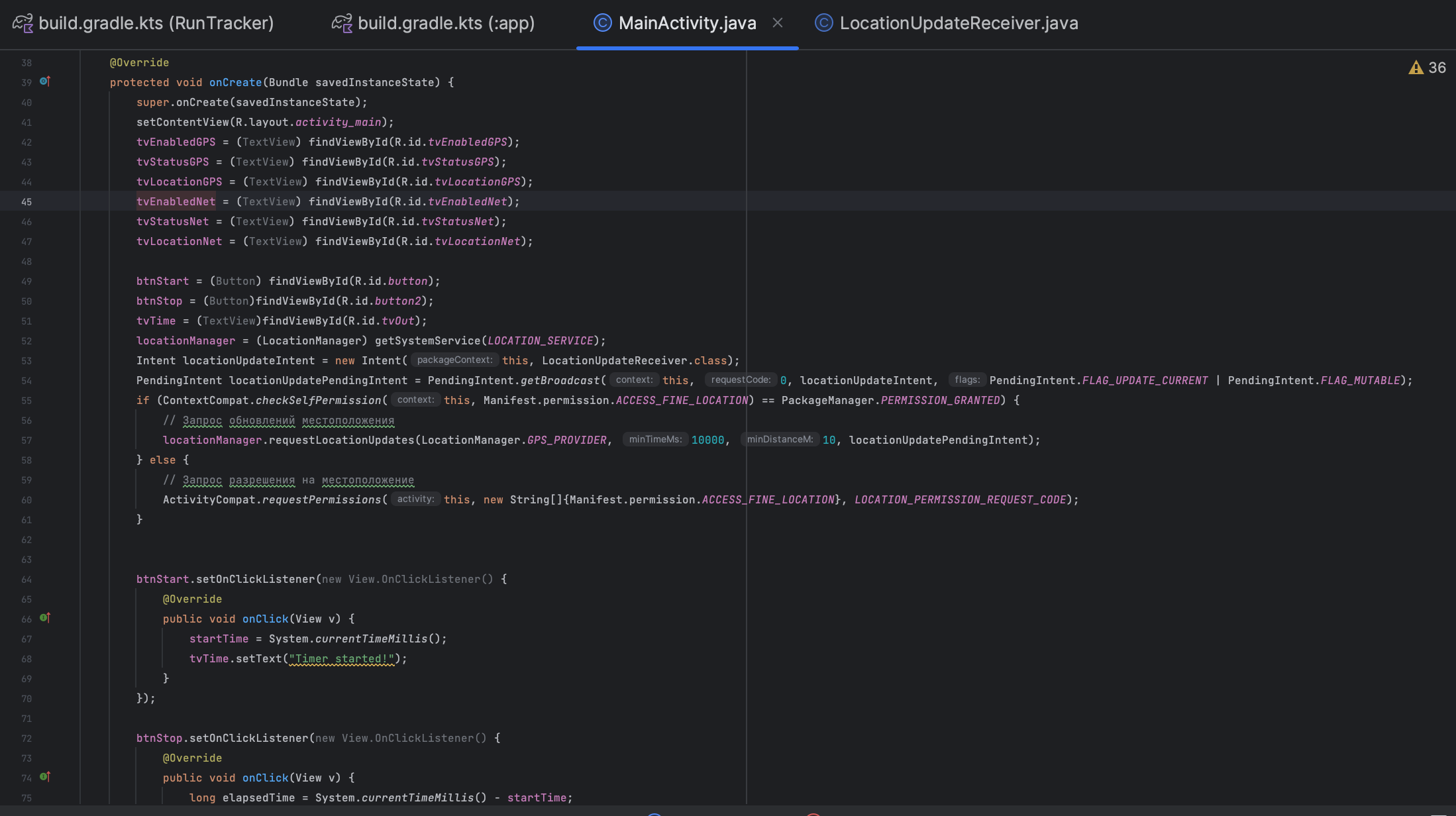
Реализовать приложение RunTracker из задания 1, изменив разметку на TableLayout и LinerLayout и применив PendingIntent API (https://developer.android.com/reference/android/app/PendingIntent?hl=en).

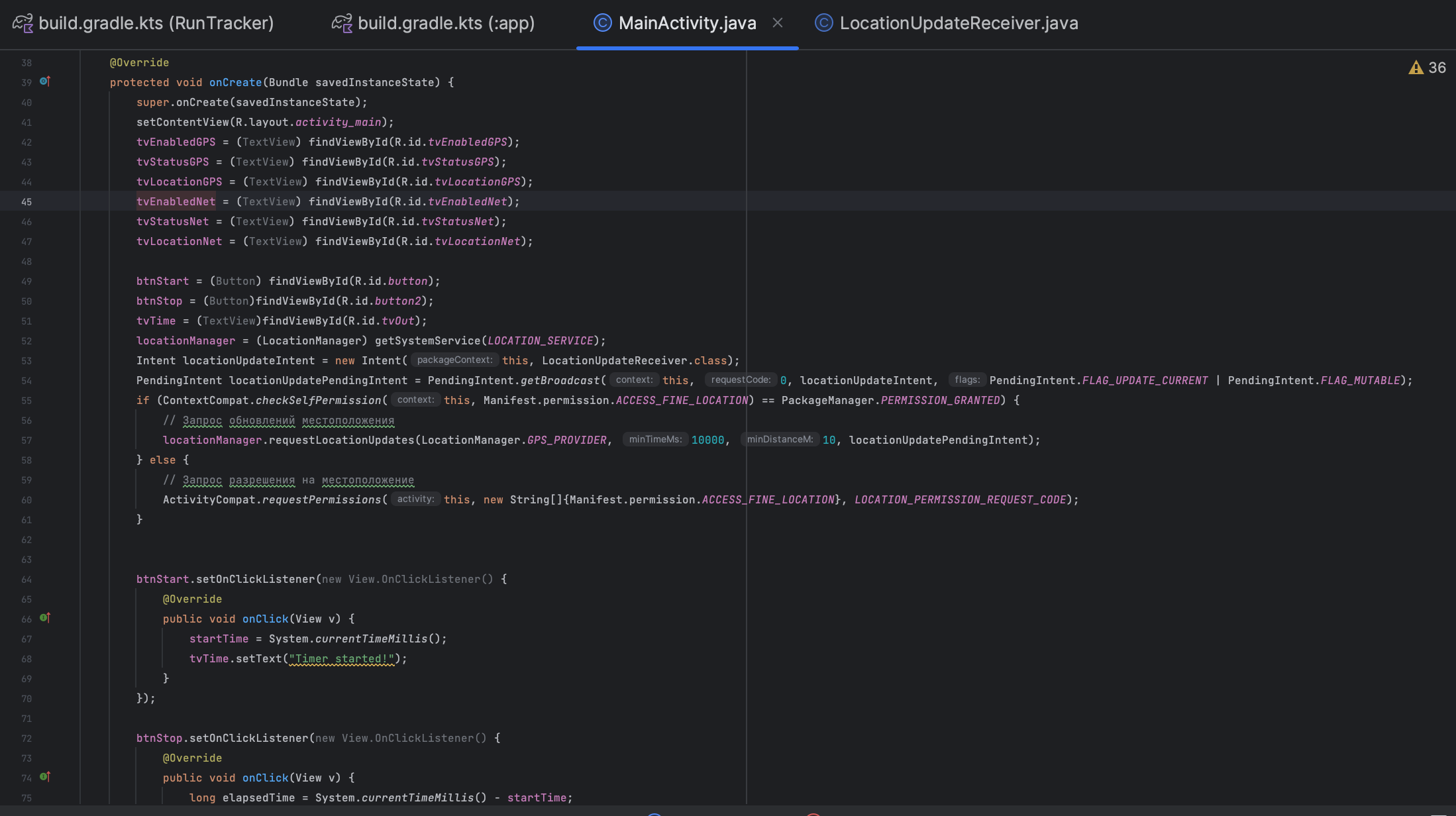
В макете использовать виджет TableLayout, чтобы разместить элементы интерфейса на экране. TableLayout состоит из пяти виджетов TableRow и одного LinearLayout . Каждый виджет TableRow содержит два виджета TextView : в первом выводится метка, а второй заполняется данными во время выполнения. LinearLayout содержит два виджета Button.

