

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Петрозаводский государственный университет»  
Институт математики и информационных технологий  
Кафедра информатики и математического обеспечения

Потес Артём Сергеевич

Отчет по учебному курсу «Верификация Программного Обеспечения»

ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ ANDROID  
В РАМКАХ ВЫПОЛНЕНИЯ ПРОЕКТА НИР «РАЗРАБОТКА ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ  
ANDROID ДЛЯ МОБИЛЬНОГО СЕРВИСА ПОДДЕРЖКИ СОРЕВНОВАНИЙ ФЕДЕРАЦИИ  
СПОРТИВНОГО ОРИЕНТИРОВАНИЯ РК»

Направление 09.04.02 — Информационные системы и технологии

Программа «Управление данными»

Преподаватель:

к.ф.-м.н., доцент К. А. Кулаков

Петрозаводск

2020

# Содержание

1. Объект тестирования .....	3
1.1 Описание приложения .....	3
1.2 Архитектура приложения.....	4
1.3 Функции мобильного приложения.....	7
2 Стратегия тестирования.....	7
2.1 Структура объекта тестирования.....	7
2.2 Стратегия блочного тестирования .....	9
2.3 Стратегия интеграционного тестирования .....	9
2.4 Стратегия аттестационного тестирования .....	9
2.5 Стратегия нагрузочного тестирования .....	9
2.6 Критерии прохождения тестирования.....	10
2.6 Условия возобновления и приостановки выполнения тестов .....	10
3. Детальный план тестов.....	11
3.0 Фикстуры .....	11
3.1 Блочное тестирование.....	13
3.1.1 Метод <code>getLine</code> .....	13
3.1.2 Метод <code>readTrack</code> .....	15
3.1.3 Метод <code>getMainParameters</code> .....	17
3.1.4 Метод <code>createTempforKilometer</code> .....	21
3.1.5 Метод <code>saveTrack</code> .....	24
3.1.6 Метод <code>sendTrack</code> .....	27
3.2 Интеграционное тестирование.....	29
3.2.1 Интеграция метода <code>saveTrack</code> в метод <code>readTrack</code> .....	29
3.2.2 Интеграция метода <code>sendTrack</code> и БД .....	32
3.3 Аттестационное тестирование.....	34
3.4 Нагрузочное тестирование .....	39
3.5 Покрытие кода тестами .....	43
3.6 Пример реализации тестов.....	43
4. Журнал тестирования.....	44
5. Журнал найденных ошибок.....	46
6. Результаты .....	47

# 1. Объект тестирования

## 1.1 Описание приложения

Спортивное ориентирование – вид спорта, в котором спортсмены, используя спортивную карту и компас, проходят неизвестную им трассу через контрольные пункты, расположенные на местности. Контрольный пункт (КП) – это точка, в которой должны побывать участники соревнований. КП располагаются на ориентирах, обозначенных на спортивной карте. На карте КП обозначаются в виде красных кругов.

Так как соревнования по спортивному ориентированию проводятся не на закрытых площадках, а на открытой местности, то главной задачей является корректное получение GPS-координат спортсмена во время преодоления дистанции. На основе полученных GPS-координат высчитываются основные беговые значения, такие как: общий километраж, средняя скорость, средний темп, максимальный и минимальный темп и т.д. Кроме вычисления беговых показателей, важной функцией является наложение трека на спортивную карту, с последующим разбиением трека на темповые зоны.

Неотъемлемой частью жизненного цикла разработки программного обеспечения является тестирование, которое позволяет определить проблемные участки кода и в следствии этого повысить качество разрабатываемого продукта.

Задачами тестирования Android приложения для спортивного ориентирования являются:

- Повышение качества как отдельных модулей приложения, так и всего приложения в целом путем выполнения тестирования различных видов;
- Выявление ошибок в системе с целью их дальнейшего исправления.

Основные определения системы:

- Модуль - программная реализация набора функций (функции), предоставляющих ту или иную возможность;
- Подсистема - набор модулей, взаимодействующих между собой;
- Система - самостоятельное приложение, состоящее из подсистем;
- Сервис (Service) – компонента приложения, которая может выполнять длительные операции в фоновом режиме и не содержит пользовательского интерфейса;
- Приёмник широковещательных сообщений (BroadcastReceiver) – компонент для получения внешних событий и реакции на них;
- Служба переднего плана – это служба, о которой пользователь активно осведомлён, и которая не является кандидатом для удаления системой в случае нехватки памяти;
- Фрагмент (Fragment) – отдельный модуль приложения, который может быть интегрирован в различные активности;
- Активность (Activity) – отдельный экран в Android;
- Класс – это шаблон для создания объекта, определяющий структуру объекта и его методы, образующие функциональный интерфейс.

Объектом тестирования является приложение на платформе Android, состоящее из классов, сервисов и активностей. Классы представляют собой набор модулей, работающих отдельно и выполняющих свои функции (отображение результатов тренировочной деятельности; наложение трека на спортивную карту; получение, хранение и отображение спортивных карт; передача данных на

сервер). Сервис получает данные о местоположении и обрабатывает их. Таким образом, будет происходить тестирование как всей системы в целом, так и каждого отдельного модуля в частности.

## 1.2 Архитектура приложения

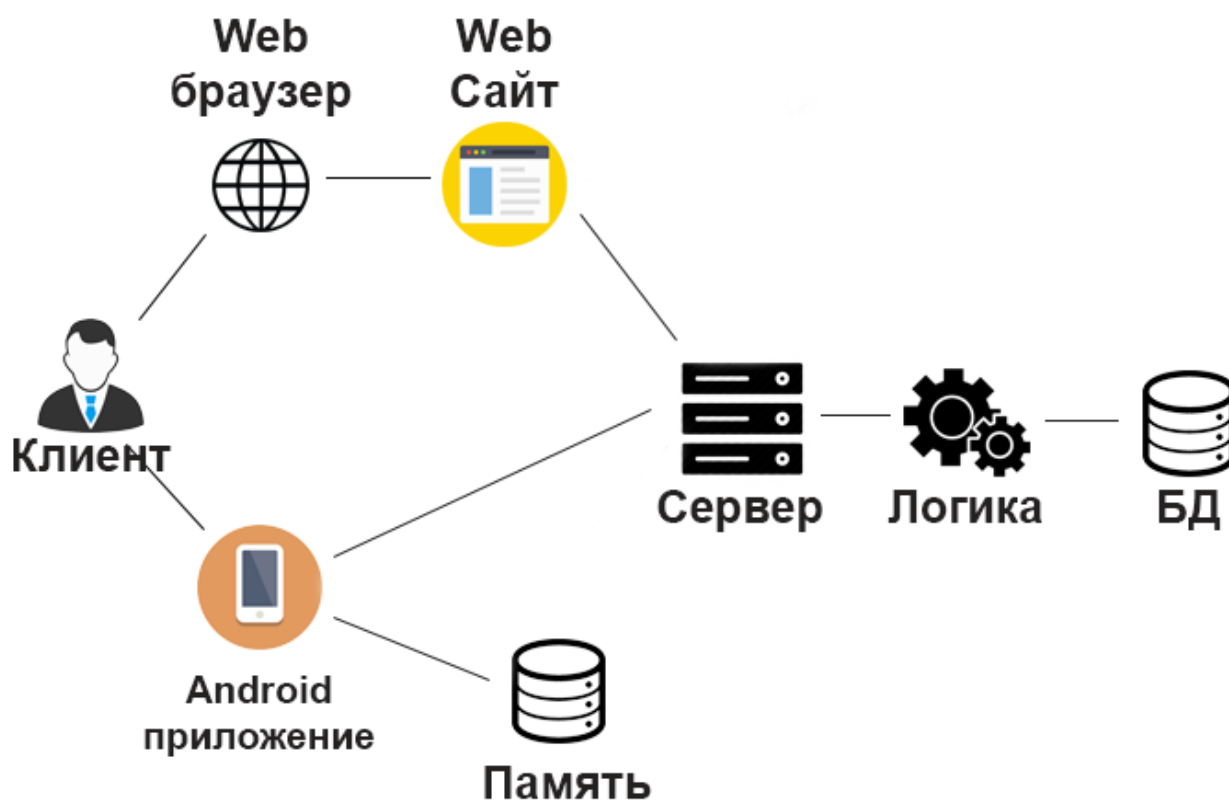


Рисунок 1: Высокоуровневая модель архитектуры

Высокоуровневая модель архитектуры (рис. 1) показывает взаимодействие всех элементов системы между собой. Тестирование будет проводиться только для части функций Android-приложения, так как система объёмная, и она не может быть в полной мере протестирована в рамках прохождения данного направления. Основными модулями являются:

- Сервер – выполняет все вычисления и работу с базой данных
  - Логика – необходимый набор функций для работы
  - БД – база данных для хранения информации о пользователях, тренировках, картах.
- Android-приложение – приложение для сбора gps-координат пользователя во время тренировки, просмотра и анализа тренировочной деятельности.
  - Память – внутренняя память устройства, в которой размещаются данные о тренировках (gps-координаты, основные параметры тренировки), загруженные спортивные карты с сервера.
- Web-сайт – сайт для просмотра треков пользователей.

Взаимодействие компонентов описывается следующим образом:

- Пользователь взаимодействует с web-сайтом при помощи форм и средств сайта.

- Web-сайт взаимодействует с сервером при помощи запросов к серверу и получения ответов от него.
- Пользователь взаимодействует с Android-приложением при помощи нажатия на экран телефона по определённым пунктам меню.
- Android-приложение взаимодействует с сервером при помощи запросов и получения ответов на них.

Система состоит из следующих модулей, представленных на рис.2:

- **ResultOfTrack**  
Данный модуль отвечает за отображение результатов забега (блок основных параметров забега, блок графиков и блок карты). В блоке основных параметров рассчитываются такие параметры как: средняя скорость, средний темп, максимальный/минимальный темп. Блок графиков включает в себя построение графика зависимости темпа от пройденного расстояния и вывод динамики среднего темпа по километрам. В блоке карты происходит получение Google map, её настройка и отрисовка трека.
- **ChooseMap**  
В данном модуле реализуется просмотр выбранной карты в отдельном окне.
- **Download\_map**  
В данном модуле запускается отдельный поток для загрузки карт с сервера приложения.
- **ListItem**  
В данном модуле реализованы методы для хранения основной информации о каждом треке (время забега, пройденное расстояние, дата) и получения дочерних треков для вывода их на экран в виде иерархического списка.
- **Listadapter**  
Адаптер списка треков, отвечающий за перестройку списка треков после его изменения (удаление элемента, сворачивание/разворачивание списка).
- **MainActivity**  
Главная Activity, в которой реализуются методы для записи и остановки трека. Также в данном модуле прописаны методы для взаимодействия с навигационным меню.
- **Fragment\_track\_list**  
Модуль отвечающий за получение списка треков, хранящихся в памяти устройства, его преобразования и передачи в ListAdapter для дальнейшего взаимодействия.
- **Sport\_map\_in\_new\_window**  
Модуль, отвечающий за наложение трека на спортивную карту, которая накладывается поверх географической, с последующим выводом в новом окне.
- **getMainParameters**  
Отдельный поток для расчёта данных, используемых в блоке основных параметров забега в модуле ResultOfTrack.
- **getDataForChartAndMap**  
Отдельный поток для расчёта данных, используемых для построения графиков в блоке графиков в модуле ResultOfTrack.
- **Send\_track**  
Отдельный поток для отправки треков на сервер.
- **ServicegetGPS**  
Сервис в котором реализована ключевая логика – получение и обработка GPS-координат и последующее их сохранение в памяти устройства.

При проведении тестирования будут рассмотрены только 2 класса: ServiceGetGPS и ResultOfTrack, так как система объёмная, и она не может быть в полной мере протестирована в рамках прохождения данного направления.

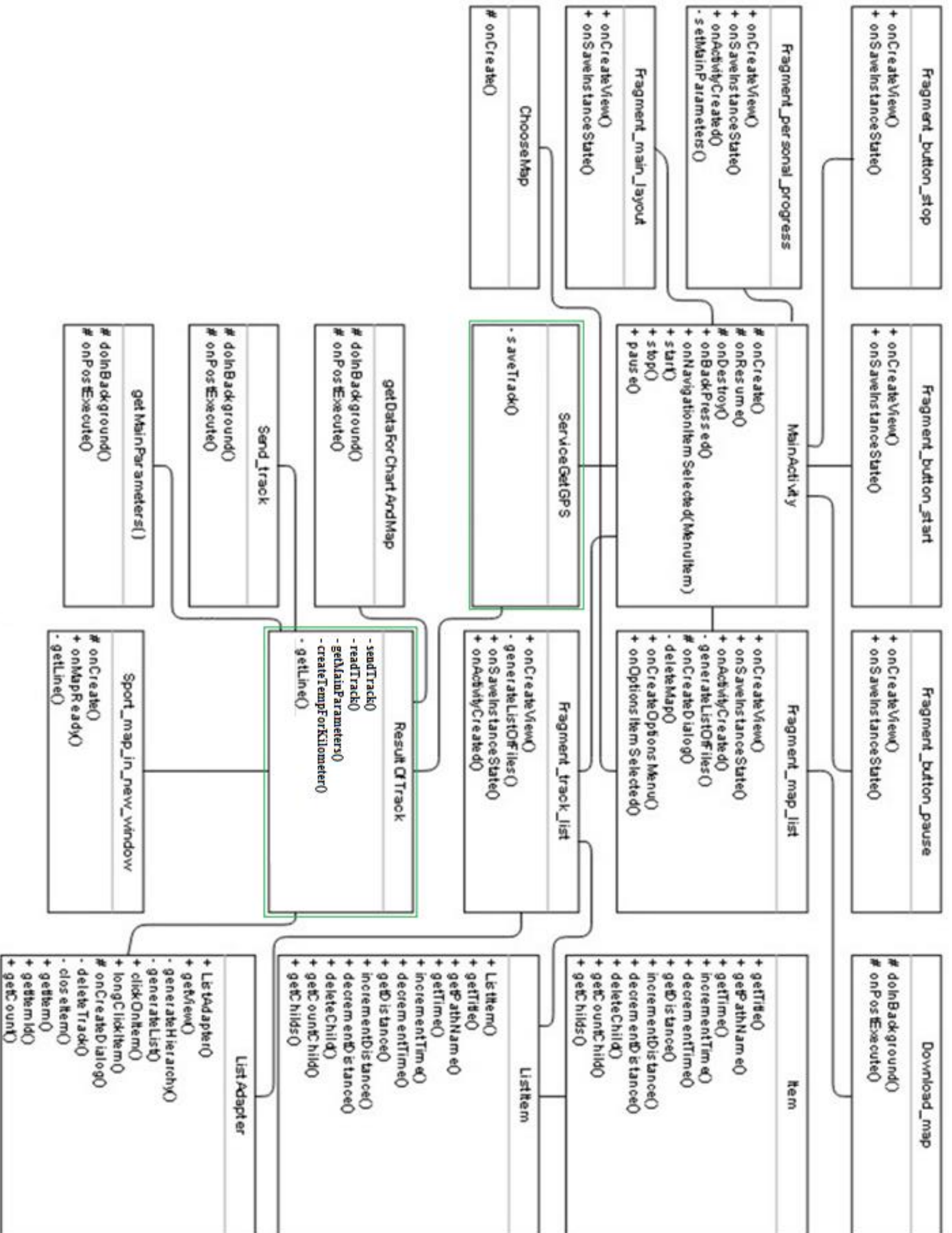


Рисунок 2: Диаграмма классов

## 1.3 Функции мобильного приложения

Основные функции Android-приложения:

1. Отображение результатов соревнований
  - 1.1. Извлечение и преобразование основных беговых показателей: время, расстояние, ср. скорость, ср. темп, макс. темп, мин. темп, набор высоты).
  - 1.2. Извлечение gps-координат и построение по ним графика зависимости темпа от километража.
  - 1.3. Извлечение данных gps-координат и построение графика отображения среднего темпа по километражу.
  - 1.4. Получение Google карты, масштабирование и центрирование её относительно центра проведения забега и отрисовка поверх неё трека с разбиением на темповые зоны.
  - 1.5. Передача результатов забега на сервер.
2. Наложение трека на спортивную карту.
  - 2.1. Загрузка спортивной карты из памяти устройства.
  - 2.2. Загрузка Google-карты.
  - 2.3. Наложение спортивной карты на географическую и отрисовка трека.
3. Работа с картами
  - 3.1. Отображение списка карт, сохраненных в памяти устройства.
  - 3.2. Загрузка новых карт с сервера.
  - 3.3. Просмотр спортивных карт в новом окне.
4. Работа со списком тренировок
  - 4.1. Отображение списка тренировок в виде иерархического списка, сгруппированного по месяцам и отсортированного внутри месяца по датам.
  - 4.2. Отображение для месяца – общие данные активности (общее время, проведенное за тренировками и суммарный километраж).
  - 4.3. Для тренировки – отображение продолжительности тренировки и пройденного километража.
  - 4.4. Перестроение списка после удаления из памяти устройства тренировки.
5. Работа с GPS-координатами
  - 5.1. Получение gps-координат через встроенные датчики в смартфоне.
  - 5.2. Обработка полученных gps-координат: расчёт основных показателей (время, километраж, набор высоты, темп, скорость).
  - 5.3. Сохранение полученных координат в памяти устройства с последующим открытием страницы просмотра результатов соревнований.

