

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Направление подготовки бакалавриата

09.03.04 - Программная инженерия

Отчет по Верификации ПО

ЦИФРОВОЙ АРБИТР

Выполнил:

студент 4 курса группы 22407

Е.Д. Топчий \_\_\_\_\_  
*подпись*

Руководитель:

доцент Кулаков К.А.

\_\_\_\_\_  
*подпись*

# Содержание

<b>1</b>	<b>Объект тестирования</b>	<b>5</b>
	<b>Описание предметной области</b>	<b>5</b>
1.1	Описание приложения . . . . .	5
1.2	Пользовательский интерфейс . . . . .	7
1.3	Архитектура приложения . . . . .	9
1.4	Тестируемые модули . . . . .	12
<b>2</b>	<b>Стратегия тестирования</b>	<b>13</b>
2.1	Тестовый стенд . . . . .	13
2.2	Тестируемый функционал . . . . .	14
2.2.1	Функция калибровки . . . . .	15
2.2.2	Функция определения времени прохождения заданного расстояния руками спортсмена . . . . .	16
2.2.3	Функция отображения статистики по результатам рабочего подхода для каждого повторения . . . . .	17
2.2.4	Функция воспроизведения звукового сопровождения . . . . .	18
2.3	Стратегия блочного тестирования . . . . .	19
2.4	Стратегия интеграционного тестирования . . . . .	20
2.5	Стратегия аттестационного тестирования . . . . .	21
2.6	Стратегия специального тестирования . . . . .	22
2.7	Критерии прохождения тестирования . . . . .	23
<b>3</b>	<b>Детальный план тестирования</b>	<b>24</b>
3.1	Блочное тестирование . . . . .	24
3.1.1	Функция калибровки . . . . .	24
3.1.2	Функция определения времени прохождения расстояния . . . . .	25
3.1.3	Функция отображения статистики . . . . .	25
3.1.4	Функция воспроизведения звукового сопровождения . . . . .	25
3.2	Описание интеграционных тестов . . . . .	27
3.2.1	Тест И1 [позитивный] (Интеграция: калибровка -> БД) . . . . .	27
3.2.2	Тест И2 [негативный] (Интеграция: калибровка -> БД) . . . . .	27
3.2.3	Тест И3 [позитивный] (Интеграция: измерение времени -> БД) . . . . .	27

3.2.4	Тест И4 [негативный] (Интеграция: измерение времени -> БД) . . . . .	28
3.2.5	Тест И5 [позитивный] (Интеграция: БД -> отображение детализации подхода) . . . . .	28
3.2.6	Тест И6 [негативный] (Интеграция: БД -> отображение детализации подхода) . . . . .	29
3.2.7	Тест И7 [позитивный] (Интеграция: калибровка -> отображение ре- зультатов) . . . . .	29
3.2.8	Тест И8 [негативный] (Интеграция: калибровка -> отображение ре- зультатов) . . . . .	29
3.2.9	Тест И9 [позитивный] (Интеграция: измерение времени -> отображе- ние результатов) . . . . .	30
3.2.10	Тест И10 [негативный] (Интеграция: измерение времени -> отобра- жение результатов) . . . . .	30
3.2.11	Тест И11 [позитивный] (Интеграция: БД -> калибровка) . . . . .	30
3.2.12	Тест И12 [негативный] (Интеграция: БД -> калибровка) . . . . .	31
3.3	Аттестационное тестирование . . . . .	32
3.3.1	Тест А1 [позитивный] (ФТ 1 Калибровка углов) . . . . .	32
3.3.2	Тест А2 [негативный] (ФТ 1 Калибровка углов при отсутствии фиксации) . . . . .	32
3.3.3	Тест А3 [позитивный] (ФТ 1 Точность измерения калибровочной ам- плитуды) . . . . .	32
3.3.4	Тест А4 [позитивный] (ФТ 2 Измерение времени выполнения подхода) . . . . .	33
3.3.5	Тест А5 [негативный] (ФТ 2 Измерение времени при недостижении расстояния) . . . . .	33
3.3.6	Тест А6 [позитивный] (ФТ 2 Точность измерения времени выполне- ния подхода) . . . . .	33
3.3.7	Тест А7 [позитивный] (ФТ 3 Отображение информации о повторениях) . . . . .	33
3.3.8	Тест А8 [негативный] (ФТ 3 Отсутствие информации о повторениях) . . . . .	34
3.3.9	Тест А9 [позитивный] (ФТ 3 Корректность отображения информации о повторениях) . . . . .	34
3.3.10	Тест А10 [позитивный] (ФТ 3 Адаптивность отображения информа- ции о повторениях) . . . . .	34
3.3.11	Тест А11 [позитивный] (ФТ 4 Звуковое сопровождение действий) . . . . .	35
3.3.12	Тест А12 [негативный] (ФТ 4 Отсутствие звукового сопровождения) . . . . .	35