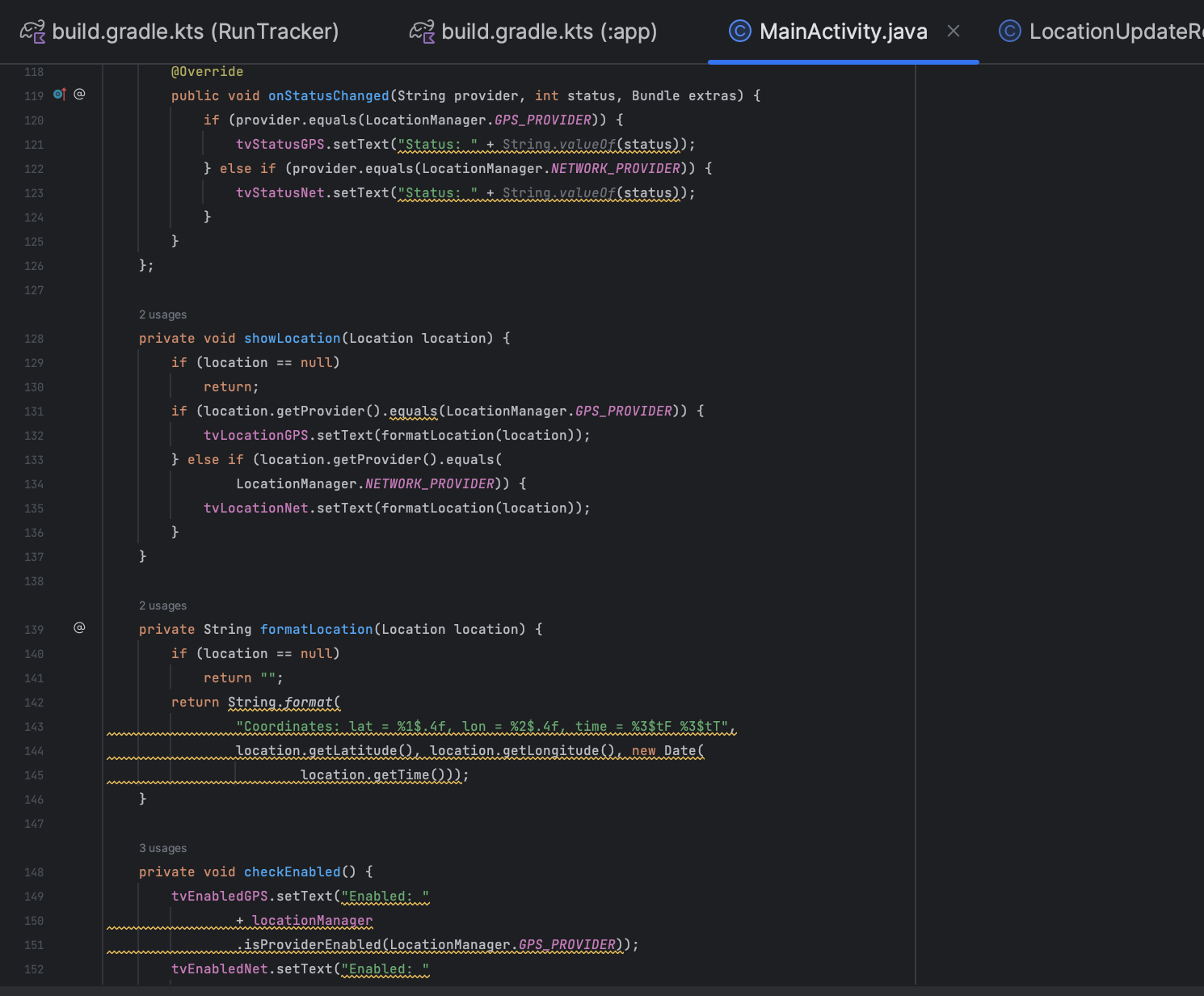
Для доставки обновлений о метостоположении в системную службу

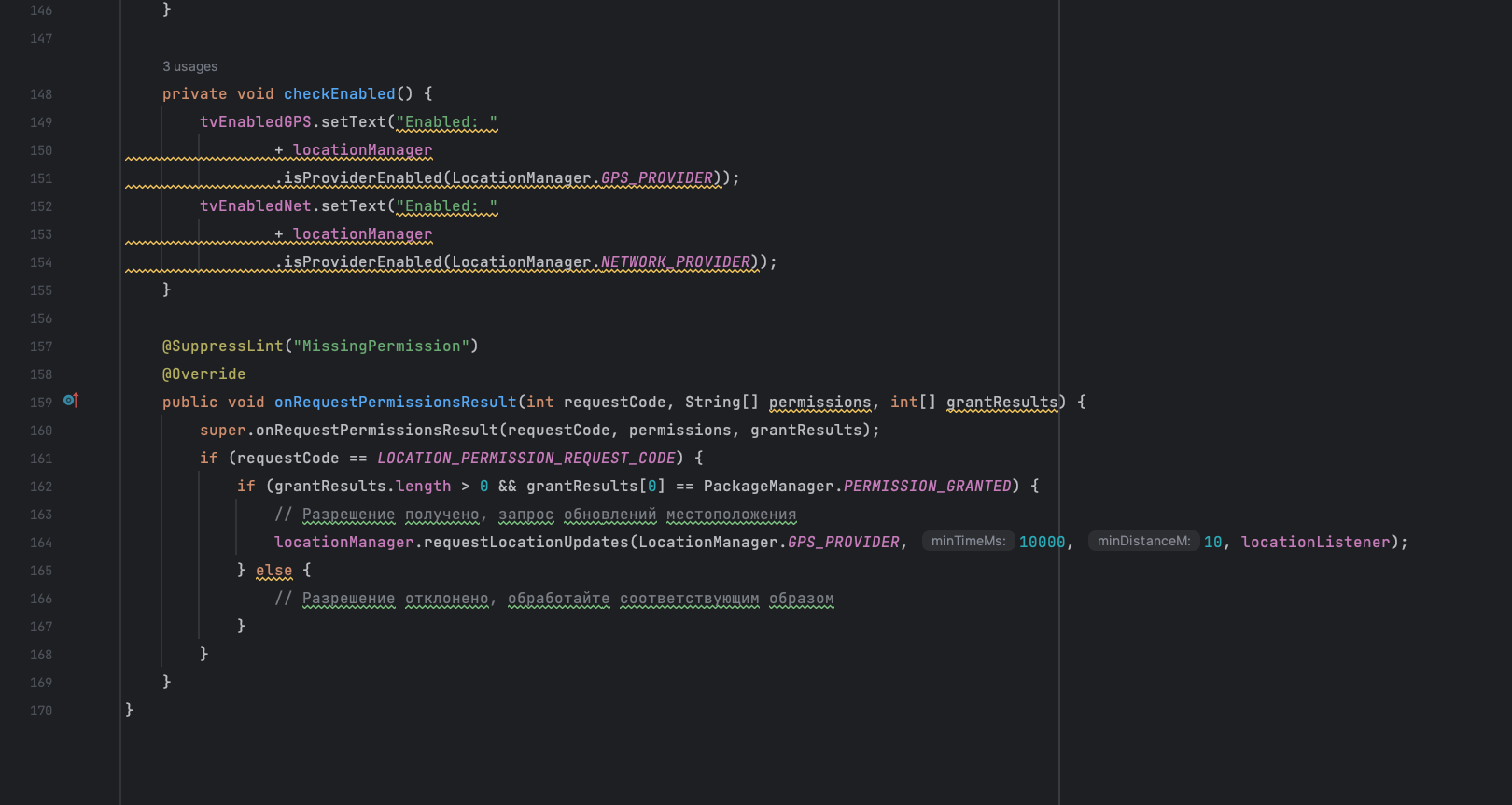
LocationManager использовать PendingIntent API.