В рамках дисциплины будут частично протестированы функциональные возможности: 1 (извлечение и преобразование результатов тренировки), за исключением пункта 1.4, также будет протестирован пункт 5.3. Остальные функциональные возможности не будут протестированы, так как система объёмная, и она не может быть в полной мере протестирована в рамках прохождения данного направления.

## 2 Стратегия тестирования

### 2.1 Структура объекта тестирования

Функции приложения реализуются за счет программных модулей. Связи тестируемых программных модулей системы представлены на Рис. 3.

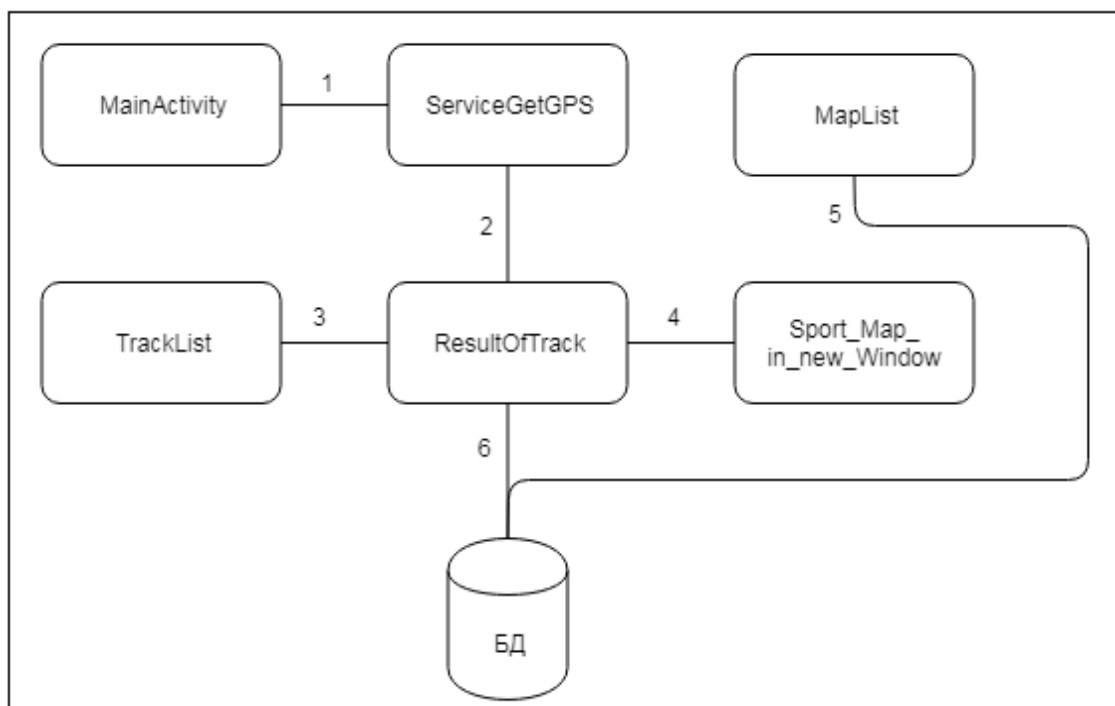


Рисунок 3: Схема взаимодействия модулей системы

1. MainActivity взаимодействует с ServiceGetGPS через отправку широковещательных сообщений о начале запуска сбора gps-координат, остановки, прекращении.
2. После окончания записи трека и его сохранения в памяти устройства ServiceGetGPS передаёт название записанного трека для отображения результатов тренировки.
3. При выборе конкретного трека из списка всех треков, хранящихся на устройстве TrackList передаёт имя выбранного трека ResultOfTrack для отображения более подробной информации о тренировке.
4. При просмотре результатов тренировки существует возможность наложить трек на спортивную карту, само наложение происходит в модуле Sport\_map\_in\_new\_window. ResultOfTrack передаёт файл трека и имя выбранной карты из памяти устройства для наложения.
5. MapList загружает новые карты из БД.
6. При просмотре результатов тренировки есть возможность передать данные тренировки на сервер. В БД передаётся файл тренировки, сохранённый в памяти устройства.

В рамках данной работы будут протестированы следующие компоненты (с описанием):

- Метод getLine модуля ResultOfTrack (Таблица 2.1);
- Метод readTrack модуля ResultOfTrack (Таблица 2.2);
- Метод getMainParameters модуля ResultOfTrack (Таблица 2.3);
- Метод sendTrack модуля ResultOfTrack (Таблица 2.4);
- Метод createTempforKilometer модуля ResultOfTrack (Таблица 2.5);
- Метод saveTrack модуля ServiceGetGPS (Таблица 2.6).

Тестирование будет проводится только для вышеописанных методов по следующим причинам:

- Определенное количество остальных методов являются "простыми то есть в них маловероятна возможность допущения какой-либо ошибки;



- Остальные модули не являются критичными для системы (то есть, не являются основополагающими), в отличие от модулей, которые будут протестированы в рамках данной работы;
- Приложение объёмное, и оно не может быть в полной мере протестировано в рамках прохождения данного направления.

## **2.2 Стратегия блочного тестирования**

Блочное тестирование будет проводиться методом автоматизированного тестирования с использованием фреймворка JUnit. Этот фреймворк позволяет конфигурировать окружение для тестов, исполнять код, который будет проверять работу некоторых классов, и выводить результаты тестов (сколько успешных тестов, сколько ошибок, и где именно произошли ошибки).

Блочное тестирование должно быть применено к следующим методам: `getLine`, `readTrack`, `getMainParameters`, `sendTrack`, `createTempforKilometer`, `saveTrack`.

Для каждого пути в блок схеме должен быть разработан как минимум один тест.

## **2.3 Стратегия интеграционного тестирования**

Блочное тестирование будет проводиться методом автоматизированного тестирования с использованием фреймворка Espresso.

Для проведения интеграционного тестирования необходимо определить все возможные входные данные, соответствующие им ожидаемые результаты. Интеграционное тестирование будет проведено для следующих взаимодействий между модулями: `ServiceGetGPS` – `ResultOfTrack`, `ResultOfTrack` – БД.

## **2.4 Стратегия аттестационного тестирования**

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор тестирования. Тестирующий человек, по заранее заданным инструкциям (`TestCases`), производит требуемые действия и сверяется с заранее заданными результатами. Тест считается пройденным, если ожидаемый результат совпадает с фактическим результатом. В противном случае тест считается не пройденным.

## **2.5 Стратегия нагрузочного тестирования**

Нагрузочное тестирование будет проводиться методом автоматизированного тестирования с использованием фреймворка Espresso. Для проведения нагрузочных тестов должна быть использована копия структуры базы данных, заполнение которой будет осуществляться при помощи фикстур, разработанных специально для нагрузочных тестовых кейсов.

В рамках нагрузочного тестирования будет проверяться скорость работы методов `getMainParameters`, `getLine`, `createTempforKilometer` модуля `ResultOfTrack` для различного количества grs-координат (100, 500, 1 000 и 2000 точек).

## **2.6 Критерии прохождения тестирования**

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Если тест завершается неудачей, то перед принятием решения целесообразно проверить правильность самого теста. Если тест завершился неудачей и тест реализован правильно, то производится заключение о найденной ошибке. Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок и количество пройденных тестов составляет не менее 85% от общего числа тестов.

## **2.6 Условия возобновления и приостановки выполнения тестов**

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 15% от их общего количества. Тестирование должно быть приостановлено при обнаружении критических ошибок.

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем тестировании. Повторное тестирование должно быть выполнено с самого начала, начиная с блочных тестов.

### 3. Детальный план тестов

#### 3.0 Фикстуры

Перед выполнением каждого теста таблицы Tracks и Points копии базы данных должны быть заполнены при помощи фикstur данными, представленными в таблицах 3.0.1 и 3.0.2. Заполнение таблиц выполняется единократно перед началом аттестационного тестирования. Перед запуском аттестационных тестов необходимо получить данные с сервера на мобильное устройство, на котором выполняется тестирование. Для выполнения теста A13 необходимо предварительно удалить трек «2020-04-12» из БД.

Перед запуском интеграционных тестов необходимо предварительно получить данные с сервера на мобильное устройство, на котором выполняется тестирование. Для выполнения интеграционных тестов И5, И6 и И7 необходимо предварительно удалить следующие треки из БД: «2020-05-12», «2020-07-12», «2020-08-12».

Таблицы Tracks и Points создаются и заполняются при помощи sql-скриптов, представленных на листингах 1, 2 и 3.

Для выполнения нагрузочного тестирования таблицы Tracks и Points должны быть заполнены согласно листингам 4 и 5 соответственно.

Таблица 3.0.1. Фикстуры для таблицы Tracks

Id_track	Track_date	Track_time	Distance	averageTemp	rise	descent	maxTemp	minTemp	maxLongHeight	maxLongDescent
1	2020-04-12	60	100	12	14	13	10	14	4	7
2	2020-05-12	30	50	10	7	4	Null	Null	3	4
3	2020-06-12	610	2000	6	13	9	5	12	17	17
4	2020-07-12	480	1500	6	23	12	4	8	13	9
5	2020-08-12	0	0	0	0	0	0	0	0	0

Таблица 3.0.2. Фикстуры для таблицы Points

Id_point	latitude	longitude	Id_track	Point_time	Point_num
1	61.776785	34.341607	1	1	1
2	61.777135	34.342331	1	15	1
3	61.777792	34.343854	1	60	1
4	61.776785	34.341607	2	1	1
5	61.777135	34.342331	2	30	1
6	61.777792	34.343854	3	1	1
7	61.769016	34.322595	3	200	1
8	61.778337	34.309956	3	400	1
9	61.769016	34.322595	4	1	1
10	61.778337	34.309956	4	300	1
11	61.781405	34.305586	4	480	1

---

```
CREATE TABLE Tracks (  
  id_track INT(10) NOT NULL AUTO_INCREMENT,  
  track_date DATETIME NOT NULL,  
  track_time INT(10),  
  distance NUMERIC(10,2),  
  averageTemp NUMERIC(10,2),  
  rise NUMERIC(10,2),  
  descent NUMERIC(10,2),  
  maxTemp NUMERIC(10,2),  
  minTemp NUMERIC(10,2),  
  maxLongHeight NUMERIC(10,2),  
  maxLongDescent NUMERIC(10,2),  
  PRIMARY KEY (id_track));
```

---

Листинг 1. Фикстуры для таблицы Tracks в рамках интеграционного и аттестационного тестирования.

---

```
CREATE TABLE Points (  
  id_point INT(10) NOT NULL AUTO_INCREMENT,  
  latitude FLOAT(2,6) NOT NULL,  
  longitude FLOAT(2,6) NOT NULL,  
  id_track INT(10) NOT NULL,  
  Point_time INT(10) NOT NULL,  
  Point_num INT(10) NOT NULL,  
  PRIMARY KEY (id_point),  
  FOREIGN KEY (id_track) REFERENCES Tracks (id_track) ON DELETE CASCADE);
```

---

Листинг 2. Фикстуры для таблицы Points в рамках интеграционного и аттестационного тестирования.

---

```
INSERT INTO Tracks VALUES (2020-03-12, 60, 100, 12, 14, 13, 10, 14, 4, 7);  
INSERT INTO Points VALUES (61.776785, 34.341607, 1, 1, 1);
```

---

Листинг 3. Пример заполнения таблиц Tracks, Points в рамках интеграционного и аттестационного тестирования.

---

```
$array = [];  
$latitude = 61.776785;  
$longitude = 34.341607;  
for ($i = 1; $i <= 5000; $i++) {  
  $latitude = $latitude + 0.000001;  
  $longitude = $longitude + 0.000001;  
  $array[] = [  
    'id_point' => $i,  
    'latitude' => $latitude,  
    'id_track' => 1,  
    'point_time' => $i,  
    'point_num' => 1,  
  ];  
}  
return $array;
```

---

Листинг 4. Фикстуры для таблицы Points в рамках нагрузочного тестирования.

```
INSERT INTO Tracks VALUES (2020-03-12, 60, 100, 12, 14, 13, 10, 14, 4, 7);
```

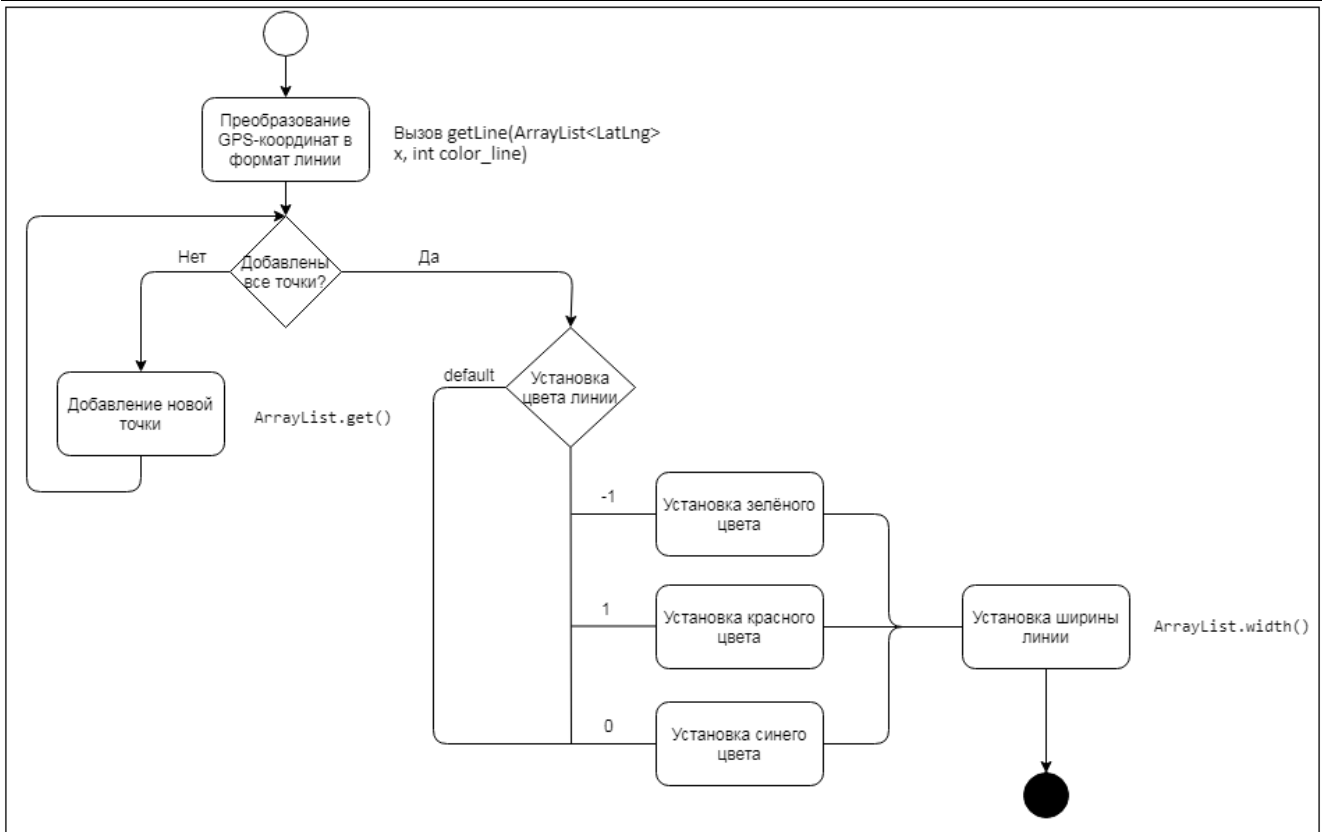
Листинг 5. Заполнение таблицы Tracks в рамках нагрузочного тестирования.

