**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра технологий программирования

Жуковский Павел Сергеевич

Программирование Мобильных и Встраиваемых Систем

Отчет по лабораторной работе №10

## «Разработка мобильных приложений с функцией геолокации и сетевым взаимодействием»

студента 3 курса 12 группы

**Преподаватель**

**Давидовская Мария Ивановна**

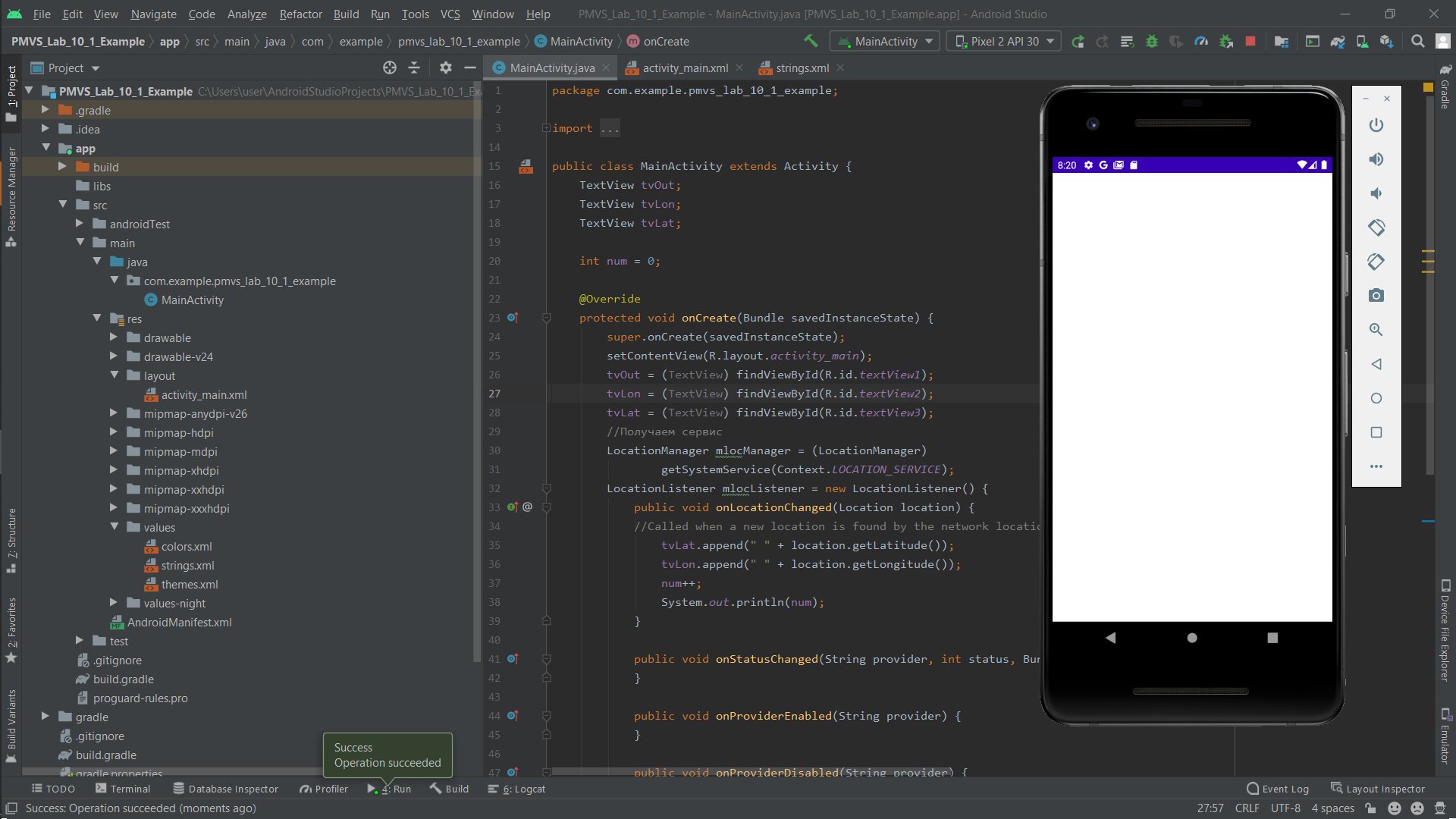
**Минск 2020**

**Ход выполнения**

**10.1. Разработка приложения, получающего координаты устройства и отслеживающего их изменение**

**Пример.**

Перенёс код для примера и изучил его:



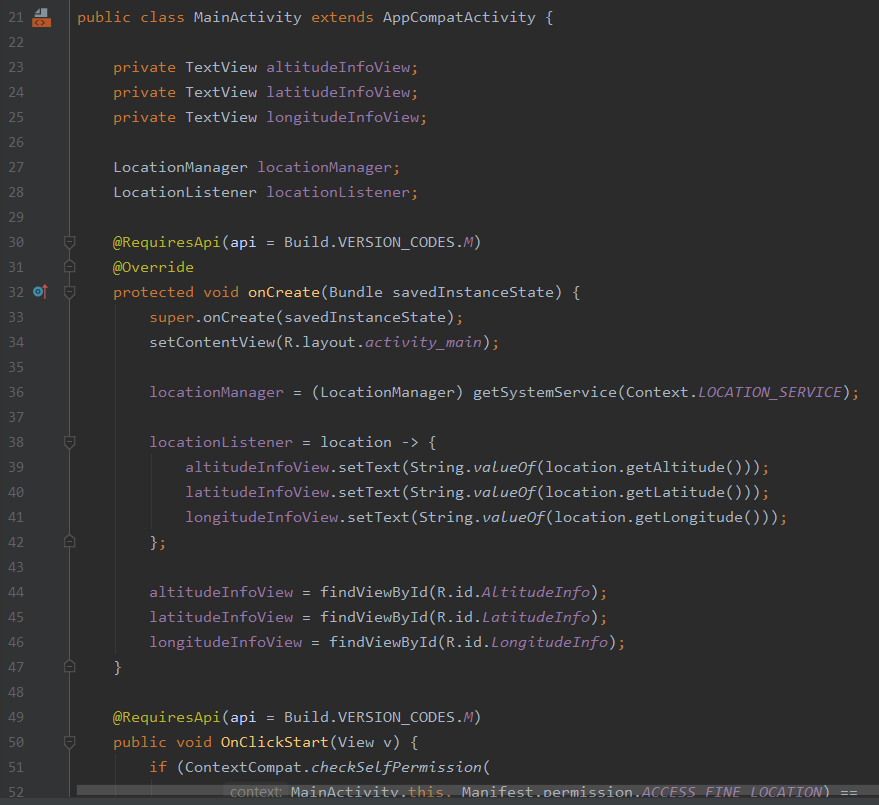
**10.2. Взаимодействие с сервером**

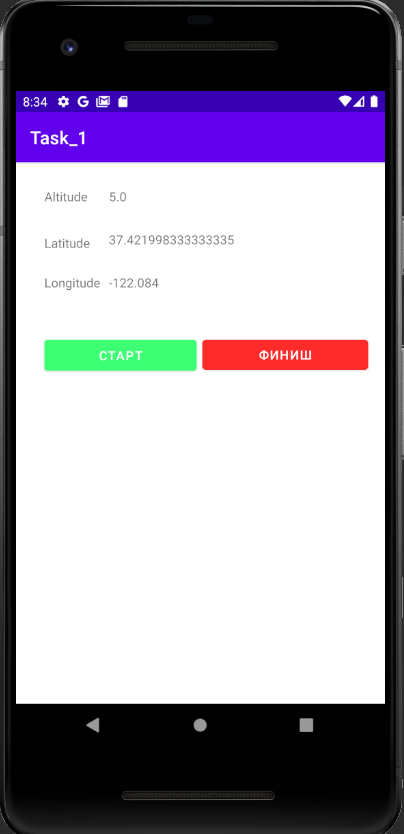
Изучил документацию

**Задания для самостоятельной работы**

**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 10.1 текущей лабораторной работы. Пример интерфейса представлен на рисунке на странице 3 раздела 10.1. 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.*