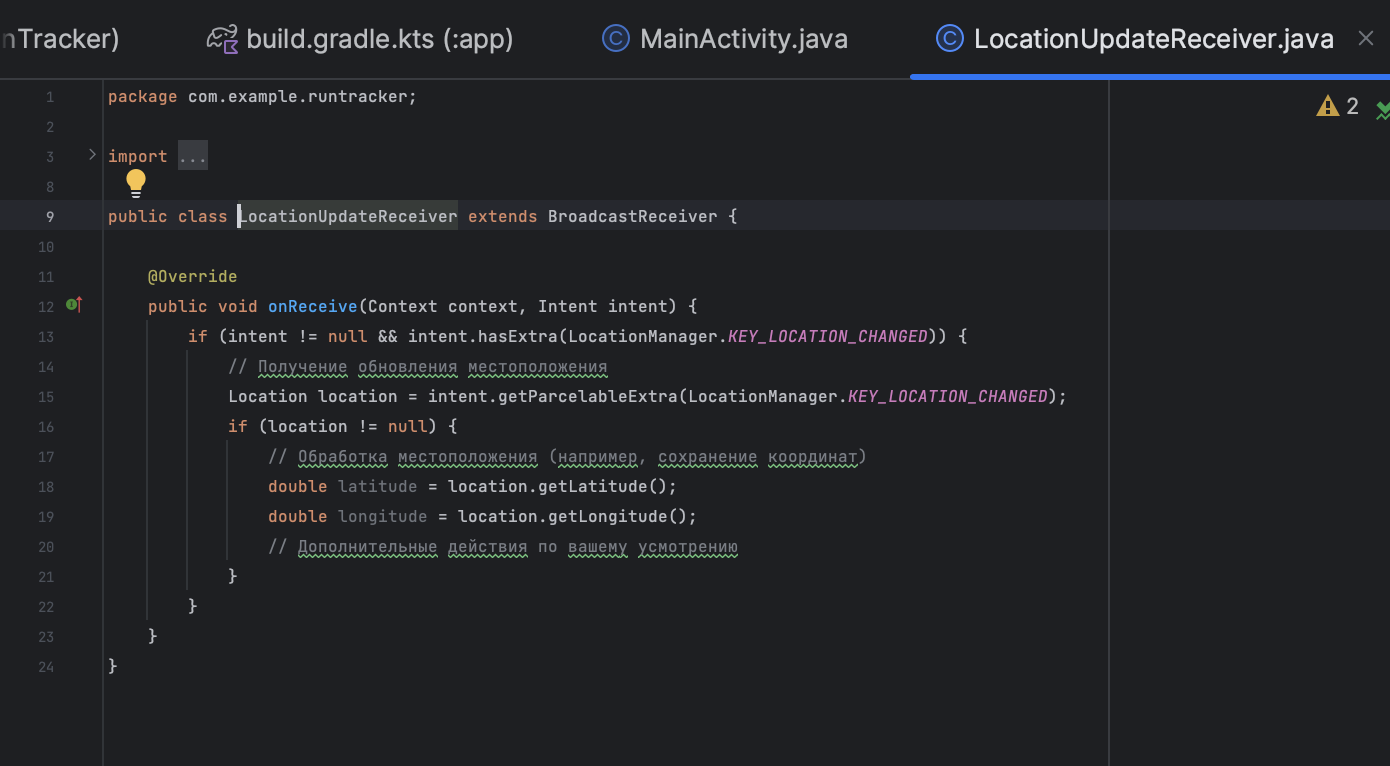


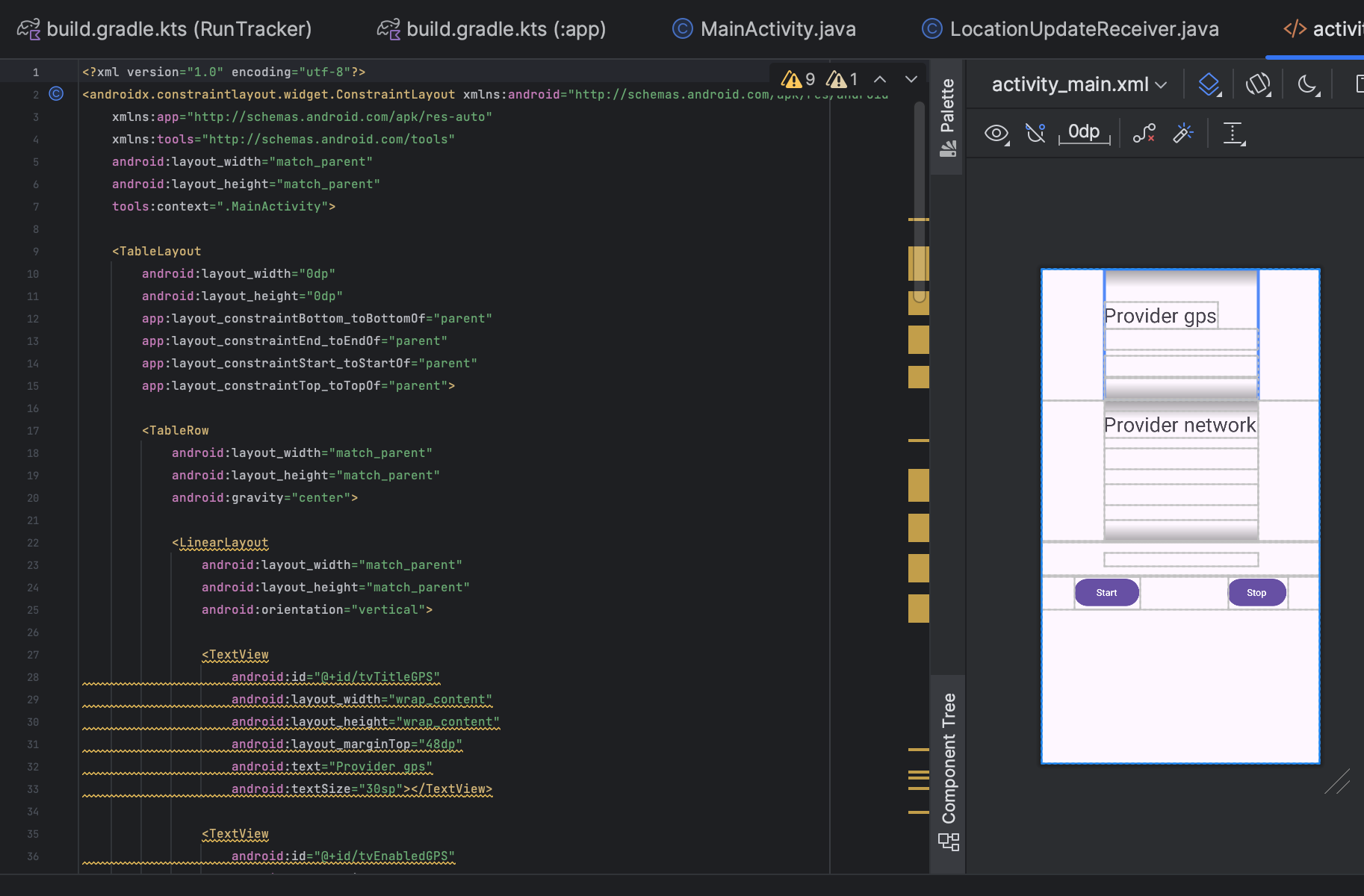


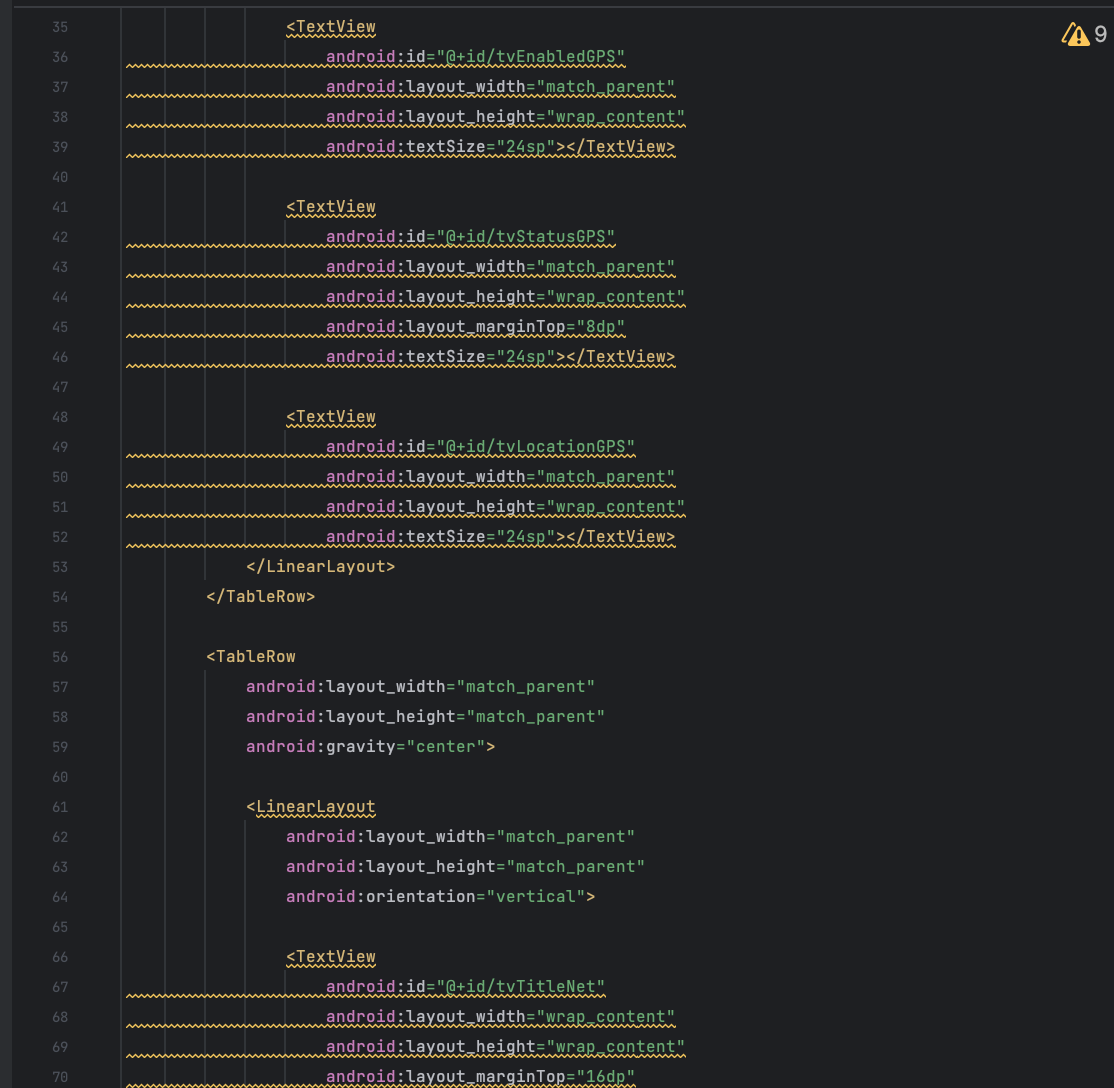


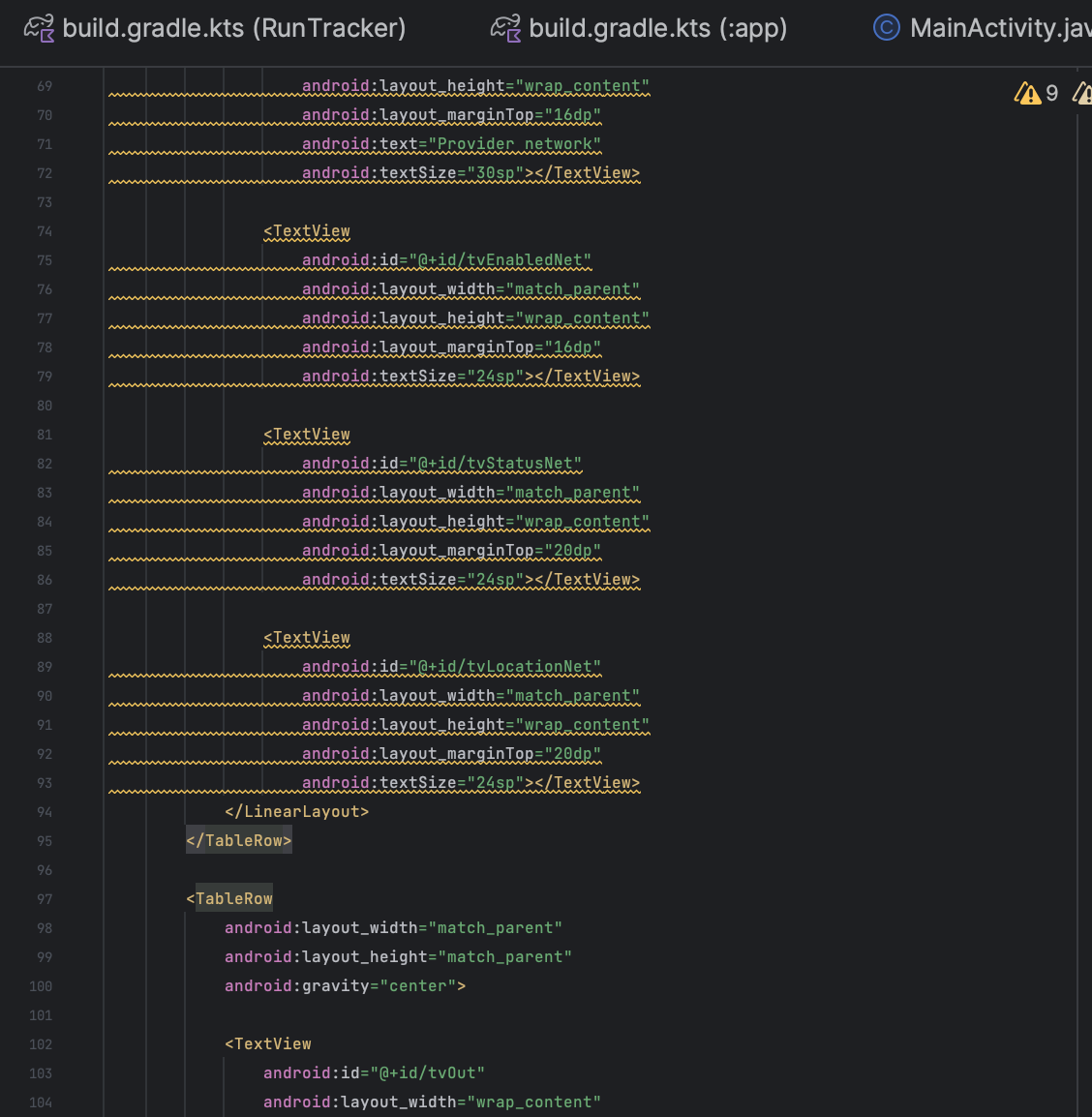