### 3.1 Блочное тестирование

#### 3.1.1 Метод getLine

Таблица 3.1.1.0. Метод getLine

Объект	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Расположение	app/src/main/java/com/example/user/o_track/ResultOfTrack
Назначение	Преобразование gps-координат в формат линии, установка цвета линии и ширины.
Входные данные	ArrayList<LatLng> x – список gps-координат линии color_line – цвет линии
Выходные данные	PolylineOptions line – список точек для отрисовки линий: <ul style="list-style-type: none"><li>• Points – список точек</li><li>• Width – ширина линии</li><li>• Color – цвет линии.</li></ul>



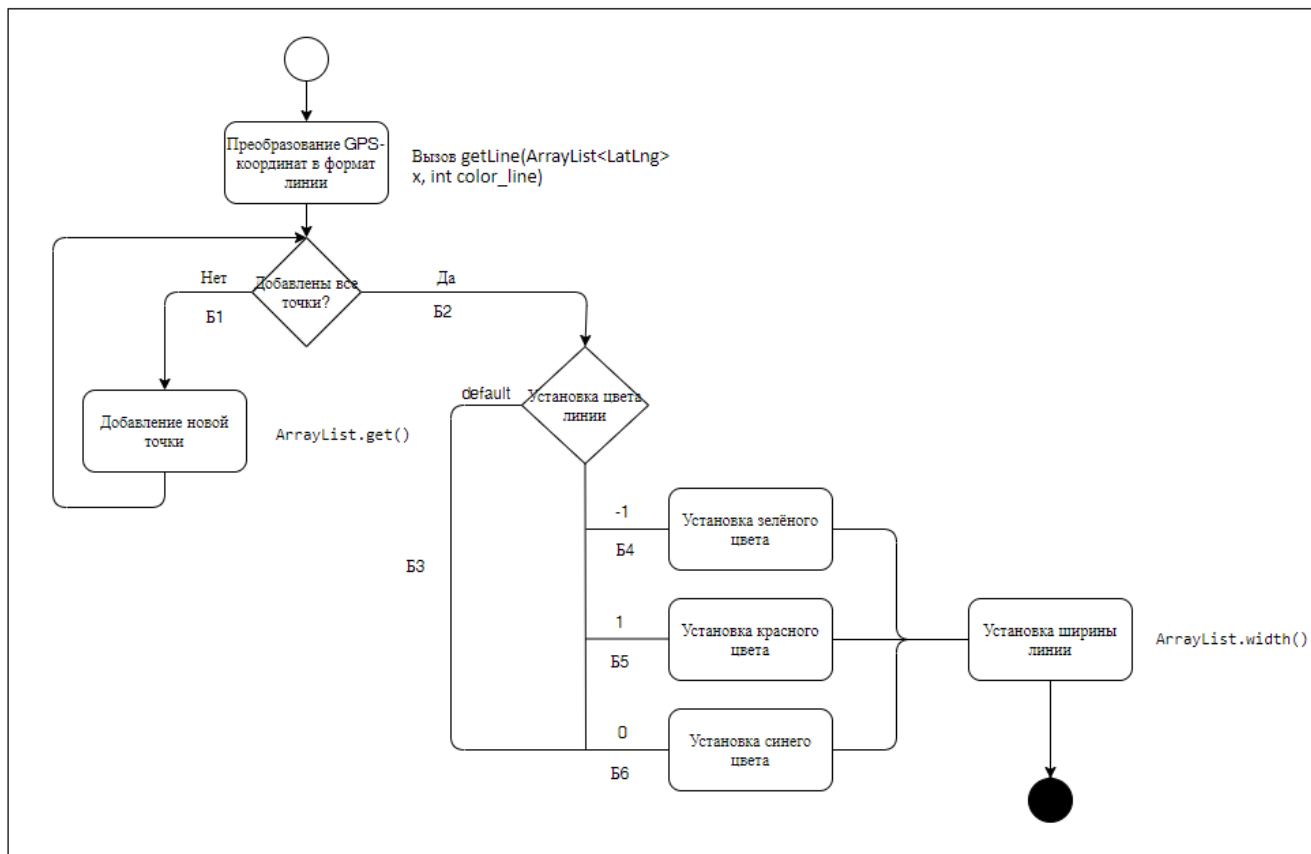


Таблица 3.1.1.1.1 Тест № Б1.1

№ теста	Б1.1
Цель теста (описание)	Общий, положительный Проверка пути Б1: преобразование gps-координат в формат линии, все gps-координаты корректные.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": 61,765, "Lng": 34,543)
Ожидаемый результат	null

Таблица 3.1.1.1.2 Тест № Б1.2

№ теста	Б1.2
Цель теста (описание)	Общий, отрицательный Проверка пути Б1: преобразование gps-координат в формат линии, неверно поданная gps-координата.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": -1, "Lng": 34,543)
Ожидаемый результат	null

Таблица 3.1.1.2 Тест № Б2

№ теста	Б2
Цель теста (описание)	Общий, положительный Проверка пути Б2 и Б3: преобразование gps-координат в формат линии, все gps-координаты корректные. Установка цвета по умолчанию – синий цвет.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": 61,765, "Lng": 34,543), color_line – любое значение не из списка (0, 1, -1)
Ожидаемый результат	PolylineOptions line:

	<ul style="list-style-type: none"> <li>• Points = [61,765, 34,543]</li> <li>• Color = rgb(0,0,255)</li> </ul>
--	---

Таблица 3.1.1.3 Тест № Б3

№ теста	Б3
Цель теста (описание)	Общий, положительный Проверка пути Б2 и Б6: преобразование gps-координат в формат линии, все gps-координаты корректные. Установка цвета – синий цвет.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": 61,765, "Lng": 34,543), color_line = 0.
Ожидаемый результат	PolylineOptions line: <ul style="list-style-type: none"> <li>• Points = [61,765, 34,543]</li> <li>• Color = rgb(0,0,255)</li> </ul>

Таблица 3.1.1.4 Тест № Б4

№ теста	Б4
Цель теста (описание)	Общий, положительный Проверка пути Б2 и Б5: преобразование gps-координат в формат линии, все gps-координаты корректные. Установка цвета – красный цвет.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": 61,765, "Lng": 34,543), color_line = 1.
Ожидаемый результат	PolylineOptions line: <ul style="list-style-type: none"> <li>• Points = [61,765, 34,543]</li> <li>• Color = rgb(255,0,0)</li> </ul>

Таблица 3.1.1.5 Тест № Б5

№ теста	Б5
Цель теста (описание)	Общий, положительный Проверка пути Б2 и Б4: преобразование gps-координат в формат линии, все gps-координаты корректные. Установка цвета – зелёный цвет.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": 61,765, "Lng": 34,543), color_line = 1.
Ожидаемый результат	PolylineOptions line: <ul style="list-style-type: none"> <li>• Points = [61,765, 34,543]</li> <li>• Color = rgb(0,255,0)</li> </ul>

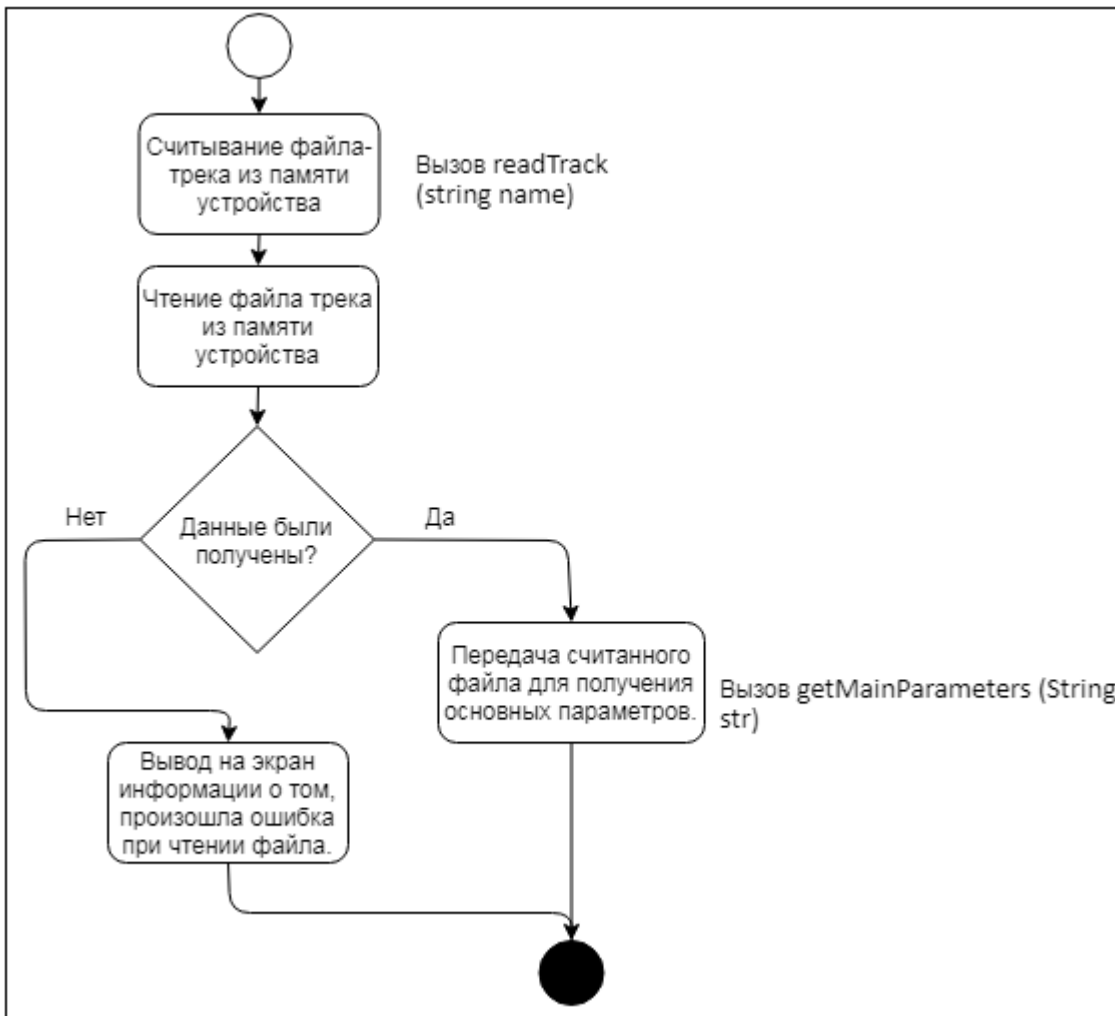
Таблица 3.1.1.6 Тест № Б6

№ теста	Б6
Цель теста (описание)	Общий, отрицательный Проверка пути Б1, Б2, Б3: преобразование gps-координат в формат линии, неверно поданная gps-координата. Линия не должна создаваться.
Объект тестирования	PolylineOptions getLine(ArrayList<LatLng> x, int color_line)
Входные данные	ArrayList("Lat": -1, "Lng": 34,543) , color_line = 0.
Ожидаемый результат	null

### 3.1.2 Метод readTrack

Таблица 3.1.2.0. Метод readTrack

Объект	Public string readTrack (string name)
Расположение	app/src/main/java/com/example/user/o_track/ResultOfTrack
Назначение	Счиывание файла трека из памяти устройства.
Входные данные	string name – имя файла - трека
Выходные данные	<ul style="list-style-type: none"> <li>• Str – строка с содержимым файла трека</li> <li>• Null – ошибка чтения файла.</li> </ul>





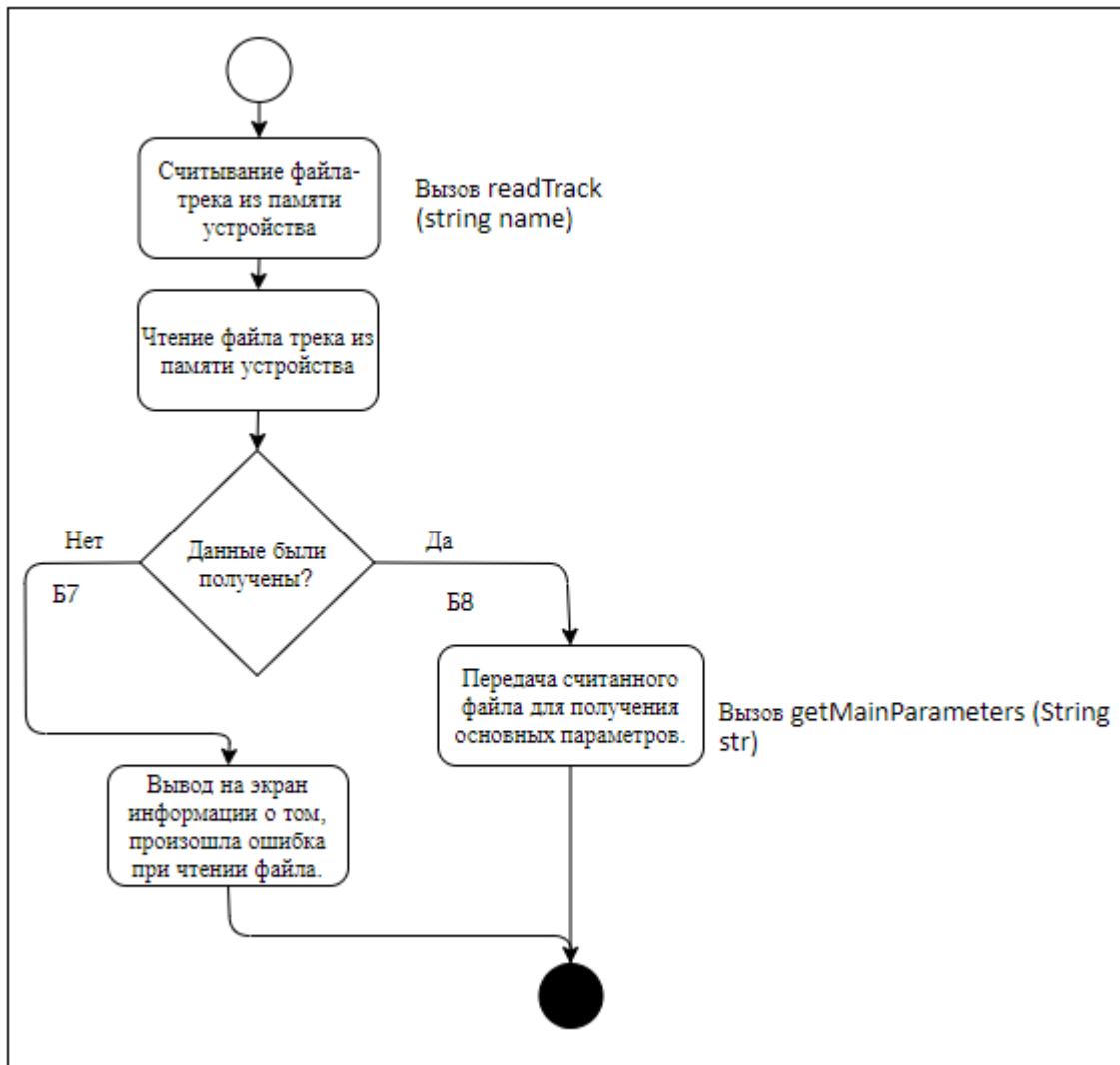


Таблица 3.1.2.1 Тест № Б7

№ теста	Б7
Цель теста (описание)	Общий, отрицательный. Проверка пути Б7: ошибка при считывании файла трека из памяти устройства.
Объект тестирования	Public string readTrack (string name)
Входные данные	String name = null
Ожидаемый результат	-1