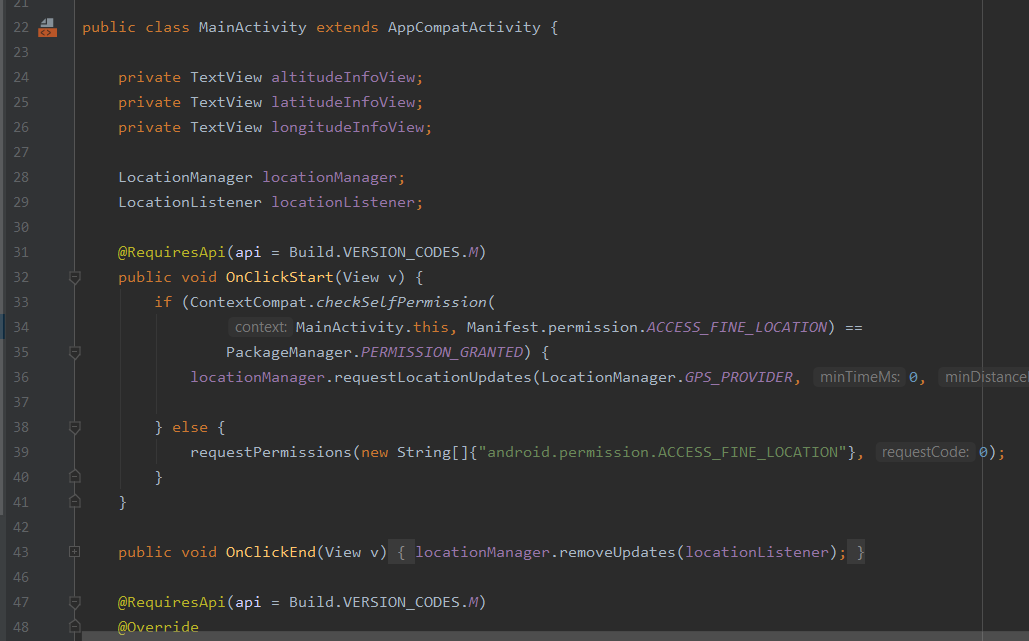


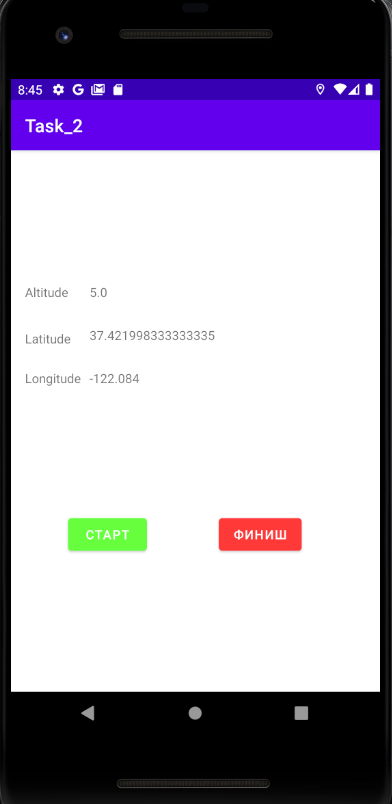
**Задание 2.**

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

*а) GPX-файл*

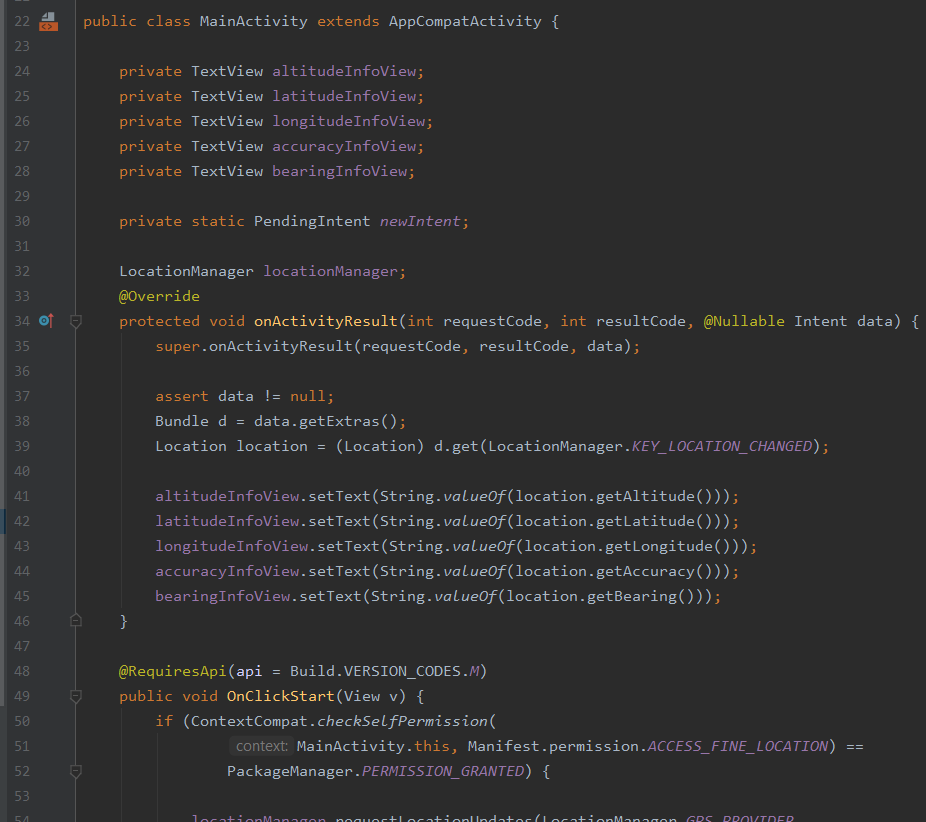
*б) KML-файл*