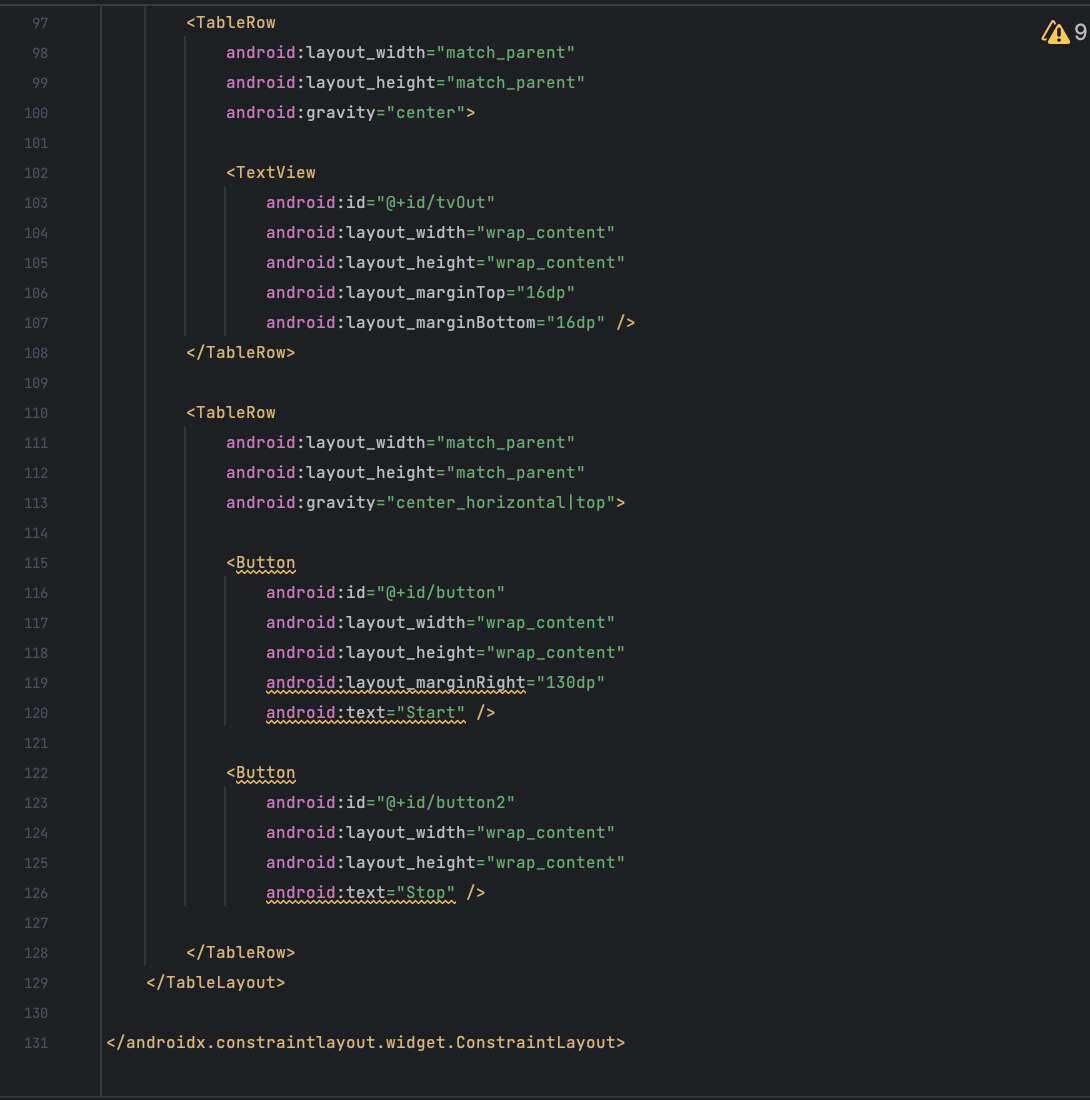


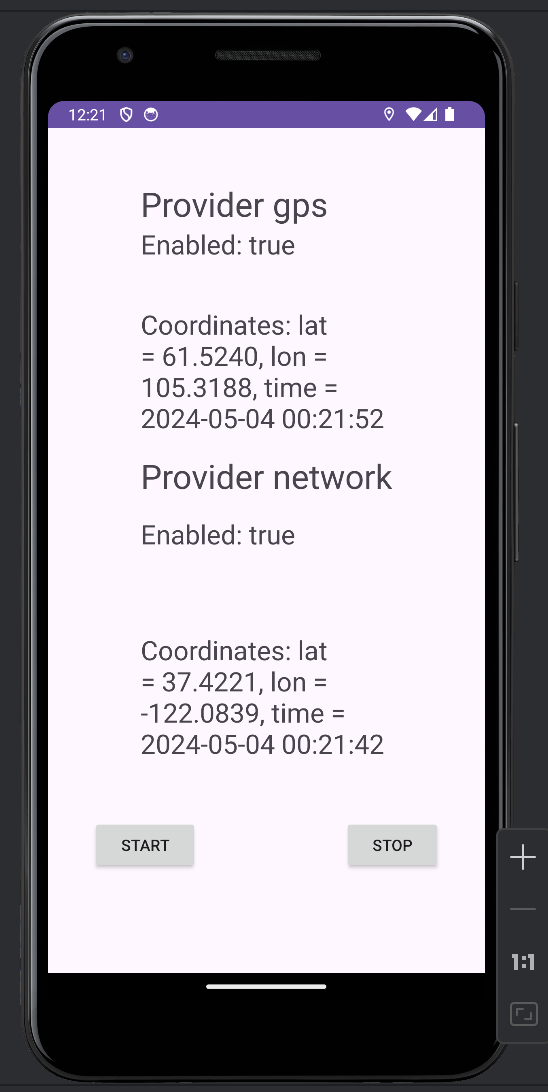
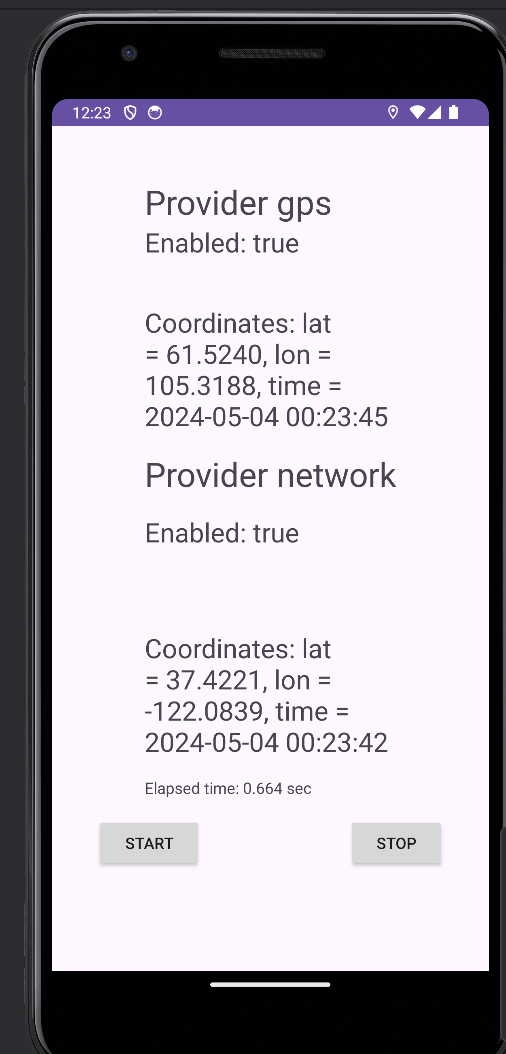