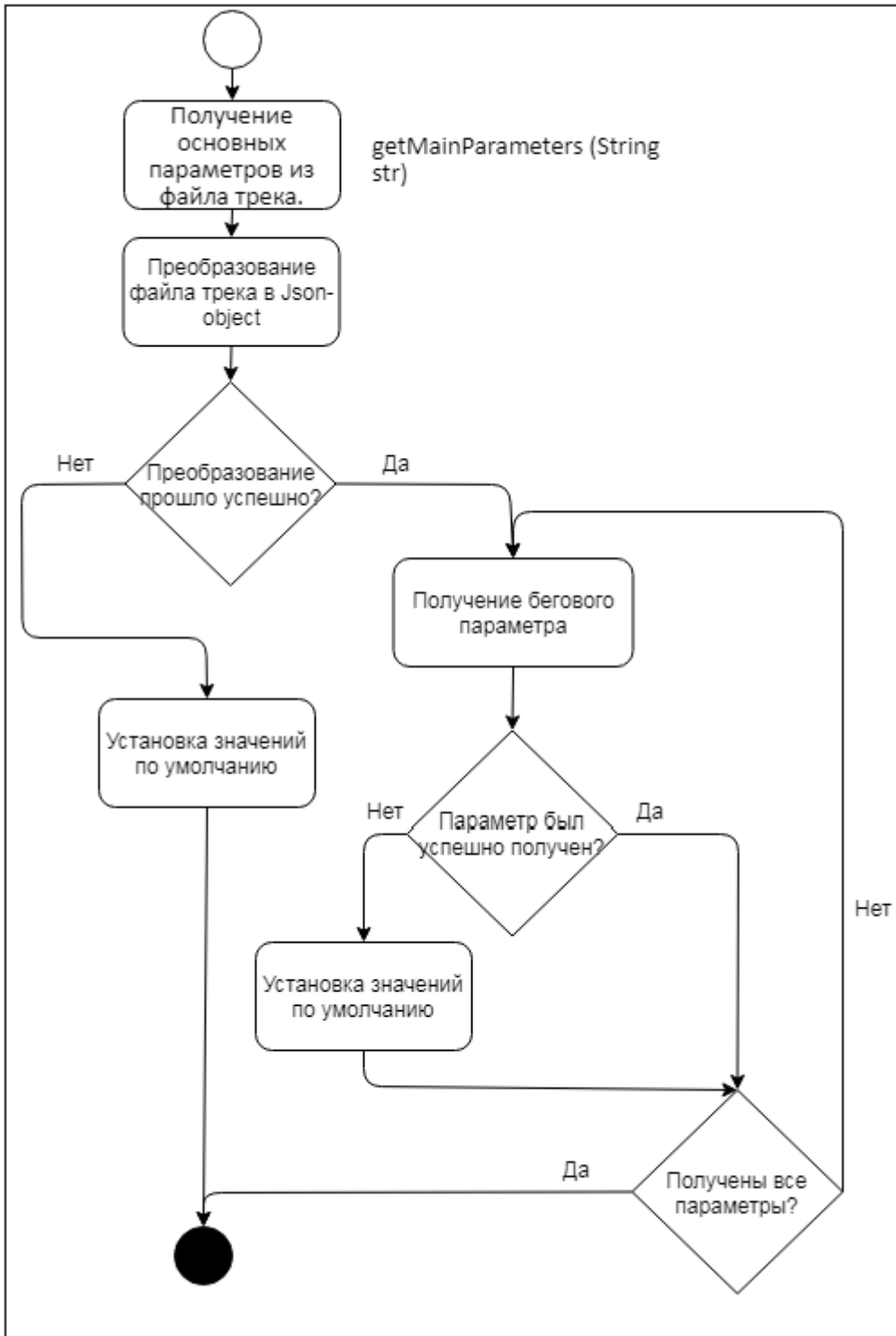
Таблица 3.1.2.2 Тест № Б8

№ теста	Б8
Цель теста (описание)	Общий, положительный. Проверка пути Б8: файл трека успешно считан, отправка на сервер.
Объект тестирования	Public string readTrack (string name)
Входные данные	String name = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"}}
Ожидаемый результат	{"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} {"1":{"Time":152373231, "Distance":"0", "averageTemp":"0", "rise":0, "descent":"0", "maxTemp":"0", "minTemp":"0", "maxLongHeight":"0", "maxLongDescent":"0"}}

### 3.1.3 Метод getMainParameters

Таблица 3.1.3.0. Метод getMainParameters

Объект	Public getMainParameters (String str)
Расположение	app/src/main/java/com/example/user/o_track/ResultOfTrack
Назначение	Считывание основных параметров из файла трека.
Входные данные	String str – файл трека.
Выходные данные	<p>Array – массив, содержащий основные параметры забега:</p> <ul style="list-style-type: none"> <li>• Time – время забега</li> <li>• Distance - километраж</li> <li>• averageTemp – средний темп</li> <li>• rise – общий спуск</li> <li>• descent – общий подъём</li> <li>• maxTemp – максимальный темп</li> <li>• minTemp – минимальный темп</li> <li>• maxLongHeight – максимальная высота</li> <li>• maxLongDescent – минимальная высота.</li> <li>• 0 – в случае если не удалось считать какие-либо данные.</li> </ul>



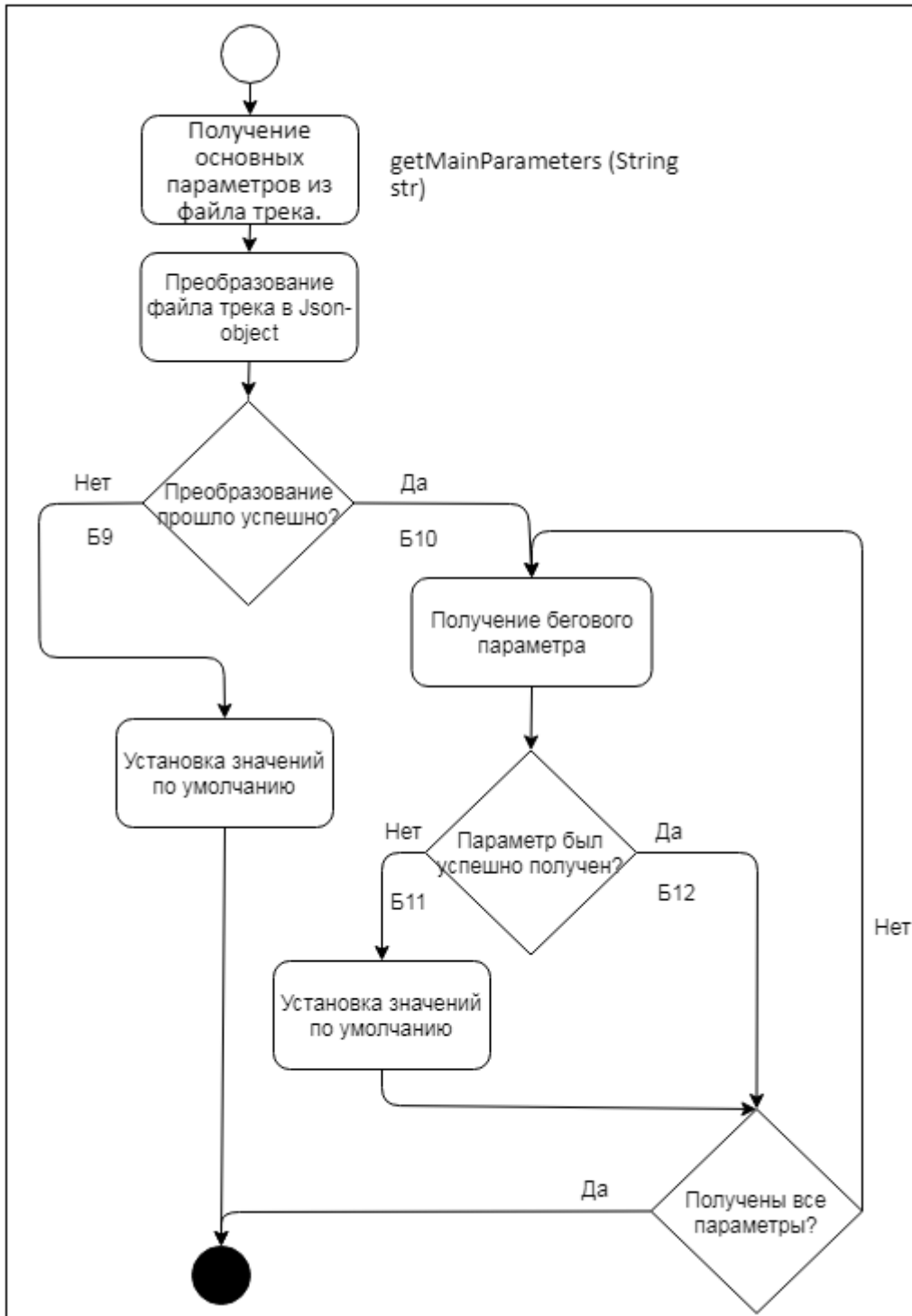


Таблица 3.1.3.1 Тест № Б9

№ теста	Б9
Цель теста (описание)	Общий, отрицательный. Проверка пути Б9: преобразование в JSON-Object произошло не успешно, установились параметры по умолчанию.
Объект тестирования	Public getMainParameters (String str)
Входные данные	String name = null
Ожидаемый результат	Array("Time": 0, "Distance": 0, "averageTemp": 0, "rise": 0, "descent": 0, "maxTemp": 0, "minTemp": 0, "maxLongHeight": 0, "maxLongDescent": 0)

**Таблица 3.1.3.2 Тест № Б10**

№ теста	Б10
Цель теста (описание)	Общий, положительный. Проверка пути Б10, Б12: преобразование в JSON-Object произошло успешно, все параметры получены успешно.
Объект тестирования	Public getMainParameters (String str)
Входные данные	String name = [{"Time": "32", "Distance ": "3200", " averageTemp ": "5.06", " rise ": "46", " descent ": "13", " maxTemp ": "3.01", " minTemp ": "10", " maxLongHeight ": "150", " maxLongDescent ": "120"}]
Ожидаемый результат	Array("Time": 32, "Distance": 3200, "averageTemp": 5.06, "rise": 46, "descent": 13, "maxTemp": 3.01, "minTemp": 10, "maxLongHeight": 150, "maxLongDescent": 120)

**Таблица 3.1.3.3.1 Тест № Б11.1**

№ теста	Б11.1
Цель теста (описание)	Общий, отрицательный. Проверка пути Б10, Б11, Б12: преобразование в JSON-Object произошло успешно, часть параметров получена не успешно.
Объект тестирования	Public getMainParameters (String str)
Входные данные	String name = [{"Time": "32", "Distance ": "null", " averageTemp ": "5.06", " rise ": "46", " descent ": "13", " maxTemp ": "null", " minTemp ": "10", " maxLongHeight ": "150", " maxLongDescent ": "120"}]
Ожидаемый результат	Array("Time": 32, "Distance": 0, "averageTemp": 5.06, "rise": 46, "descent": 13, "maxTemp": 0, "minTemp": 10, "maxLongHeight": 150, "maxLongDescent": 120)

**Таблица 3.1.3.3.2 Тест № Б11.2**

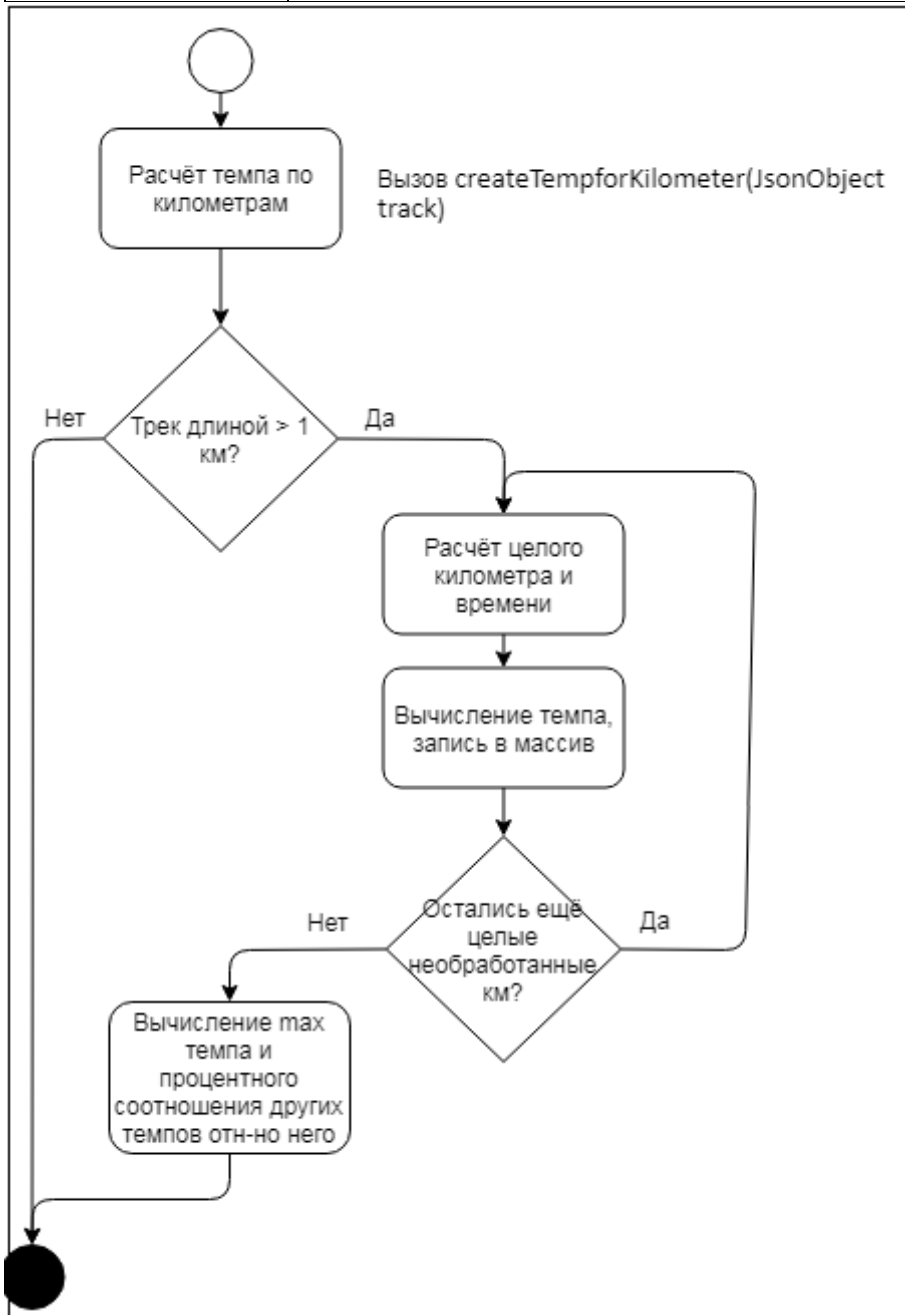
№ теста	Б11.2
Цель теста (описание)	Общий, отрицательный. Проверка пути Б10, Б11, Б12: преобразование в JSON-Object произошло успешно, часть параметров получена не успешно и часть данных нет в JSON-Object.
Объект тестирования	Public getMainParameters (String str)
Входные данные	String name = [{"Time": "32", "Distance ": "null", " averageTemp ": "5.06", " rise ": "46", " descent ": "13", " maxTemp ": "null", " minTemp ": "10", " maxLongHeight ": "150"}]
Ожидаемый результат	Array("Time": 32, "Distance": 0, "averageTemp": 5.06, "rise": 46, "descent": 13, "maxTemp": 0, "minTemp": 10, "maxLongHeight": 150, "maxLongDescent": 0)