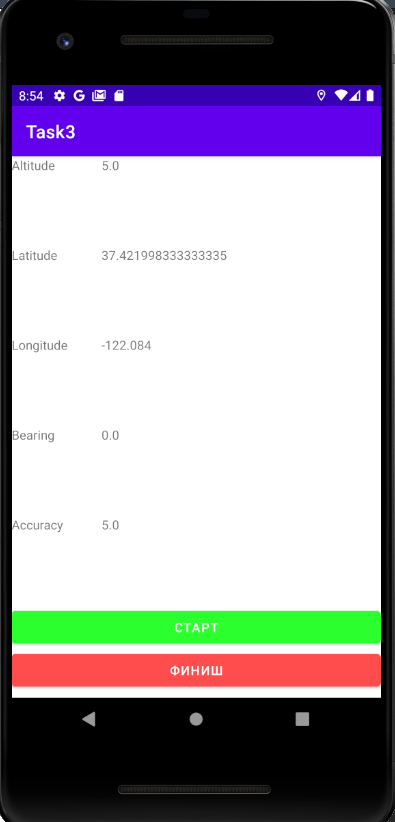




**Задание 3.**

*Реализовать приложение RunTracker из задания 1, изменив разметку на TableLayout и LinerLayout и применив PendingIntent API. 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.*