**Задание 4**

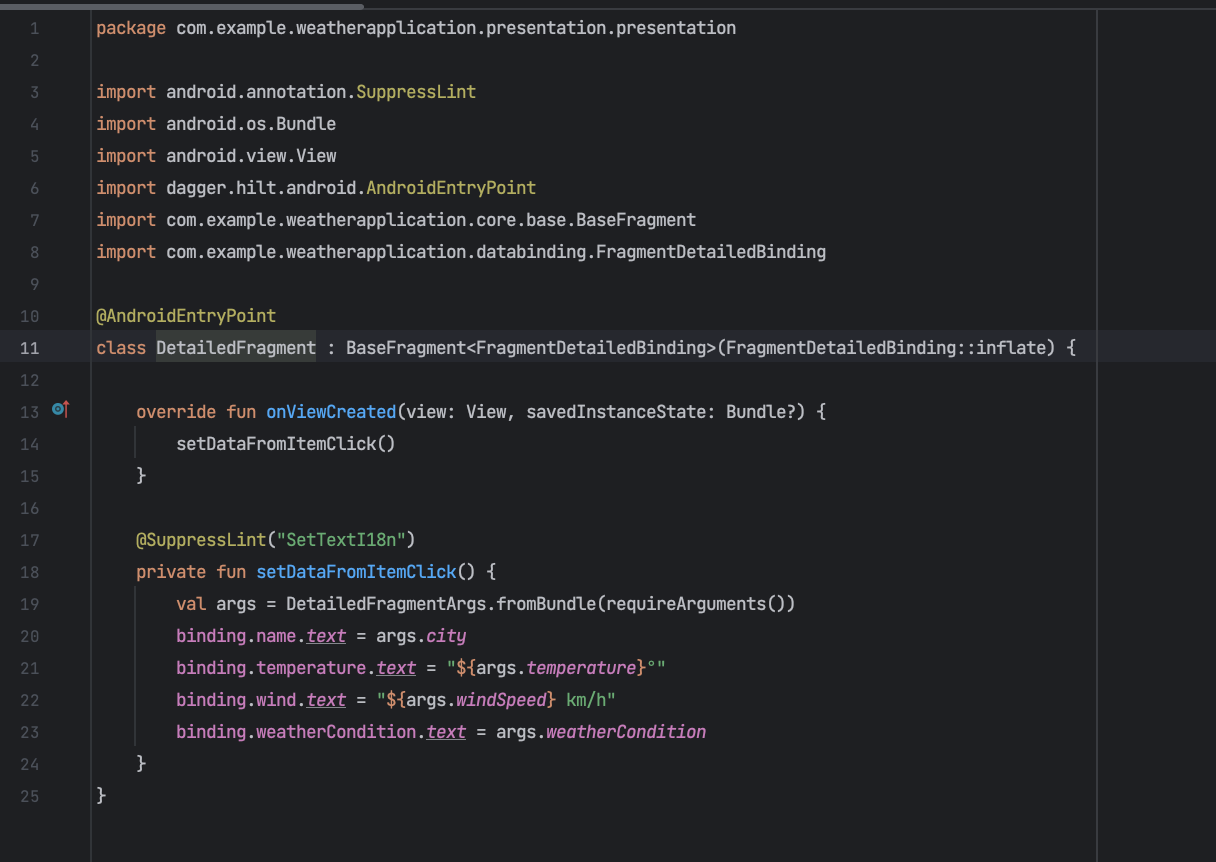
Реализуйте погодное приложение на Java или Kotlin с получением ответа от публичного API и извлечением ответа JSON. Приложение должно иметь минимум 2 встроенных города (Областной город по месту рождения, любой город название которого начинается на первую букву Вашей фамилии на русском или английском языке). Пользователь может добавить свой город. Приложение должно

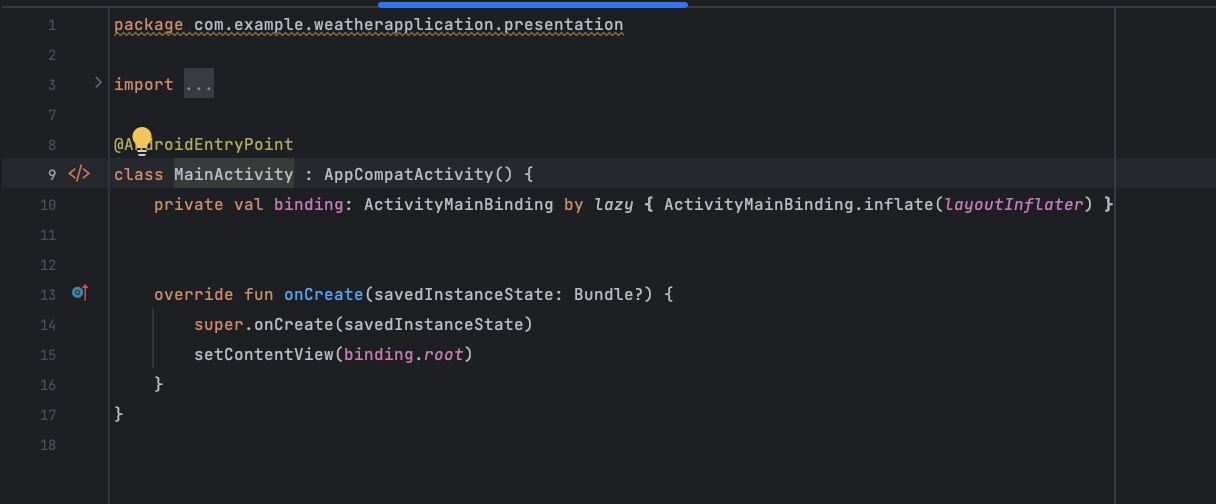
a) выводить города с указанием температуры,

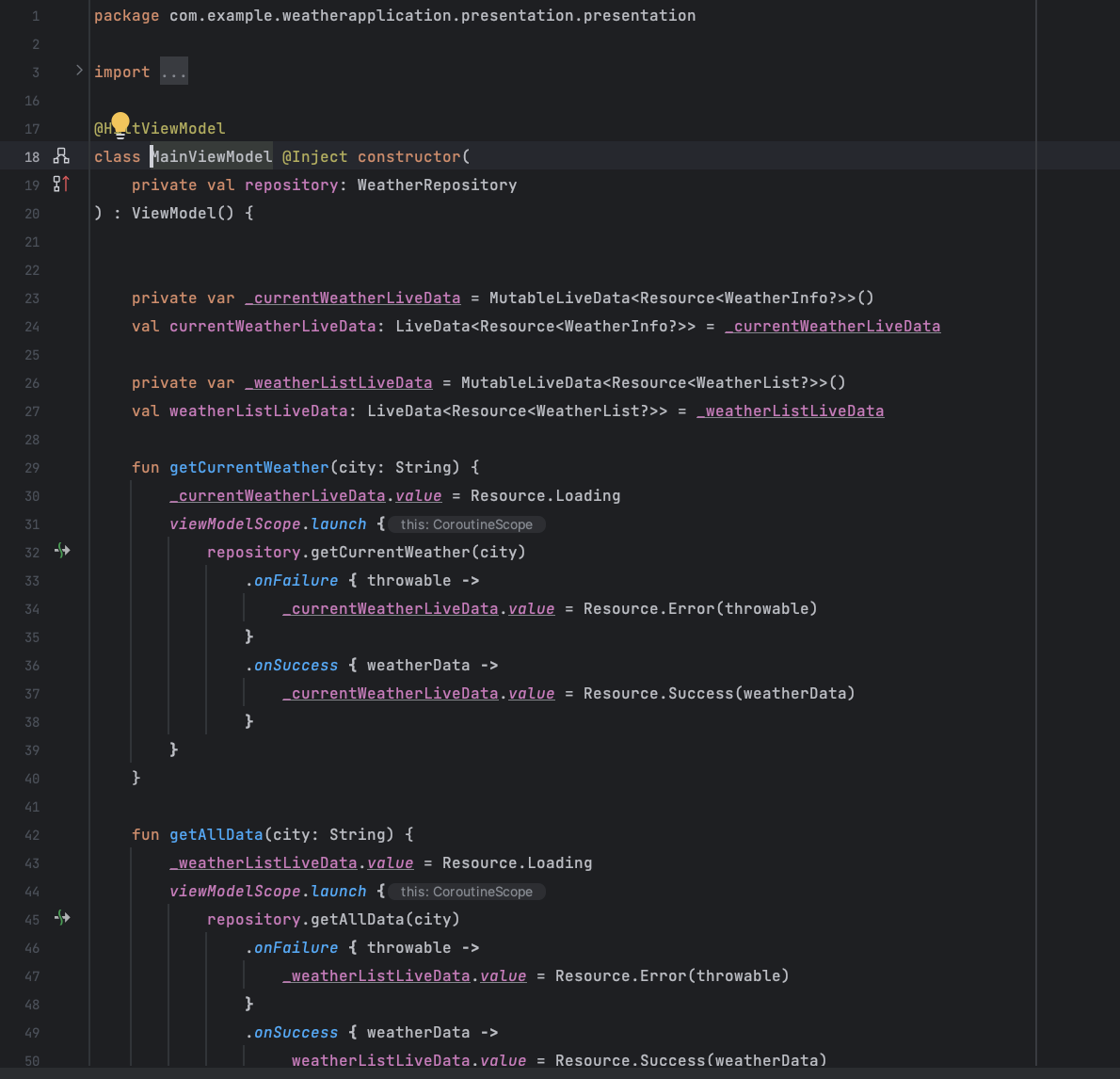
b) должно уметь показывать более подробную информацию по городу