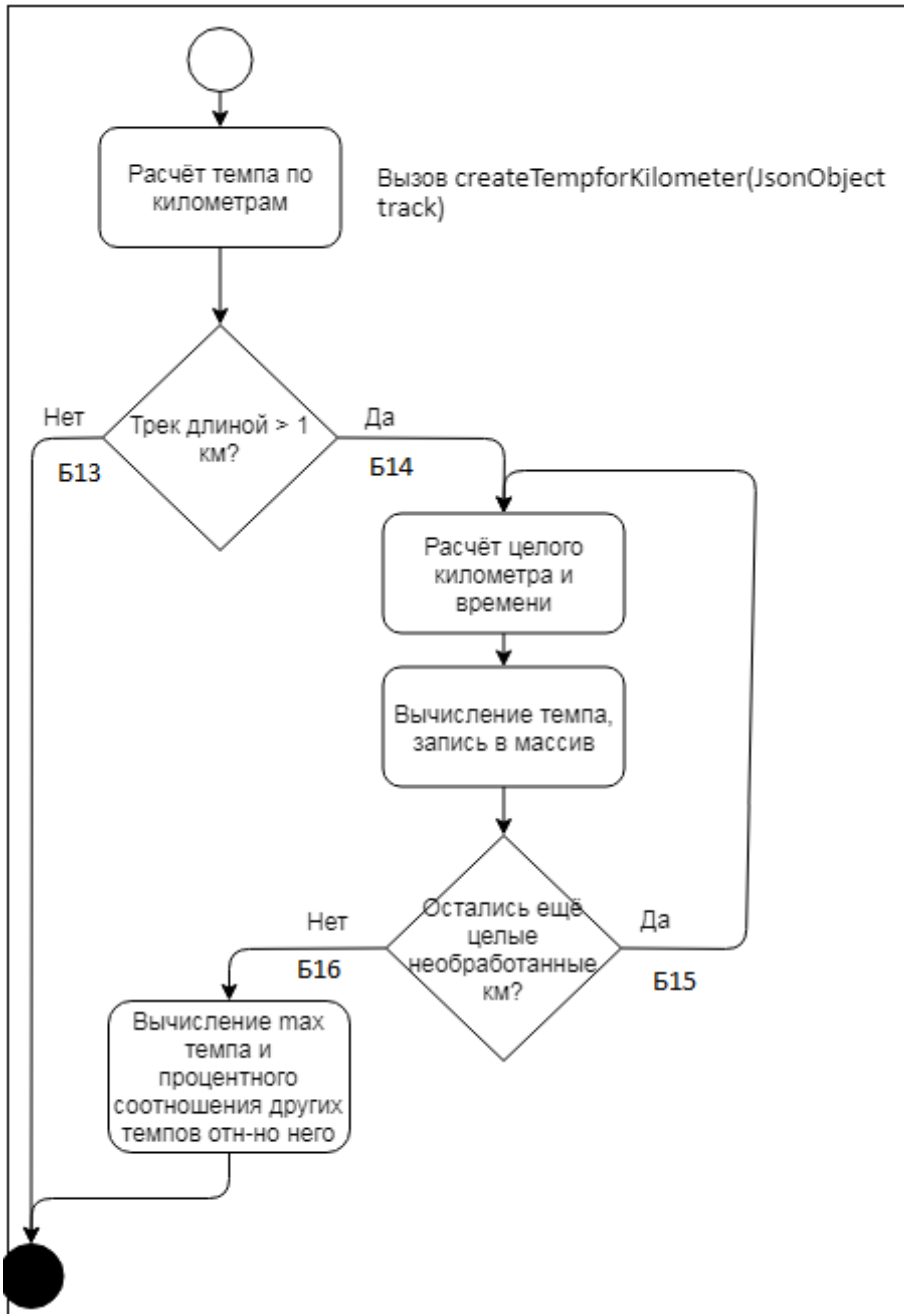
### 3.1.4 Метод createTempforKilometer

Таблица 3.1.4.0. Метод createTempforKilometer

Объект	Public createTempforKilometer(JsonObject track)
Расположение	app/src/main/java/com/example/user/o_track/ResultOfTrack

Назначение	Расчёт среднего темпа по километрам.
Входные данные	Track – записанный трек тренировки.
Выходные данные	Array – массив значений среднего темпа ко км.





**Таблица 3.1.4.1 Тест № Б13**

№ теста	Б13
Цель теста (описание)	Общий, отрицательный. Проверка пути Б13: Получен трек общей протяжённостью меньше 1 км, расчёт темпа по километрам не проводится.
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} "1":{"type":1, "latitude":"61,766", "longitude":"34,544", "time":152373256, "altitude":"154,28"}}
Ожидаемый результат	Null

**Таблица 3.1.4.2 Тест № Б14.1**

№ теста	Б14.1
Цель теста (описание)	Общий, положительный. Проверка пути Б14, Б16: Получен трек общей протяжённостью > 1 км, но < 2 км, расчёт темпа производится только для 1-го км.
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} "1":{"type":1, "latitude":"61,82", "longitude":"34,91", "time":152373256, "altitude":"154,28"} "2":{"type":1, "latitude":"61,95", "longitude":"34,95", "time":152374256, "altitude":"154,28"}}
Ожидаемый результат	Array[3.56]

**Таблица 3.1.4.3 Тест № Б14.2**

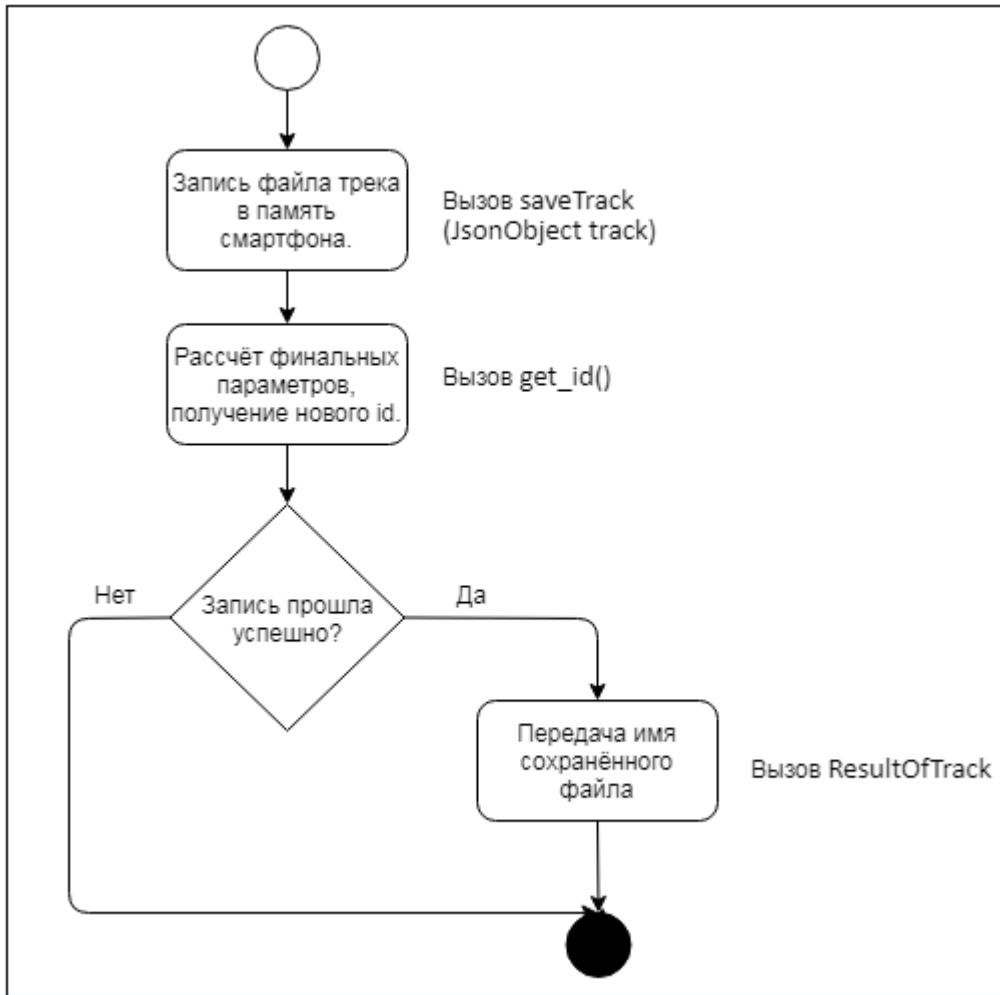
№ теста	Б14.2
Цель теста (описание)	Общий, положительный. Проверка пути Б14, Б15: Получен трек общей протяжённостью > 2 км, но < 3 км, расчёт темпа производится для 1-го и 2-го км.
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} "1":{"type":1, "latitude":"61,82", "longitude":"34,91", "time":152373256, "altitude":"154,28"} "2":{"type":1, "latitude":"62,14", "longitude":"35,41", "time":152374256, "altitude":"154,28"}}
Ожидаемый результат	Array[3.56, 10.17]

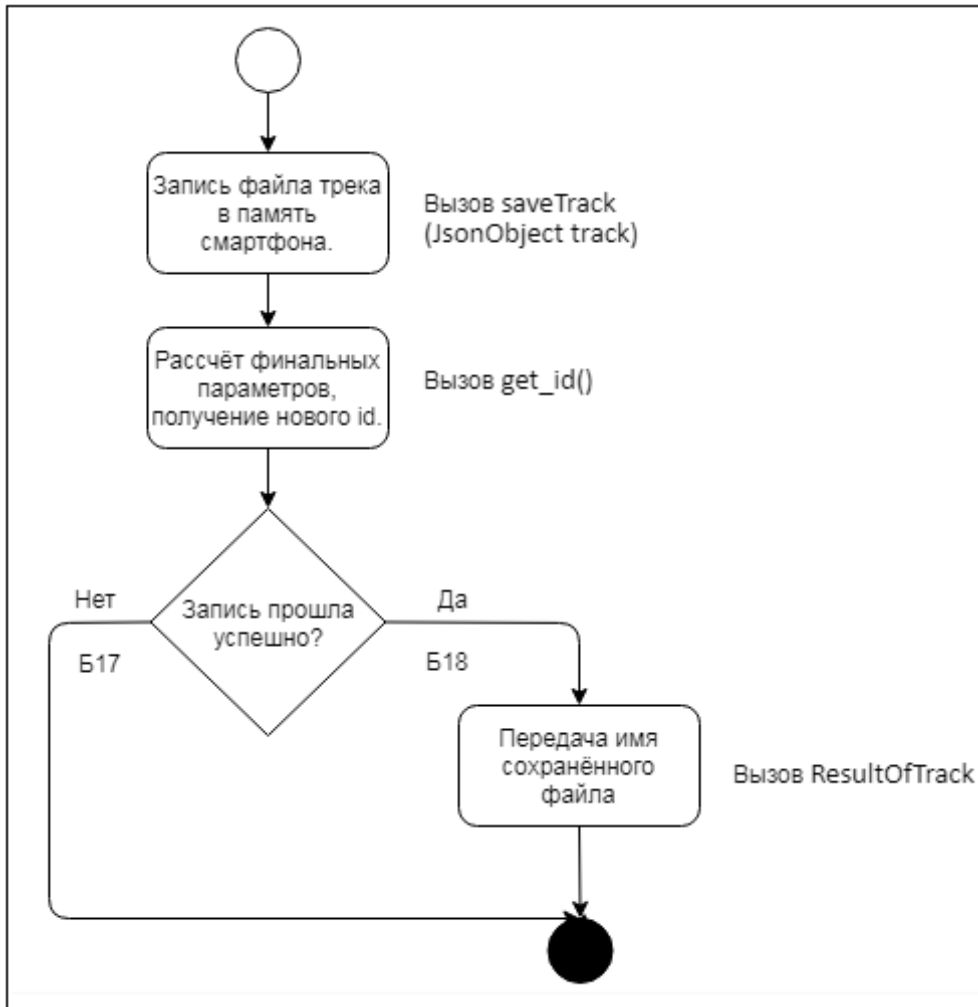
### 3.1.5 Метод saveTrack

Таблица 3.1.5.0 Метод saveTrack

Объект	Private saveTrack (JsonObject track)
Расположение	app/src/main/java/com/example/user/o_track/ServiceGetGPS
Назначение	Сохранение трека в памяти устройства и передача данных о названии сохранённого трека для отображения результатов.
Входные данные	Track – набор GPS координат (трек).
Выходные данные	<ul style="list-style-type: none"> <li>• Null – в случае если запись трека прошла неудачно</li> <li>• Name – название записанного трека.</li> </ul>







**Таблица 3.1.5.1 Тест № Б15**

№ теста	Б15
Цель теста (описание)	Общий, отрицательный. Проверка пути Б17: запись трека в память смартфона произошла не успешно.
Объект тестирования	Private saveTrack (JsonObject track)
Входные данные	Null
Ожидаемый результат	Null