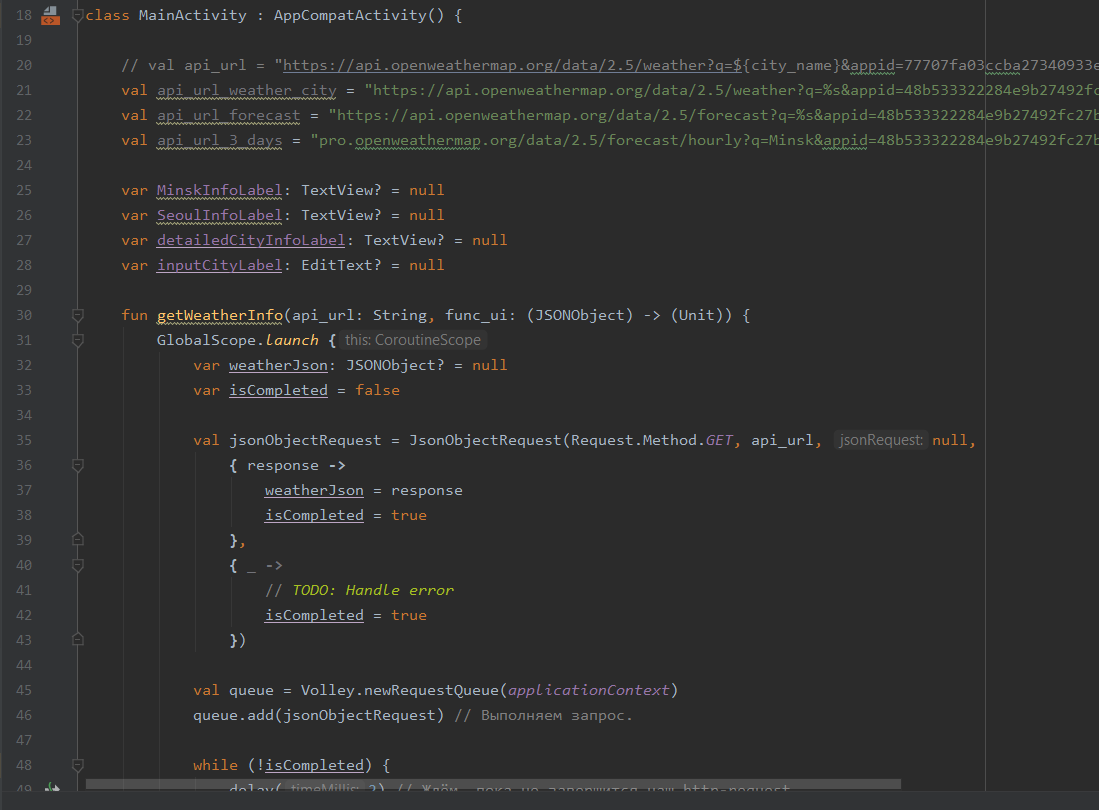
**Задание 4.**

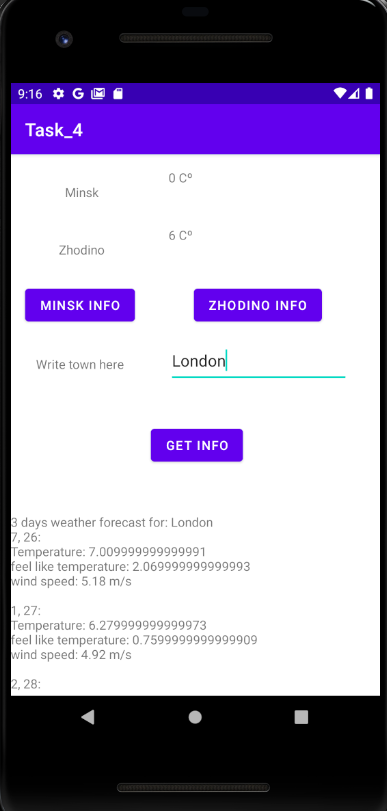
*Реализуйте погодное приложение на Java или Kotlin с использованием AsyncTask, получением ответа от API и извлечением ответа JSON. Приложение должно иметь минимум 2 встроенных города (Областной город по месту рождения, любой город название которого начинается на первую букву Вашей фамилии на русском 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). http://developer.yahoo.com/weather/, или на http://openweathermap.org/API, или любой другой на выбор.*





**Контрольные вопросы**

*1. Какой интерфейс используется для получения уведомлений от LocationManager, когда местоположение изменилось?*

LocationListener.

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

NMEA (National Marine Electronics Association) — это формат передачи

сообщений между корабельными приборами. Он включает в себя систему

сообщений для обмена информацией между навигационными GPS-

приемниками и потребителями навигационной информации.

*3. Как расшифровывается аббревиатура GNSS?*

Спу́тниковая систе́ма навига́ции (англ. Global Navigation Satellite System,

GNSS, ГНСС) — система, предназначенная для определения местоположения

(географических координат) наземных, водных и воздушных объектов, а также

низкоорбитальных космических аппаратов. Спутниковые системы навигации

также позволяют получить скорости и направления движения приёмника

сигнала.

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

GPX (GPS eXchange Format)

Формат позволяет хранить информацию об ориентирах, маршрутах и треках.

Для каждой точки хранятся её долгота, широта и высота над уровнем моря.