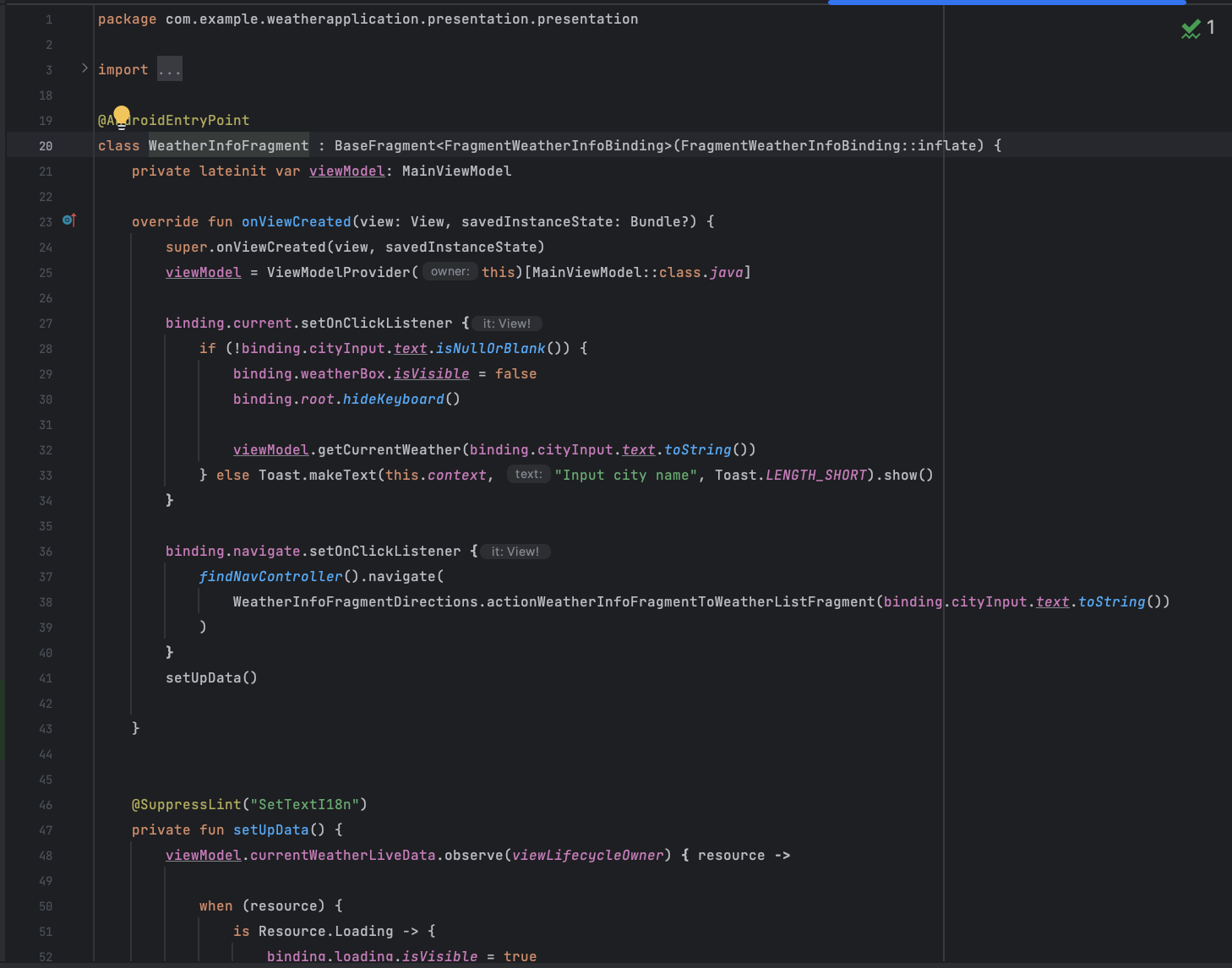
c) уметь показывать прогноз погоды (3 или 7 дней). 6

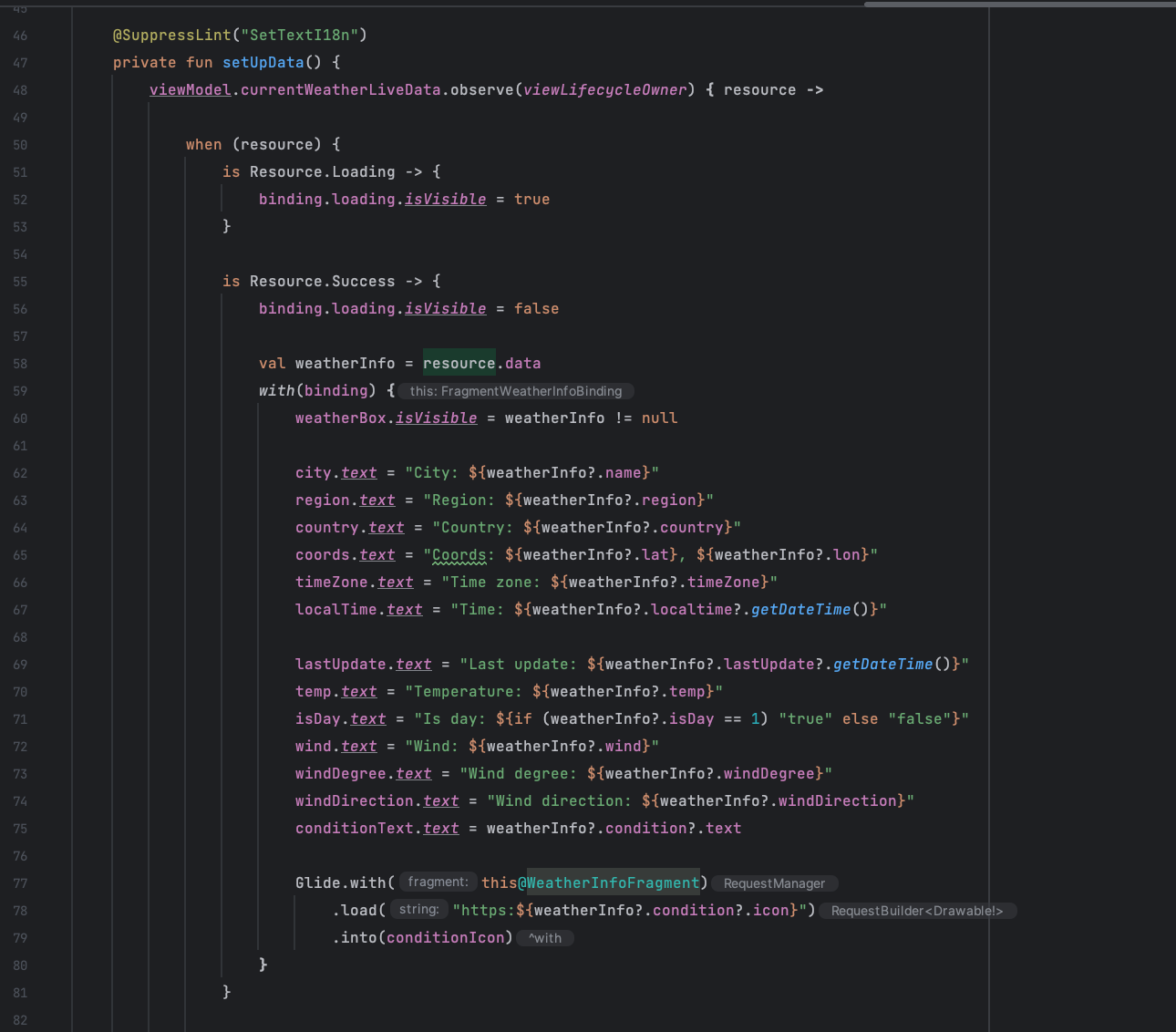
http://developer.yahoo.com/weather/, или на http://openweathermap.org/API, или любой другой на выбор.