**Таблица 3.1.5.2 Тест № Б16**

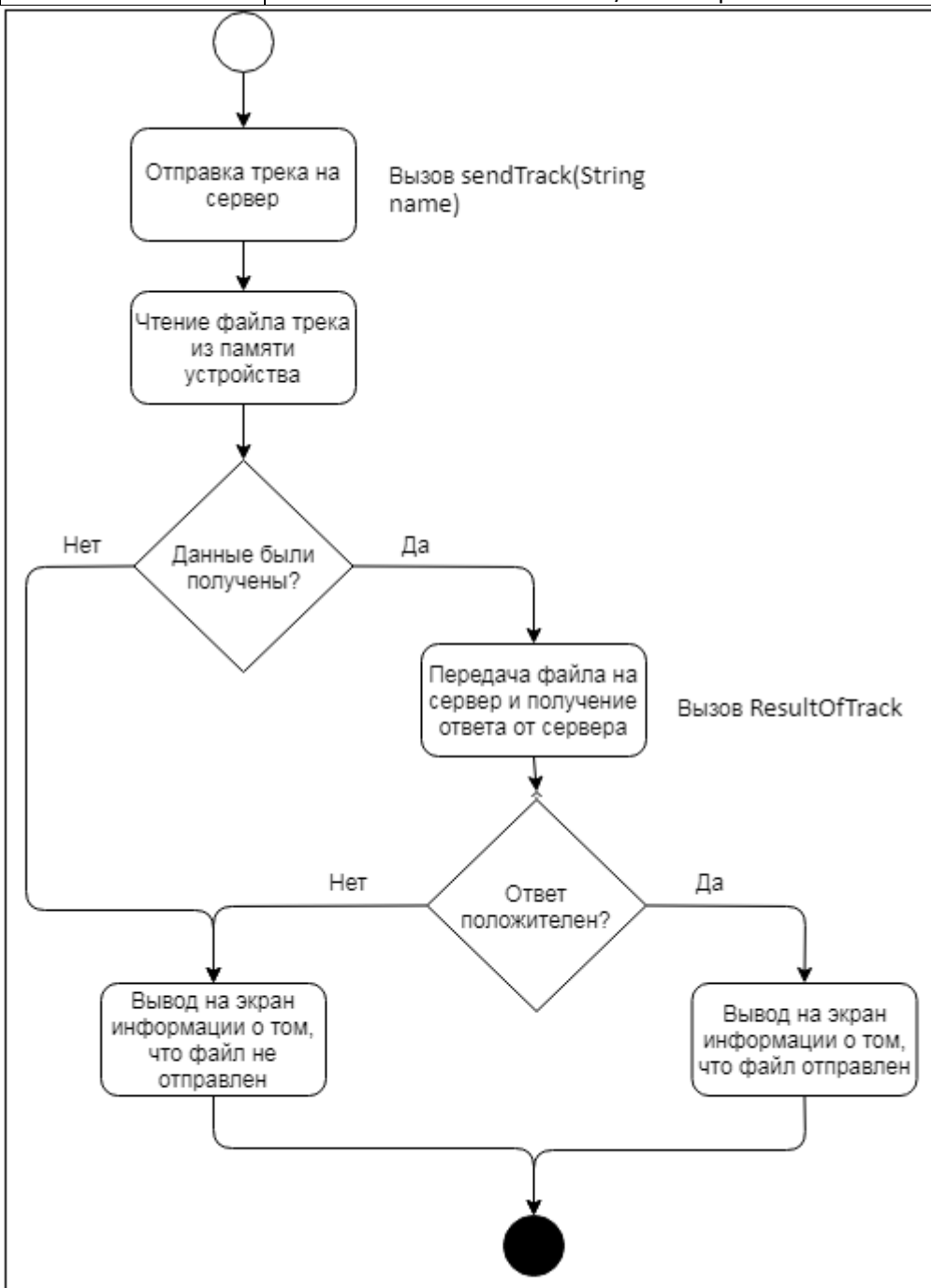
№ теста	Б16
Цель теста (описание)	Общий, положительный. Проверка пути Б18: запись трека в память смартфона произошла успешно.
Объект тестирования	Private saveTrack (JsonObject track)
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longititude":"34,543", "time":152373231, "alltitude":"154,27"} "1":{"type":1, "latitude":"61,82", "longititude":"34,91", "time":152373256, "alltitude":"154,28"} "2":{"type":1, "latitude":"62,14", "longititude":"35,41", "time":152374256, "alltitude":"154,28"}}

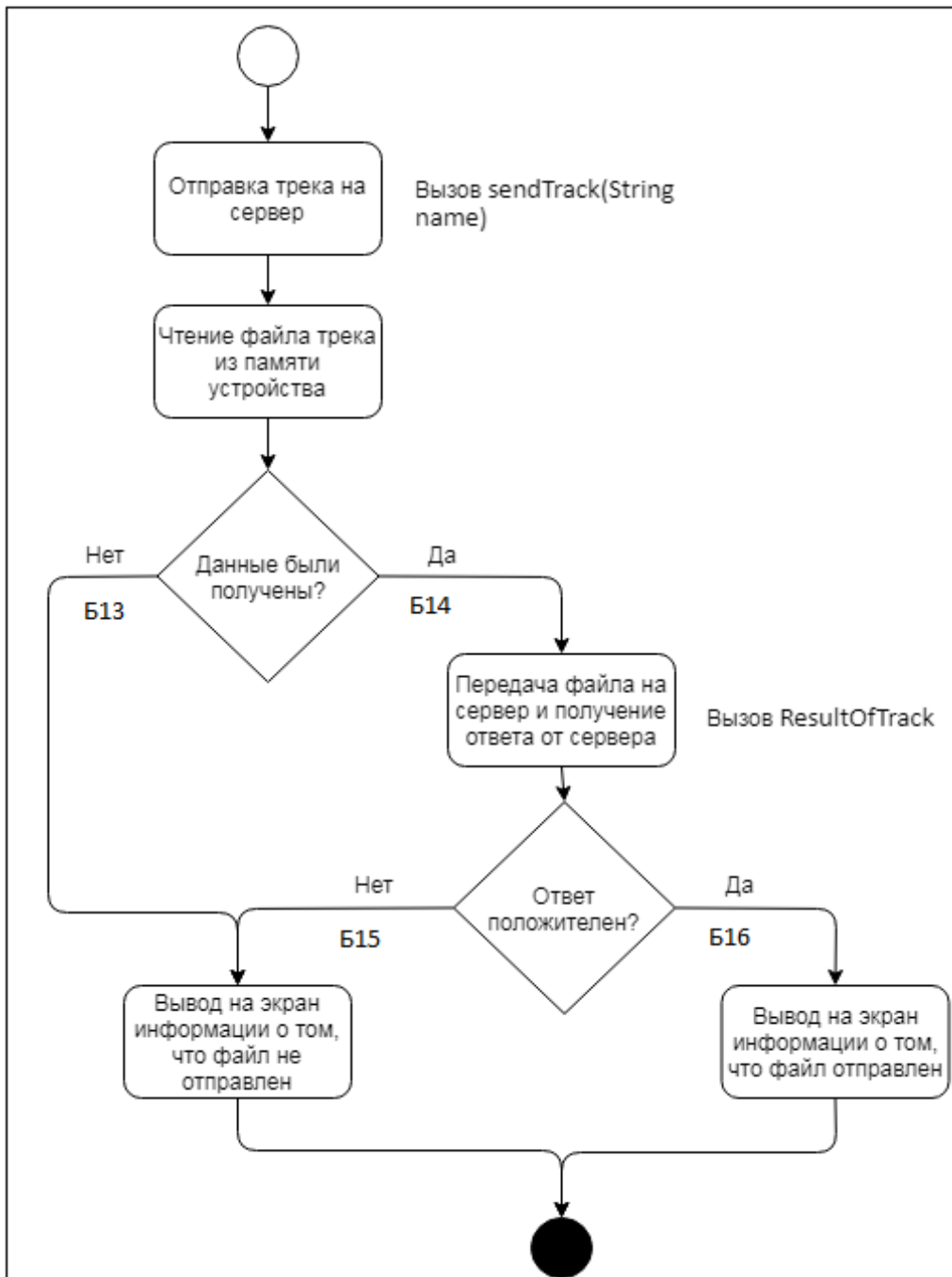
	"3":{"Time": "32", " Distance ": "1537", " averageTemp ": "5.06", " rise ": "46", " descent ": "13", " maxTemp ": "14", " minTemp ": "10", " maxLongHeight ": "150"}}
Ожидаемый результат	20.12.2020_1

### 3.1.6 Метод sendTrack

Таблица 3.1.6.0. Метод sendTrack

Объект	Public sendTrack(String name)
Расположение	app/src/main/java/com/example/user/o_track/ResultOfTrack
Назначение	Чтение трека из памяти устройства и отправка трека на сервер.
Входные данные	Name – имя трека.
Выходные данные	<ul style="list-style-type: none"> <li>• Str – ответ от сервера</li> <li>• Null – ошибка чтения/записи файла.</li> </ul>





**Таблица 3.1.6.1 Тест № Б17**

№ теста	Б17
Цель теста (описание)	Общий, отрицательный. Проверка пути Б13: трек не был найден.
Объект тестирования	Public sendTrack(String name)
Входные данные	String name = 25.12.2020_1
Ожидаемый результат	Null

**Таблица 3.1.6.2 Тест № Б18**

№ теста	Б18
---------	-----

Цель теста (описание)	Общий, положительный. Проверка пути Б14, Б15: трек был считан, правлен на сервер.
Объект тестирования	Public sendTrack(String name)
Входные данные	String name = 14.12.2020_1
Ожидаемый результат	Трек успешно отправлен.

**Таблица 3.1.6.3 Тест № Б19**

№ теста	Б19
Цель теста (описание)	Общий, положительный. Проверка пути Б14, Б16: трек был считан и отправлен на сервер.
Объект тестирования	Public sendTrack(String name)
Входные данные	String name = 14.12.2020_1
Ожидаемый результат	Трек успешно отправлен.

## **3.2 Интеграционное тестирование**

При проведении интеграционного тестирования будут протестированы связи №2 и 6 (рис. 3). Остальные связи не будут протестированы по причине того, что система объёмная, и все связи в ней не могут быть в полной мере протестированы в рамках прохождения данного направления;

### **3.2.1 Интеграция метода saveTrack в метод readTrack**

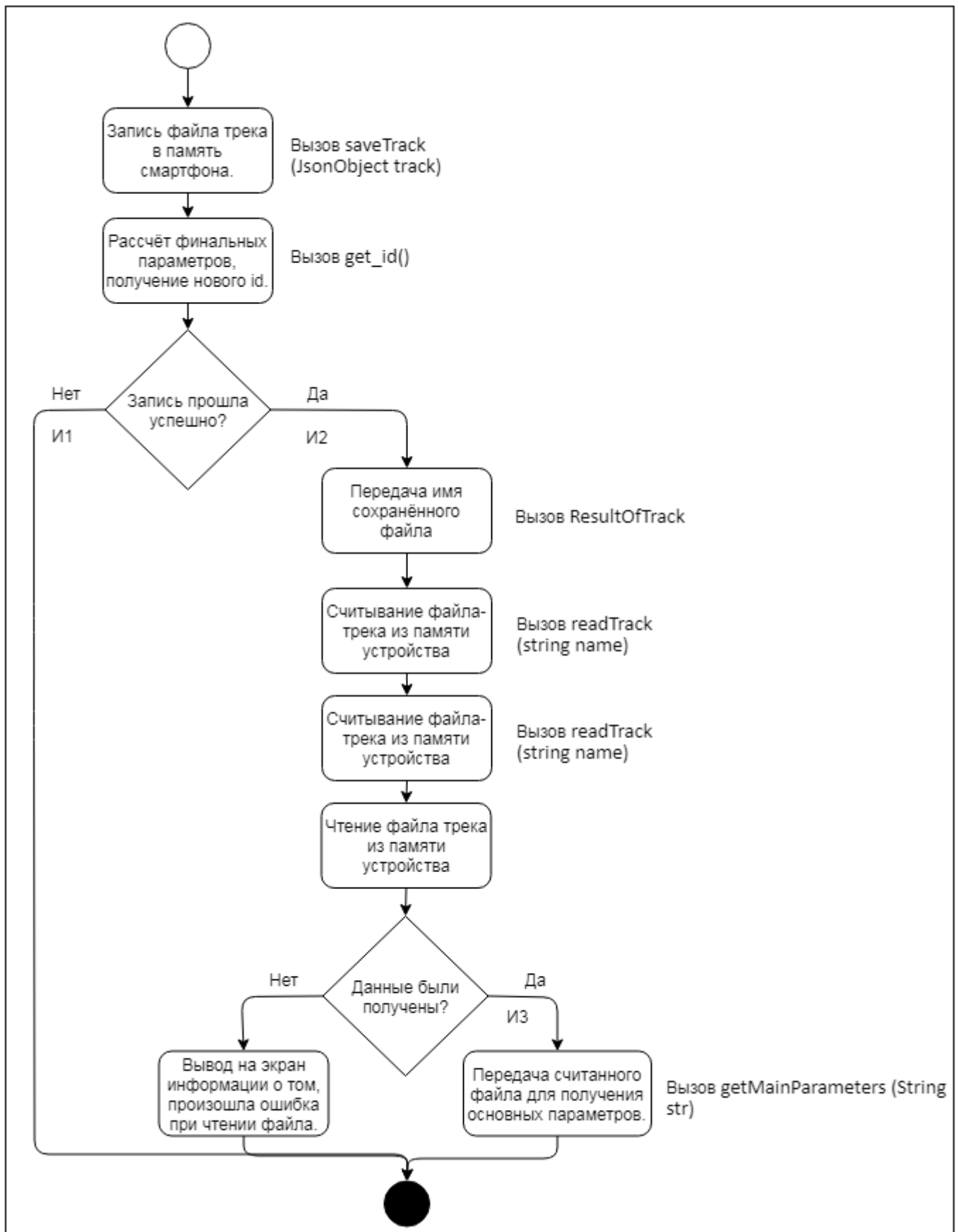


Таблица 3.2.1.1 Тест № И1

№ теста	И1
---------	----

Цель теста (описание)	Общий отрицательный. Тест проверяет отсутствие передачи данных и вызова метода readTrack при неудачной записи трека в память устройства.  Начало теста -> Вызов метода SaveTrack -> неуспешная запись трека в память устройства -> конец теста.
Объект тестирования	Взаимодействие между компонентами saveTrack (ServiceGetGRS) и readTrack (ResultOfTrack).
Входные данные	Null
Ожидаемый результат	Null

**Таблица 3.2.1.2 Тест № И2**

№ теста	И2
Цель теста (описание)	Общий, положительный. Тест проверяет передачу записанных данных в память устройства и вызов метода readTrack.  Начало теста -> Вызов метода SaveTrack -> запись трека в память смартфона -> передача данных -> вызов метода readTrack -> успешное считывание данных тренировки -> конец теста.
Объект тестирования	Взаимодействие между компонентами saveTrack (ServiceGetGRS) и readTrack (ResultOfTrack).
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} "1":{"type":1, "latitude":"61,82", "longitude":"34,91", "time":152373256, "altitude":"154,28"} "2":{"type":1, "latitude":"62,14", "longitude":"35,41", "time":152374256, "altitude":"154,28"} "3":{"Time": "32", " Distance ": "1537", " averageTemp ": "5.06", " rise ": "46", " descent ": "13", " maxTemp ": "14", " minTemp ": "10", " maxLongHeight ": "150"} "4":{"Time":152373476, "Distance":"1543", "averageTemp":"8.07", "rise":12, "descent":7, "maxTemp":4, "minTemp":10.04, "maxLongHeight":5, "maxLongDescent":3}}
Ожидаемый результат	{"Time":152373476, "Distance":"1543", "averageTemp":"8.07", "rise":12, "descent":7, "maxTemp":4, "minTemp":10.04, "maxLongHeight":5, "maxLongDescent":3}

**Таблица 3.2.1.3 Тест № И3**

№ теста	И3
Цель теста (описание)	Общий, положительный. Тест проверяет передачу записанных данных в память устройства (если было всего 2 координаты) и вызов метода readTrack.