Пример:

&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;

&lt;gpx

xmlns=&quot;http://www.topografix.com/GPX/1/1&quot;

version=&quot;1.1&quot;

creator=&quot;Wikipedia&quot;

xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;

xsi:schemaLocation=&quot;http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd&quot;&gt;

&lt;time&gt;2011-09-22T18:56:51Z&lt;/time&gt;

&lt;metadata&gt;

&lt;name&gt;Name&lt;/name&gt;

&lt;desc&gt;Description&lt;/desc&gt;

&lt;author&gt;

&lt;name&gt;Autor&lt;/name&gt;

&lt;/author&gt;

&lt;/metadata&gt;

&lt;trk&gt;

&lt;name&gt;exercise&lt;/name&gt;

&lt;trkseg&gt;

&lt;trkpt lat=&quot;59.934721667&quot; lon=&quot;30.310183333&quot;&gt;

&lt;time&gt;2011-09-22T18:56:51Z&lt;/time&gt;

&lt;ele&gt;0&lt;/ele&gt;

&lt;fix&gt;none&lt;/fix&gt;

&lt;sat&gt;2&lt;/sat&gt;

&lt;/trkpt&gt;

&lt;trkpt lat=&quot;59.934731667&quot; lon=&quot;30.310083333&quot;&gt;

&lt;time&gt;2011-09-22T18:56:56Z&lt;/time&gt;

&lt;ele&gt;172&lt;/ele&gt;

&lt;fix&gt;2d&lt;/fix&gt;

&lt;sat&gt;5&lt;/sat&gt;

&lt;/trkpt&gt;

&lt;trkpt lat=&quot;59.934646667&quot; lon=&quot;30.309901667&quot;&gt;

&lt;time&gt;2011-09-22T18:57:01Z&lt;/time&gt;

&lt;ele&gt;172&lt;/ele&gt;

&lt;fix&gt;3d&lt;/fix&gt;

&lt;sat&gt;9&lt;/sat&gt;

&lt;/trkpt&gt;

&lt;/trkseg&gt;

&lt;/trk&gt;

&lt;/gpx&gt;

KML (Keyhole Markup Language).

В сервис &quot;Google Планета Земля&quot; можно загружать геоданные в формате KML,

чтобы изучать их или показывать другим. Формат KML (Keyhole Markup

Language) разработан специально для хранения в этом сервисе географических

данных и связанного с ними контента.

Пример:

&lt;?xml version=»1.0″ standalone=»yes»?&gt;

&lt;kml xmlns=»http://earth.google.com/kml/2.1″&gt;

&lt;Placemark&gt;

&lt;name&gt;Дворец спорта&lt;/name&gt;

&lt;description&gt;Дворец спорта «Нагорный» просп. Гагарина, 29&lt;/description&gt;

&lt;Point&gt;

&lt;coordinates&gt;43.979459,56.291908,0 &lt;/coordinates&gt;

&lt;/Point&gt;

&lt;/Placemark&gt;

&lt;/kml&gt;

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

Запрашивая информацию местоположения с использованием

PendingIntent, фактически приказываем LocationManager отправлять некую

разновидность Intent в будущем. Таким образом, компоненты приложения (и

даже весь процесс) могут прекратить существование, а LocationManager будет

доставлять интенты, пока не будет отправлен приказ ему остановиться,

запустив новые компоненты, которые отреагируют на интенты нужным

образом. Например, такая схема позволит предотвратить поглощение

приложением лишних ресурсов, в то время как оно активно отслеживает

местоположение устройства.

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

Location.

*7. Какими данными описывается местоположение?*

Ширина и долгота.

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

&lt;uses-permission android:name=&quot;android.permission.INTERNET&quot;/&gt;

*9. Для каких задач применяется класс AsyncTask?*

Его цель – это выполнение тяжелых задач и передача в UI-поток

результатов работы. Но при этом нам не надо задумываться о создании Handler

и нового потока.

*10.Для чего предназначены методы AsyncTask, такие как onPreExecute(), doInBackground(Params), onPostExecute(Result)?*

doInBackground – будет выполнен в новом потоке, здесь решаем все свои

тяжелые задачи. Т.к. поток не основной - не имеет доступа к UI.

onPreExecute – выполняется перед doInBackground, имеет доступ к UI.

onPostExecute – выполняется после doInBackground (не срабатывает в случае,

если AsyncTask был отменен - об этом в следующих уроках), имеет доступ к UI.