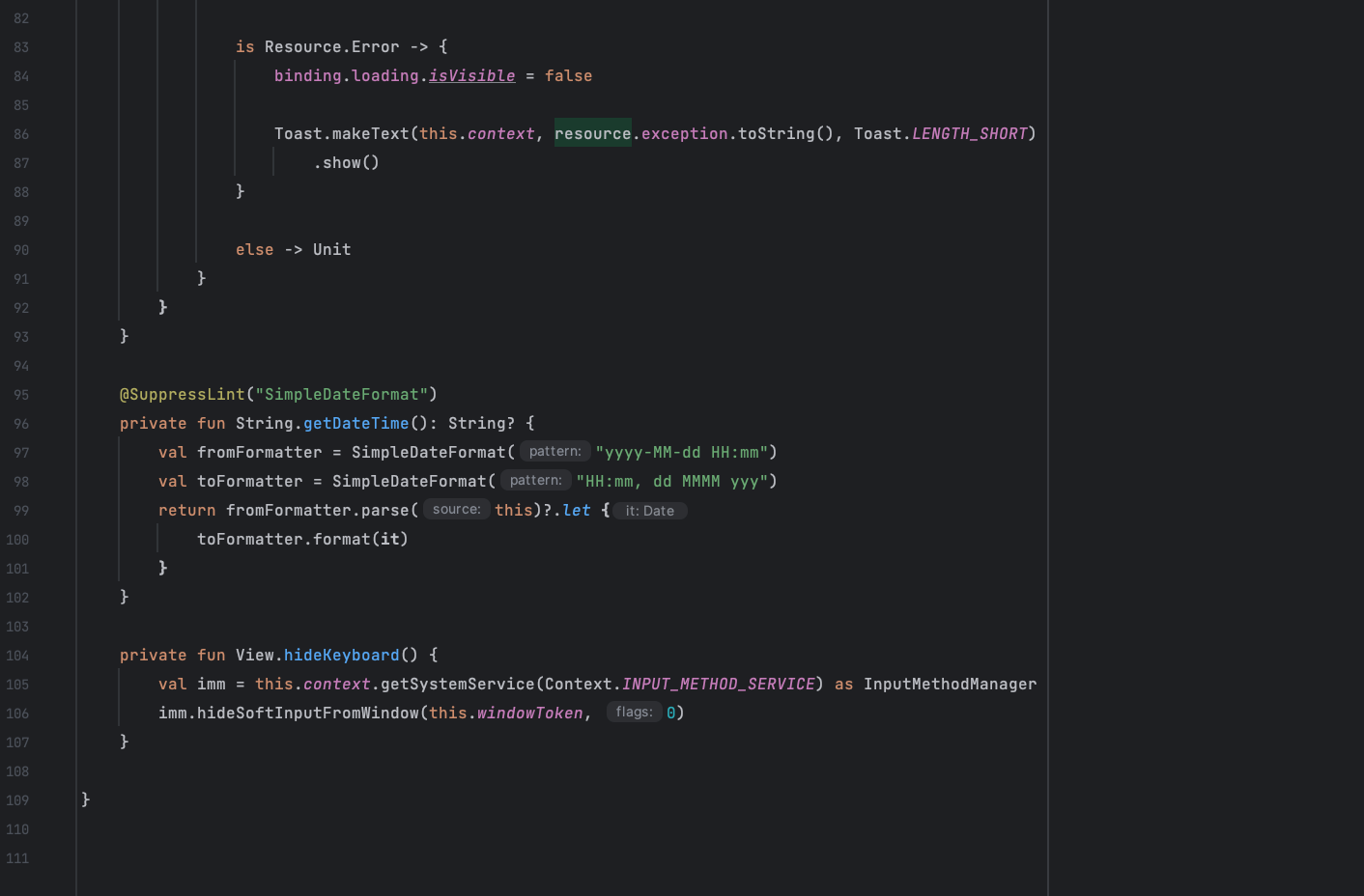
Код приложения:  