	Начало теста -> Вызов метода SaveTrack -> запись трека в память смартфона -> передача данных -> вызов метода readTrack -> успешное считывание данных тренировки -> конец теста.
Объект тестирования	Взаимодействие между компонентами saveTrack (ServiceGetGRS) и readTrack (ResultOfTrack).
Входные данные	track = {"0":{"type":1, "latitude":"61,765", "longitude":"34,543", "time":152373231, "altitude":"154,27"} "1":{"type":1, "latitude":"61,82", "longitude":"34,91", "time":152373256, "altitude":"154,28"} "2":{"Time":24, "Distance":"743", "averageTemp":"4.07", "rise":2, "descent":"7", "maxTemp":"4.34", "minTemp":"12.04", "maxLongHeight":"8", "maxLongDescent":"12"}}
Ожидаемый результат	{"Time":24, "Distance":"743", "averageTemp":"4.07", "rise":2, "descent":"7", "maxTemp":"4.34", "minTemp":"12.04", "maxLongHeight":"8", "maxLongDescent":"12"}

### 3.2.2 Интеграция метода sendTrack и БД



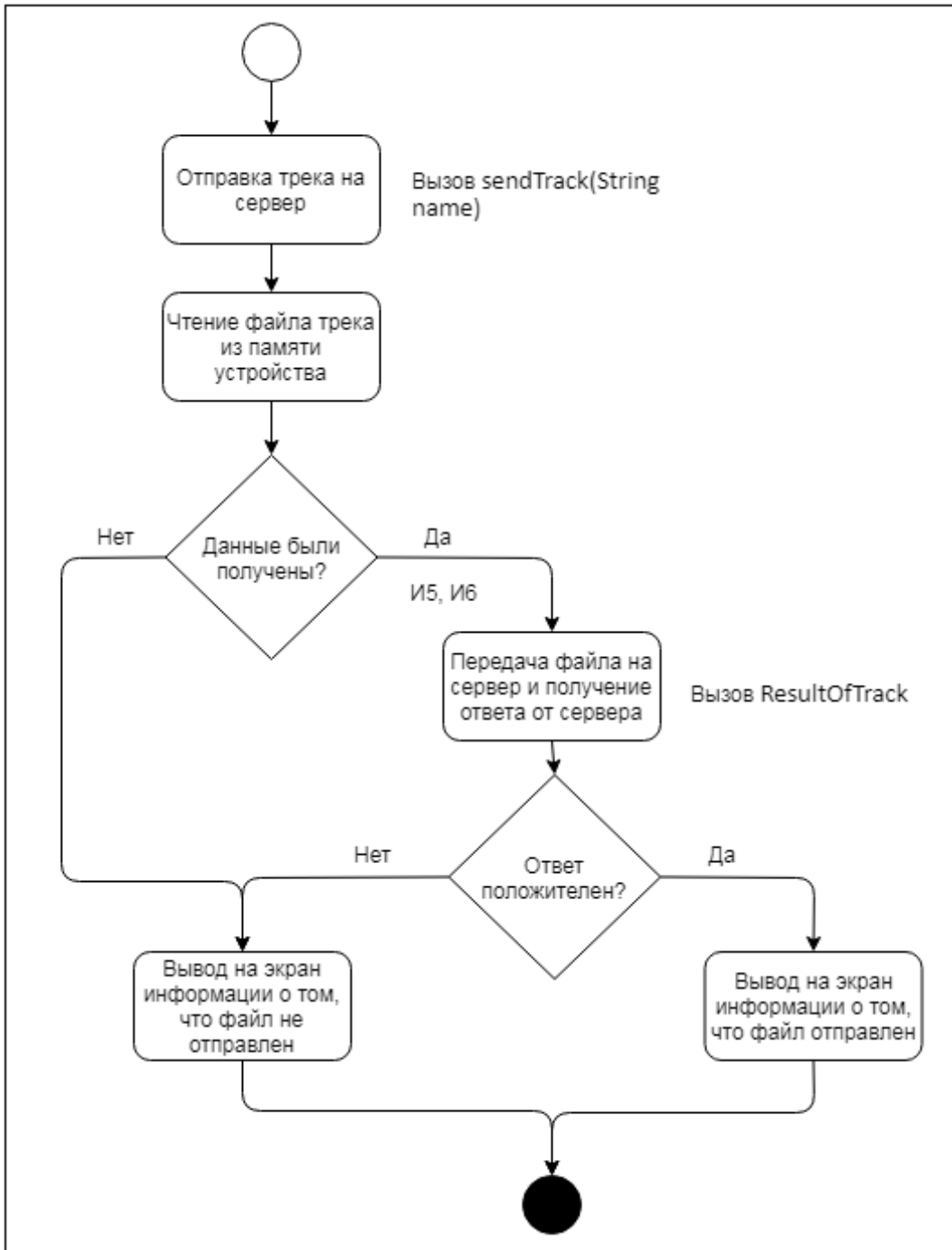


Таблица 3.2.2.1 Тест № И4

№ теста	И4
Цель теста (описание)	Общий, положительный. Отправка трека на сервер с полностью заполненными полями. Начало теста -> Вызов метода sendTrack -> чтение трека из памяти устройства -> отправка трека на сервер в БД -> получение ответа -> конец теста.
Объект тестирования	Взаимодействие между компонентами sendTrack и БД.
Входные данные	2020-07-12
Ожидаемый результат	0

Таблица 3.2.2.2 Тест № И5

№ теста	И5
Цель теста (описание)	Общий, положительный. Отправка трека на сервер с частично не заполненными полями.  Начало теста -> Вызов метода sendTrack -> чтение трека из памяти устройства -> отправка трека на сервер в БД -> получение ответа -> конец теста.
Объект тестирования	Взаимодействие между компонентами sendTrack и БД.
Входные данные	2020-05-12
Ожидаемый результат	0

**Таблица 3.2.2.3 Тест № И6**

№ теста	И6
Цель теста (описание)	Общий, отрицательный. Отправка пустого трека на.  Начало теста -> Вызов метода sendTrack -> чтение трека из памяти устройства -> отправка трека на сервер в БД -> получение ответа -> конец теста.
Объект тестирования	Взаимодействие между компонентами sendTrack и БД.
Входные данные	2020-08-12
Ожидаемый результат	-1

### 3.3 Аттестационное тестирование

**Таблица 3.3.1 Тест № А1**

№ теста	А1
Цель теста (описание)	Общий, положительный. Тест проверяет возможность получения и отображения основных беговых параметров тренировки.
Объект тестирования	Функциональность 1.1 модуля ResultOfTrack.
Входные данные	Тренировка «2020-04-12».
Ожидаемый результат	На экране, в блоке основных беговых показателей отразилась следующая информация: <ul style="list-style-type: none"> <li>• Время забега = 1 мин</li> <li>• Километраж = 100 метров</li> <li>• Средний темп = 12 мин/км</li> <li>• Общий спуск = 14 метров</li> <li>• Общий подъём = 13 метров</li> <li>• Максимальный темп = 10 мин/км</li> <li>• Минимальный темп = 14 мин/км</li> <li>• Максимальный подъём = 4 метра</li> <li>• Максимальный спуск = 7 метров</li> </ul>

**Таблица 3.3.2 Тест № А2**

№ теста	А2
Цель теста (описание)	Общий отрицательный. Тест проверяет возможность получения и отображения основных беговых параметров тренировки в случае если часть значений отсутствует.
Объект тестирования	Функциональность 1.1 модуля ResultOfTrack.
Входные данные	Тренировка «2020-05-12».
Ожидаемый результат	В полях где значений нет должен быть проставлен «-». На экране, в блоке основных беговых показателей отразилась следующая информация: <ul style="list-style-type: none"> <li>• Время забега = 30 сек</li> <li>• Километраж = 50 метров</li> <li>• Средний темп = 10 мин/км</li> <li>• Общий спуск = 7 метров</li> <li>• Общий подъём = 4 метров</li> <li>• Максимальный темп = -</li> <li>• Минимальный темп = -</li> <li>• Максимальный подъём = 3 метра</li> <li>• Максимальный спуск = 4 метров</li> </ul>

**Таблица 3.3.3 Тест № А3**

№ теста	А3
Цель теста (описание)	Общий, положительный. Тест проверяет возможность получения и преобразование основных беговых параметров тренировки из базовой единицы измерения (сек -> мин, метры -> км) с последующим их отображением на экране.
Объект тестирования	Функциональность 1.1 модуля ResultOfTrack.
Входные данные	Тренировка «2020-07-12».
Ожидаемый результат	Общее время тренировки и километраж отображаются в мин и км. На экране, в блоке основных беговых показателей отразилась следующая информация: <ul style="list-style-type: none"> <li>• Время забега = 8 мин</li> <li>• Километраж = 1,5 км</li> </ul>

**Таблица 3.3.4 Тест № А4**

№ теста	А4
Цель теста (описание)	Общий, положительный. Тест проверяет возможность получения и преобразование основных беговых параметров тренировки из базовой единицы измерения (сек -> мин, метры -> км) с последующим их отображением на экране.
Объект тестирования	Функциональность 1.1 модуля ResultOfTrack.
Входные данные	Тренировка «2020-06-12».

Ожидаемый результат	Общее время тренировки отображается в виде «мин:сек». На экране, в блоке основных беговых показателей отразилась следующая информация: <ul style="list-style-type: none"> <li>• Время забега = 10:10 мин</li> </ul>
---------------------	--

**Таблица 3.3.5 Тест № А5**

№ теста	А5
Цель теста (описание)	Общий отрицательный. Тест проверяет возможность получения и отображения основных беговых параметров тренировки в случае если все значения отсутствуют.
Объект тестирования	Функциональность 1.1 модуля ResultOfTrack.
Входные данные	Тренировка «2020-08-12».
Ожидаемый результат	В полях где значений нет должен быть проставлен «-». На экране, в блоке основных беговых показателей отразилась следующая информация: <ul style="list-style-type: none"> <li>• Время забега = -</li> <li>• Километраж = -</li> <li>• Средний темп = -</li> <li>• Общий спуск = -</li> <li>• Общий подъём = -</li> <li>• Максимальный темп = -</li> <li>• Минимальный темп = -</li> <li>• Максимальный подъём = -</li> <li>• Максимальный спуск = -</li> </ul>

**Таблица 3.3.6 Тест № А6**

№ теста	А6
Цель теста (описание)	Общий отрицательный. Тест проверяет отрисовку графика зависимости темпа от километража. В случае если данных нет, то график не отрисовывается.
Объект тестирования	Функциональность 1.2 модуля ResultOfTrack.
Входные данные	Тренировка «2020-08-12».
Ожидаемый результат	На экране просмотра результата тренировки график зависимости темпа от километража не отобразился.

**Таблица 3.3.7 Тест № А7**

№ теста	А7
Цель теста (описание)	Общий, положительный. Тест проверяет отрисовку графика зависимости темпа от километража.
Объект тестирования	Функциональность 1.2 модуля ResultOfTrack.

Входные данные	Тренировка «2020-07-12».
Ожидаемый результат	На экране просмотра результата тренировки отрисовался график зависимости темпа от километража состоящий из 3-ёх значений: <ul style="list-style-type: none"> <li>• 0 мин/км</li> <li>• 4,04 мин/км</li> <li>• 7,18 мин/км</li> </ul>

**Таблица 3.3.8 Тест № А8**

№ теста	А8
Цель теста (описание)	Общий, положительный. Тест проверяет отрисовку графика зависимости темпа от километража.
Объект тестирования	Функциональность 1.2 модуля ResultOfTrack.
Входные данные	Тренировка «2020-05-12».
Ожидаемый результат	На экране просмотра результата тренировки отрисовался график зависимости темпа от километража состоящий из 2-ух значений: <ul style="list-style-type: none"> <li>• 0 мин/км</li> <li>• 8,53 мин/км</li> </ul>

**Таблица 3.3.9 Тест № А9**

№ теста	А9
Цель теста (описание)	Общий отрицательный. Тест проверяет создание и отрисовку блока среднего темпа по километражу, в случае если за тренировку было пройдено < 1 км.
Объект тестирования	Функциональность 1.3 модуля ResultOfTrack.
Входные данные	Тренировка «2020-04-12».
Ожидаемый результат	На экране просмотра результата тренировки в области блока среднего темпа по километрам ничего не отрисовалось.

**Таблица 3.3.10 Тест № А10**

№ теста	А10
Цель теста (описание)	Общий, положительный. Тест проверяет создание и отрисовку блока среднего темпа по километражу в случае если за тренировку было пройдено целое кол-во километров.
Объект тестирования	Функциональность 1.3 модуля ResultOfTrack.
Входные данные	Тренировка «2020-06-12».
Ожидаемый результат	На экране просмотра результата тренировки в области блока среднего темпа по километрам отобразилось 2 значения по 2 км: <ul style="list-style-type: none"> <li>• 1 км = 2,15 мин/км</li> <li>• 2 км = 2,42 мин/км</li> </ul>

**Таблица 3.3.11 Тест № A11**

№ теста	A11
Цель теста (описание)	Общий, положительный. Тест проверяет создание и отрисовку блока среднего темпа по километражу, в случае если за тренировку было преодолено не целое кол-во км.
Объект тестирования	Функциональность 1.3 модуля ResultOfTrack.
Входные данные	Тренировка «2020-07-12».
Ожидаемый результат	На экране просмотра результата тренировки в области блока среднего темпа по километрам отобразилось 1 значение за 1 км: <ul style="list-style-type: none"> <li>• 1 км = 4,04 мин/км</li> </ul>

**Таблица 3.3.12 Тест № A12**

№ теста	A12
Цель теста (описание)	Общий, отрицательный. Тест проверяет возможность передачи трека на сервер, если трек уже существует в БД. При условии наличия интернет-соединения.
Объект тестирования	Функциональность 1.5 модуля ResultOfTrack.
Входные данные	Тренировка «2020-07-12».
Ожидаемый результат	На экране просмотра результата тренировки после нажатия кнопки «отправить трек» ничего не произойдет.

**Таблица 3.3.13 Тест № A13**

№ теста	A13
Цель теста (описание)	Общий, положительный. Тест проверяет возможность передачи трека на сервер, если трека нет в БД. При условии наличия интернет-соединения.
Объект тестирования	Функциональность 1.5 модуля ResultOfTrack.
Входные данные	Тренировка «2020-04-12».
Ожидаемый результат	На экране просмотра результата тренировки после нажатия кнопки «отправить трек» отобразится сообщение «Трек успешно передан».

**Таблица 3.3.14 Тест № A14**

№ теста	A14
Цель теста (описание)	Общий, отрицательный. Тест проверяет возможность передачи трека на сервер, если трека нет в БД. При условии отсутствия интернет-соединения.
Объект тестирования	Функциональность 1.5 модуля ResultOfTrack.
Входные данные	Тренировка «2020-04-12».

Ожидаемый результат	На экране просмотра результата тренировки после нажатия кнопки «отправить трек» отобразится сообщение «Отсутствует интернет-соединение».
---------------------	--

**Таблица 3.3.15 Тест № А15**

№ теста	А15
Цель теста (описание)	Общий, положительный. Тест проверяет возможность сохранения трека в памяти устройства с последующим его открытием для просмотра результатов. При условии включенной гео-локации.
Объект тестирования	Функциональность 5.3 модуля ServicegetGPS и ResultOfTrack.
Входные данные	-
Ожидаемый результат	После запуска тренировки с получением нескольких gps-координат и последующей остановкой записи трека, трек сохраняется и автоматически открывается страница просмотра результатов тренировки, где отображаются основные параметры тренировки.

**Таблица 3.3.16 Тест № А16**

№ теста	А16
Цель теста (описание)	Общий, отрицательный. Тест проверяет возможность сохранения трека в памяти устройства с последующим его открытием для просмотра результатов. При условии отключенной гео-локации.
Объект тестирования	Функциональность 5.3 модуля ServicegetGPS и ResultOfTrack.
Входные данные	-
Ожидаемый результат	После запуска тренировки и последующей остановкой записи трека, трек сохраняется и автоматически открывается страница просмотра результатов тренировки, где вместо основных параметров тренировки отображаются «-».

### 3.4 Нагрузочное тестирование

**Таблица 3.4.1 Тест № Н1.1**

№ теста	Н1.1
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции getMainParameters в условиях малого количества gps-координат (~ 100): основные параметры расположены в конце файла тренировки.
Объект тестирования	Public getMainParameters (String str)
Входные данные	Файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.2 Тест № Н1.2**

№ теста	Н1.2
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции <code>getMainParameters</code> в условиях небольшого количества <code>gps</code> -координат (~ 500): основные параметры расположены в конце файла тренировки.
Объект тестирования	<code>Public getMainParameters (String str)</code>
Входные данные	Файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.3 Тест № Н1.3**

№ теста	Н1.3
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции <code>getMainParameters</code> в условиях среднего количества <code>gps</code> -координат (~ 1000): основные параметры расположены в конце файла тренировки.
Объект тестирования	<code>Public getMainParameters (String str)</code>
Входные данные	Файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.4 Тест № Н1.4**

№ теста	Н1.4
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции <code>getMainParameters</code> в условиях большого количества <code>gps</code> -координат (~ 2000): основные параметры расположены в конце файла тренировки.
Объект тестирования	<code>Public getMainParameters (String str)</code>
Входные данные	Файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.5 Тест № Н2.1**

№ теста	Н2.1
---------	------



Цель теста (описание)	Общий, нагрузочный. Проверка работы функции getLine в условиях малого количества gps-координат (~ 100).
Объект тестирования	PolylineOptions getLine()
Входные данные	GPS-координаты файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.6 Тест № Н2.2**

№ теста	Н2.2
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции getLine в условиях небольшого количества gps-координат (~ 500).
Объект тестирования	PolylineOptions getLine()
Входные данные	GPS-координаты файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 2 секунды.

**Таблица 3.4.7 Тест № Н2.3**

№ теста	Н2.3
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции getLine в условиях среднего количества gps-координат (~ 1000).
Объект тестирования	PolylineOptions getLine()
Входные данные	GPS-координаты файла трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 3 секунд.

**Таблица 3.4.8 Тест № Н2.4**

№ теста	Н2.4
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции getLine в условиях большого количества gps-координат (~ 2000).
Объект тестирования	PolylineOptions getLine()
Входные данные	GPS-координаты файла трека «2020-03-12»

Ожидаемый результат	Время выполнения функции не превышает 4 секунд.
---------------------	---

**Таблица 3.4.9 Тест № Н3.1**

№ теста	Н3.1
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции createTempForKilometer в условиях малого количества gps-координат (~ 100).
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	Файл трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 1 секунды.

**Таблица 3.4.10 Тест № Н3.2**

№ теста	Н3.2
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции createTempForKilometer в условиях небольшого количества gps-координат (~ 500).
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	Файл трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 2 секунды.

**Таблица 3.4.11 Тест № Н3.3**

№ теста	Н3.3
Цель теста (описание)	Общий, нагрузочный. Проверка работы функции createTempForKilometer в условиях среднего количества gps-координат (~ 1000).
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	Файл трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 3 секунд.

**Таблица 3.4.12 Тест № Н3.4**

№ теста	Н3.4
---------	------

Цель теста (описание)	Общий, нагрузочный. Проверка работы функции createTempForKilometer в условиях большого количества gps-координат (~ 2000).
Объект тестирования	Public createTempforKilometer(JsonObject track)
Входные данные	Файл трека «2020-03-12»
Ожидаемый результат	Время выполнения функции не превышает 4 секунд.

### 3.5 Покрытие кода тестами

Расчёт тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$$T_{cov} = \frac{L_{tc}}{L_{code}} \times 100\%$$

Где:

$T_{cov}$  - тестовое покрытие;

$L_{tc}$  - количество строк кода, покрытых тестами;

$L_{code}$  - общее количество строк кода.

Тогда  $T_{cov} = (1975/4764) * 100\% = 41,5 \%$

### 3.6 Пример реализации тестов

Ниже приведены примеры реализации блочных тестов: Листинг 6 и Листинг 7.

```

public void testGetLine() {
    ResultOfTrack resultOfTrack = new ResultOfTrack();
    ArrayList<ArrayList> points = new ArrayList<ArrayList>(); // GPS
    points.add(new ArrayList<LatLng>()); //Трек
    points.get(0).add(new LatLng( v: 61.77080000000001, v1: 34.314715)); //GPS coordinates
    PolylineOptions line = new PolylineOptions(); //output line
    ArrayList<LatLng> x = points.get(0);
    line.add(x.get(0));
    line.color(16711680);
    line.width(4);
    assertEquals(line.getColor(), resultOfTrack.getLine(points.get(0), color_line: 1).getColor());
    assertEquals(line.getPoints(), resultOfTrack.getLine(points.get(0), color_line: 1).getPoints());
    assertEquals(line.getWidth(), resultOfTrack.getLine(points.get(0), color_line: 1).getWidth());

    line.color(65280);
    assertEquals(line.getColor(), resultOfTrack.getLine(points.get(0), color_line: -1).getColor());

    line.color(255);
    assertEquals(line.getColor(), resultOfTrack.getLine(points.get(0), color_line: 0).getColor());
}

```

Листинг 6. Пример блочного тестирования.

```

public void testGetLine_negative(){
    ResultOfTrack resultOfTrack = new ResultOfTrack();
    ArrayList<ArrayList> points = new ArrayList<ArrayList>(); // GPS
    points.add(new ArrayList<LatLng>()); //Трек
    points.get(0).add(new LatLng( v: 0, v1: 0)); //GPS coordinates
    PolylineOptions line = new PolylineOptions(); //output line
    ArrayList<LatLng> x = points.get(0);
    line.add(x.get(0));
    fail(resultOfTrack.getLine(points.get(0), color_line: 4).toString());
}

```

Листинг 7. Пример блочного тестирования.

## 4. Журнал тестирования

№ теста	Дата	Результат	Номер ошибки в журнале
Блочное тестирование			
Б1.1	13.12.2020	Пройден	—
Б1.2	13.12.2020	Пройден	—
Б2	13.12.2020	Пройден	—
Б3	13.12.2020	Пройден	—
Б4	13.12.2020	Пройден	—
Б5	13.12.2020	Пройден	—
Б6	13.12.2020	Пройден	—
Б7	13.12.2020	Пройден	—
Б8	13.12.2020	Пройден	—

Б9	13.12.2020	Пройден	—
Б10	13.12.2020	Пройден	—
Б11.1	13.12.2020	Пройден	—
Б11.2	13.12.2020	Пройден	—
Б13	13.12.2020	Не пройден	Отчёт об ошибке №1
Б14.1	13.12.2020	Не пройден	Отчёт об ошибке №2
Б14.2	13.12.2020	Не пройден	Отчёт об ошибке №3
Б15	13.12.2020	Пройден	—
Б16	13.12.2020	Пройден	—
Б17	13.12.2020	Пройден	—
Б18	13.12.2020	Пройден	—
Б19	13.12.2020	Пройден	—
Интеграционное тестирование			
И1	17.12.2020	Пройден	—
И2	17.12.2020	Пройден	—
И3	17.12.2020	Пройден	—
И4	17.12.2020	Пройден	—
И5	17.12.2020	Пройден	—
И6	17.12.2020	Пройден	—
Аттестационное тестирование			
А1	20.12.2020	Пройден	—
А2	20.12.2020	Пройден	—
А3	20.12.2020	Пройден	—
А4	20.12.2020	Пройден	—
А5	20.12.2020	Пройден	—
А6	20.12.2020	Пройден	—
А7	20.12.2020	Пройден	—
А8	20.12.2020	Пройден	—
А9	20.12.2020	Пройден	—
А10	20.12.2020	Пройден	—
А11	20.12.2020	Пройден	—
А12	20.12.2020	Пройден	—
А13	20.12.2020	Пройден	—
А14	20.12.2020	Не пройден	Отчёт об ошибке №4
А15	20.12.2020	Пройден	—
А16	20.12.2020	Пройден	—
Нагрузочное тестирование			
Н1.1	07.01.2021	Пройден (0,452174320458 s)	—
Н1.2	07.01.2021	Пройден (0,463759397297 s)	—
Н1.3	07.01.2021	Пройден (0,657362468375 s)	—
Н1.4	07.01.2021	Пройден (0,745274859577 s)	—
Н2.1	07.01.2021	Пройден (0,434757629205 s)	—
Н2.2	07.01.2021	Пройден (0,756384958475 s)	—
Н2.3	07.01.2021	Пройден (1,195737497575 s)	—
Н2.4	07.01.2021	Пройден (1,857574382386 s)	—
Н3.1	07.01.2021	Пройден (0,512334534454 s)	—
Н3.2	07.01.2021	Пройден (1,834658234576 s)	—
Н3.3	07.01.2021	Не пройден (3,467867895679 s)	Отчёт об ошибке №5
Н3.4	07.01.2021	Не пройден (5,013453456467 s)	Отчёт об ошибке №6

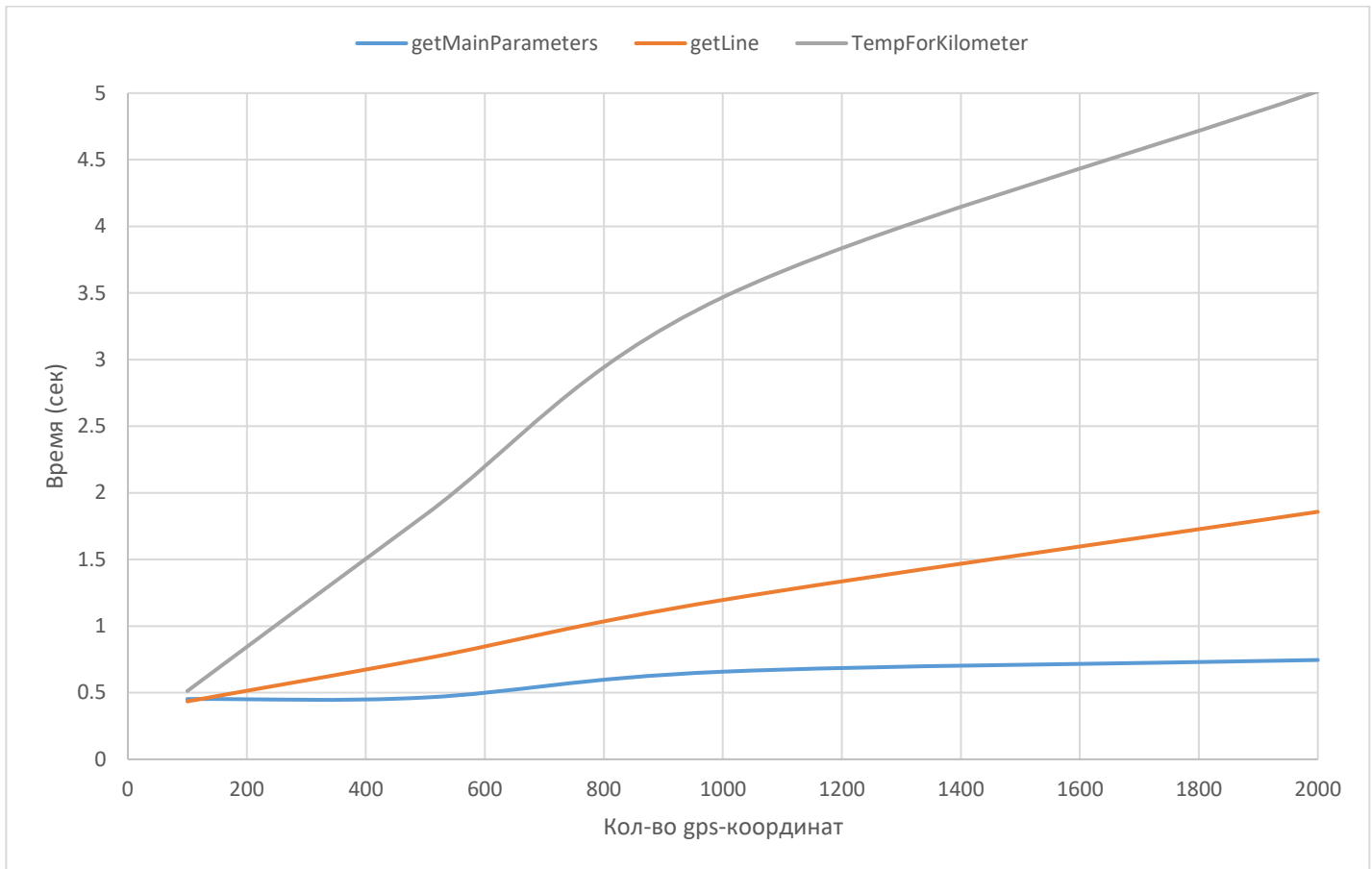


Рисунок 4: Результаты проведения нагрузочного тестирования.

## 5. Журнал найденных ошибок

№ отчёта об ошибке	1
Дата составления отчёта	13.12.2020
Номер теста	Б13
Объект тестирования	Public createTempforKilometer(JsonObject track)
Ожидаемый результат	Null
Фактический результат	Array[5,31]
Воспроизводимость	Всегда
Приоритет	Некритичная ошибка
Вариант решения	Не рассчитывать средний темп для нецелых километров.
Статус	Решено

№ отчёта об ошибке	2
Дата составления отчёта	13.12.2020
Номер теста	Б14.1
Объект тестирования	Public createTempforKilometer(JsonObject track)
Ожидаемый результат	Array[3,56]
Фактический результат	Array[3,56, 4,12]
Воспроизводимость	Всегда
Приоритет	Некритичная ошибка
Вариант решения	Не рассчитывать средний темп для нецелых километров.
Статус	Решено

№ отчёта об ошибке	3
Дата составления отчёта	13.12.2020

Номер теста	Б14.2
Объект тестирования	Public createTempforKilometer(JsonObject track)
Ожидаемый результат	Array[3,56, 10,17]
Фактический результат	Array[3,56, 10,17, 6,54]
Воспроизводимость	Всегда
Приоритет	Некритичная ошибка
Вариант решения	Не рассчитывать средний темп для нецелых километров.
Статус	Решено

№ отчёта об ошибке	4
Дата составления отчёта	20.12.2020
Номер теста	A14
Объект тестирования	Функциональность 1.5 модуля ResultOfTrack.
Ожидаемый результат	На экране просмотра результата тренировки после нажатия кнопки «отправить трек» отобразится сообщение «Отсутствует интернет-соединение».
Фактический результат	На экране просмотра результата тренировки после нажатия кнопки «отправить трек» не отобразилось сообщение «Отсутствует интернет-соединение».
Воспроизводимость	Всегда
Приоритет	Некритичная ошибка
Вариант решения	Добавить выводимое сообщение об ошибке на экран.
Статус	Решено

№ отчёта об ошибке	5
Дата составления отчёта	07.01.2021
Номер теста	Н3.3
Объект тестирования	Public createTempforKilometer(JsonObject track)
Ожидаемый результат	Время выполнения функции не превышает 3 секунд.
Фактический результат	Время выполнения функции составило 3,467867895679 сек.
Воспроизводимость	Всегда
Приоритет	Средний
Вариант решения	Добавить в БД отдельную таблицу, которая будет содержать средние темпы по км для каждой тренировки.
Статус	Не решено

№ отчёта об ошибке	6
Дата составления отчёта	07.01.2021
Номер теста	Н3.4
Объект тестирования	Public createTempforKilometer(JsonObject track)
Ожидаемый результат	Время выполнения функции не превышает 4 секунд.
Фактический результат	Время выполнения функции составило 5,013453456467 сек.
Воспроизводимость	Всегда
Приоритет	Средний
Вариант решения	Добавить в БД отдельную таблицу, которая будет содержать средние темпы по км для каждой тренировки.
Статус	Не решено

## 6. Результаты

Данное тестирование помогло работе проекта выявить ошибки в приложении. В ходе блочного, интеграционного, аттестационного и нагрузочного тестирования модулей приложения в рамках выполнения проекта НИР было выявлено 6 не критических ошибок.

Ошибки №1 – 4 были исправлены. Предположительно работа системы с технической точки зрения является работоспособной, все заявленные функции выполняются без ошибок. В ходе выполнения тестирования были выявлены недостатки, которые требуют рассмотрения в будущем.