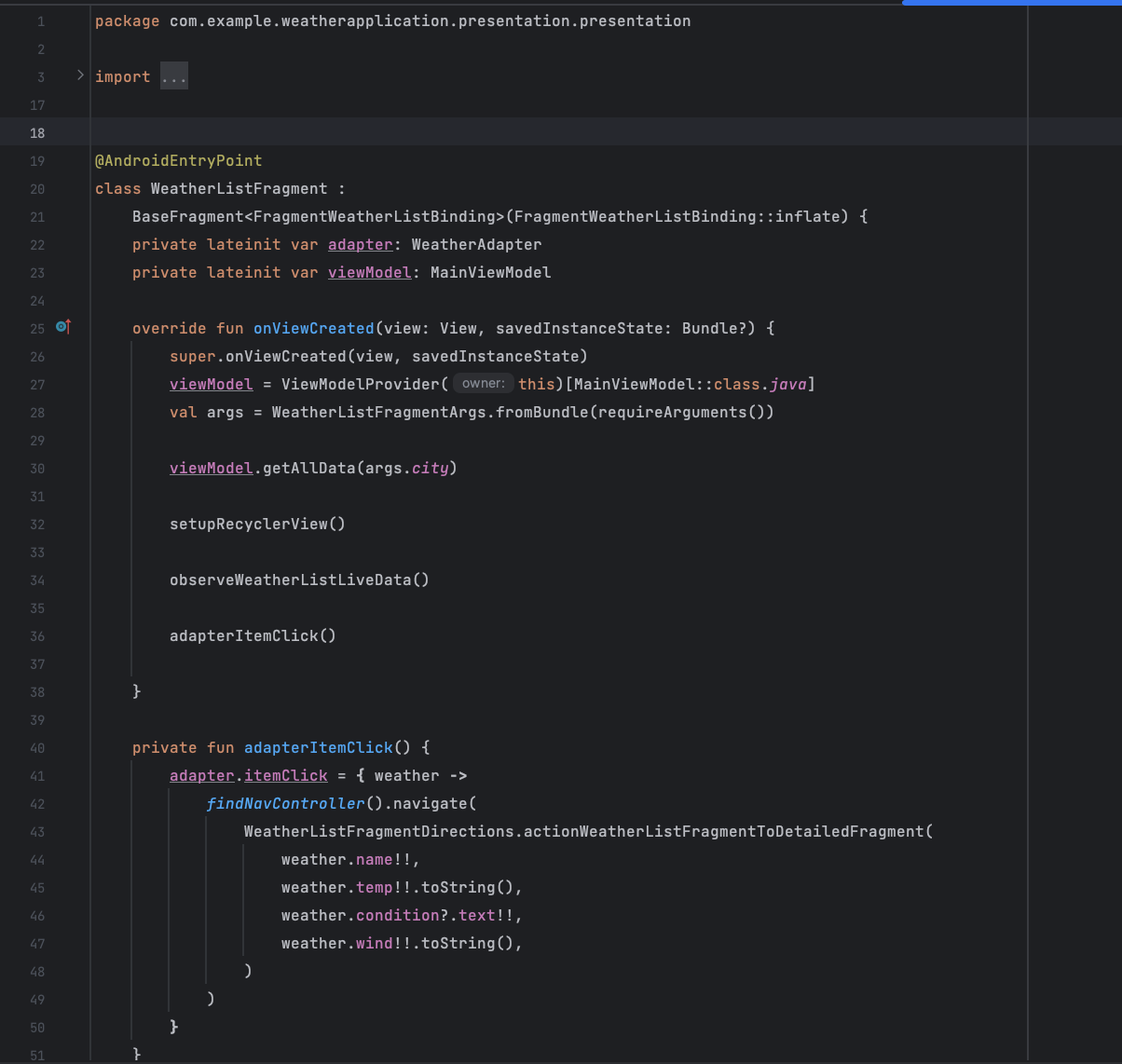




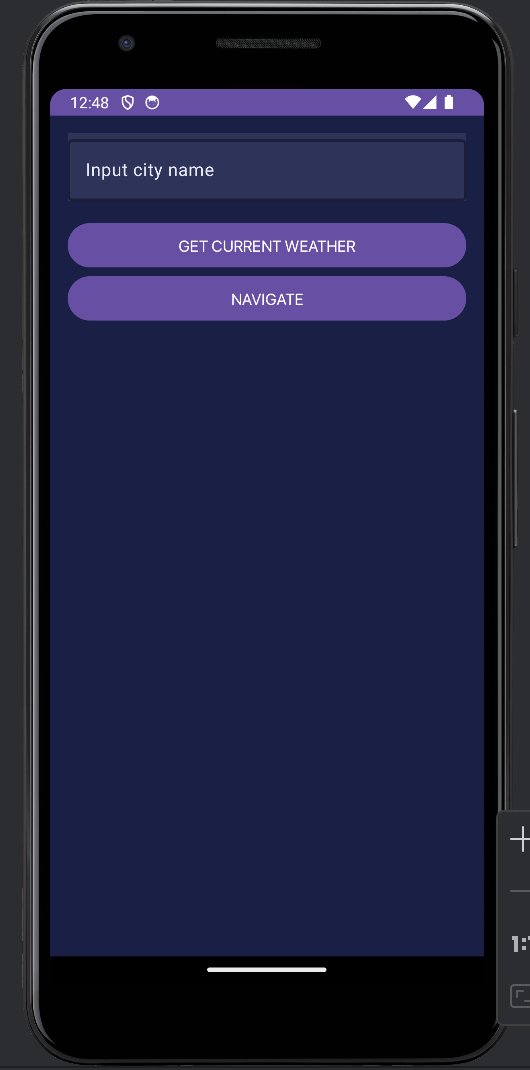
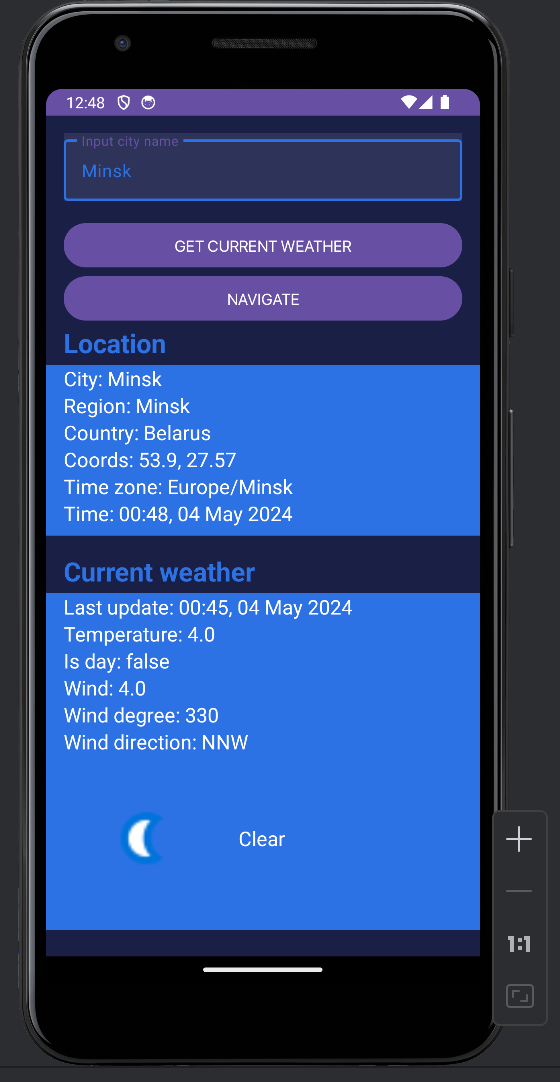
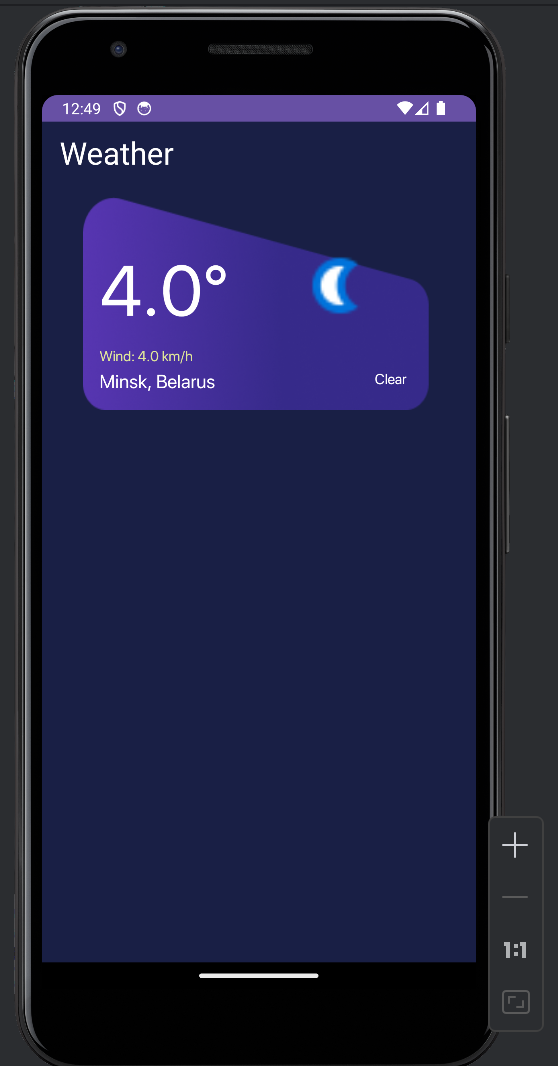










**    
  
Ответы на контрольные вопросы**

1. Какой интерфейс используется для получения уведомлений от

LocationManager, когда местоположение изменилось?

Ответ: LocationListener

2. Как расшифровывается аббревиатура NMEA? И для чего применяется?

Ответ: NMEA - National Marine Electronics Association. NMEA – это стандартный формат данных, поддерживаемый всеми производителями GPS. Его цель – предоставить пользователям оборудования возможность комбинировать аппаратные и программные средства.

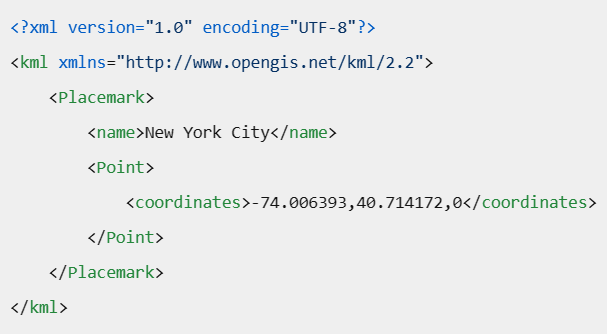
3. Как расшифровывается аббревиатура GNSS?

Ответ: Global Navigation Satellite System

4. Расшифруйте аббревиатуры GPX, KML. Для каких задач применяются файлы формата GPX и KML? Приведите примеры файлов формата GPX и KML.

Ответ: GPX – GPS Exchange Format. Этот формат предназначен для безопасного хранения и передачи данных для устройств GPS или мобильных навигационных приложений. KML - Keyhole Markup Language, это язык разметки, который был разработан для аннотирования карт.

Примеры файлов GPX и KML:



5. Когда рекомендуется применять для определения местоположения PendingAPI вместо LocationManager?

### Ответ: когда требуется, чтобы обновления информации поступало в несколько компонентов приложения. Поддержка асинхронных операций

6. Как называется класс данных, использующийся для представления географического местоположения?

Ответ: Location

7. Какими данными описывается местоположение?

Ответ: Широта (latitude), Долгота (longitude).

8. Какие строки необходимо добавить в AndroidManifest.xml, чтобы приложение для определения местоположения получило доступ к Интернет?

Ответ: <uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>

9. Какие концепции реализации многопоточности в Android можем использовать?

Ответ: 1. UI-поток (Main Thread):

Это основной поток приложения, который обрабатывает события пользовательского интерфейса.

Важно не блокировать UI-поток долгими операциями, чтобы приложение не зависало.

AsyncTask:

Устаревший, но все еще используемый класс для выполнения асинхронных задач в фоновом режиме.

Позволяет выполнять длительные операции в фоне и обновлять UI после завершения.

Handler и Looper:

Позволяют отправлять сообщения между потоками.

Handler работает с Looper, который обрабатывает сообщения в цикле.

Thread и Runnable:

Thread – это класс для создания новых потоков.

Runnable – это интерфейс, который представляет задачу, которую можно выполнить в отдельном потоке.

Executor и ThreadPoolExecutor:

Позволяют управлять пулом потоков для выполнения задач.

Executor – это интерфейс, а ThreadPoolExecutor – его реализация.

Coroutines:

Новый подход к асинхронному программированию, представленный в Kotlin.

Позволяют писать асинхронный код в структуре, похожей на синхронный.

LiveData и ViewModel:

LiveData – это компонент архитектуры Android, который позволяет обновлять UI в реакции на изменения данных.

ViewModel – это компонент, который хранит данные между конфигурациями активности/фрагмента.

RxJava:

Библиотека для реактивного программирования.

Позволяет работать с асинхронными потоками данных.

10.Для каких основных компонент может применять асинхронную обработку и многопоточность и с помощью каких подходов ( привести примеры классов, библиотек и т. д.)?

Ответ:

Загрузка данных из сети:

Retrofit: Библиотека для работы с REST API. Позволяет выполнять сетевые запросы асинхронно.

OkHttp: HTTP-клиент, который также поддерживает асинхронные запросы.

Работа с базой данных:

Room: Библиотека для работы с SQLite. Позволяет выполнять запросы к базе данных асинхронно.

LiveData: Компонент архитектуры Android, который обновляет UI в реакции на изменения данных асинхронно.

Обработка изображений:

Glide: Библиотека для загрузки и кэширования изображений. Выполняет операции асинхронно.

Параллельное выполнение задач:

Executor и ThreadPoolExecutor: Позволяют управлять пулом потоков для выполнения задач.

AsyncTask: Устаревший, но все еще используемый класс для выполнения асинхронных задач в фоновом режиме.

Работа с файлами и хранилищем:

File I/O: Чтение и запись файлов асинхронно.

Storage Access Framework: Позволяет выбирать файлы из хранилища асинхронно.

Обработка событий и обновление UI:

Handler и Looper: Позволяют отправлять сообщения между потоками и обновлять UI асинхронно.

ViewModel и LiveData: Позволяют хранить данные между конфигурациями активности/фрагмента и обновлять UI асинхронно.

Примеры классов и библиотек: AsyncTask, Executor, ThreadPoolExecutor, LiveData и ViewModel