3.3.13	Тест А13 [позитивный] (ФТ 4 Соответствие звукового сопровождения действиям) . . . . .	35
3.4	Специальное тестирование . . . . .	36
3.4.1	Тест С1 [позитивный] (Точность калибровки при различных условиях)	36
3.4.2	Тест С2 [позитивный] (Точность измерения времени при различных скоростях) . . . . .	36
3.4.3	Тест С3 [негативный] (Некорректное измерение времени при сбоях) .	36
3.4.4	Тест С4 [негативный] (Некорректное отображение информации при перегрузке системы) . . . . .	37
3.4.5	Тест С5 [позитивный] (Синхронизация звукового сопровождения с действиями) . . . . .	37
3.5	Пример кода блочного теста Б1 - Проверка фиксирования начального положения . . . . .	38
3.6	Пример кода интеграционного теста И5 - Интеграция: БД -> отображение детализации подхода . . . . .	39
3.7	Журнал тестирования . . . . .	40
3.8	Журнал найденных ошибок . . . . .	44
<b>4</b>	<b>Покрытие кода тестами</b>	<b>46</b>
4.1	Покрытие тестируемых функций . . . . .	46
4.2	Инструменты измерения покрытия . . . . .	46
4.3	Заключение . . . . .	46
<b>5</b>	<b>Заключение</b>	<b>47</b>

# 1 Объект тестирования

## 1.1 Описание приложения

Традиционно, соревнования по пауэрлифтингу проводятся без цифровых систем, отвечающих бы за соблюдение правильности выполнения упражнения спортсменами. Цифровизация спортивного тренировочного оборудования позволяет использовать приложения для распознавания движения человека при выполнении упражнений в тренажерном зале. Поскольку оценка участников осуществляется судьями, возникают следующие проблемы:

- Ошибки при фиксации правильности выполнения подхода.
- Большое количество судей для проведения соревнований.
- Ошибки при протоколировании из-за внесения результатов вручную.

Для решения данных проблем было разработано приложение цифровой судейской системы "Цифровой арбитр" предназначенное для анализа правильности выполнения спортсменом упражнения на тренажерах MB Barbell. Система снимает показания встроенного в смартфон акселерометра в реальном времени и оценивает амплитуду движения рычага при выполнении упражнения. Приложение позволяет фиксировать начальное и конечное положения рук спортсмена с расчетом амплитуды движения рук во время выполнения упражнения. Перед выполнением рабочего подхода выполняется калибровка, состоящая из произвольного количества повторений на тренажере с пустым весом. Калибровка определяет крайние положения рук при выполнении упражнения на данном тренажере для данного спортсмена. Все результаты калибровочных измерений используются для дальнейшего расчета среднего калибровочного значения амплитуды движения рук спортсмена. Таким образом, приложение "Цифровой арбитр" обладает следующими функциями:

1. **Функция индивидуальной калибровки под каждого спортсмена.**
2. Фиксация правильности выполнения упражнения без участия судьи.
3. Подсчет пройденного расстояния при выполнении упражнения.
4. **Детализация подхода в терминах пройденного расстояния, затраченного времени и количества повторений.**
5. Протоколирование всех выполненных подходов.

6. **Функция определения тренажера по QR коду.**
7. **Звуковое сопровождение при работе на тренажере.**
8. **Функции определения времени прохождения заданного расстояния руками спортсмена.**

Только выделенный жирным функционал в дальнейшем будет тестироваться.

## 1.2 Пользовательский интерфейс

Структура перехода по элементам пользовательского интерфейса, представлена на диаграмме переходов:

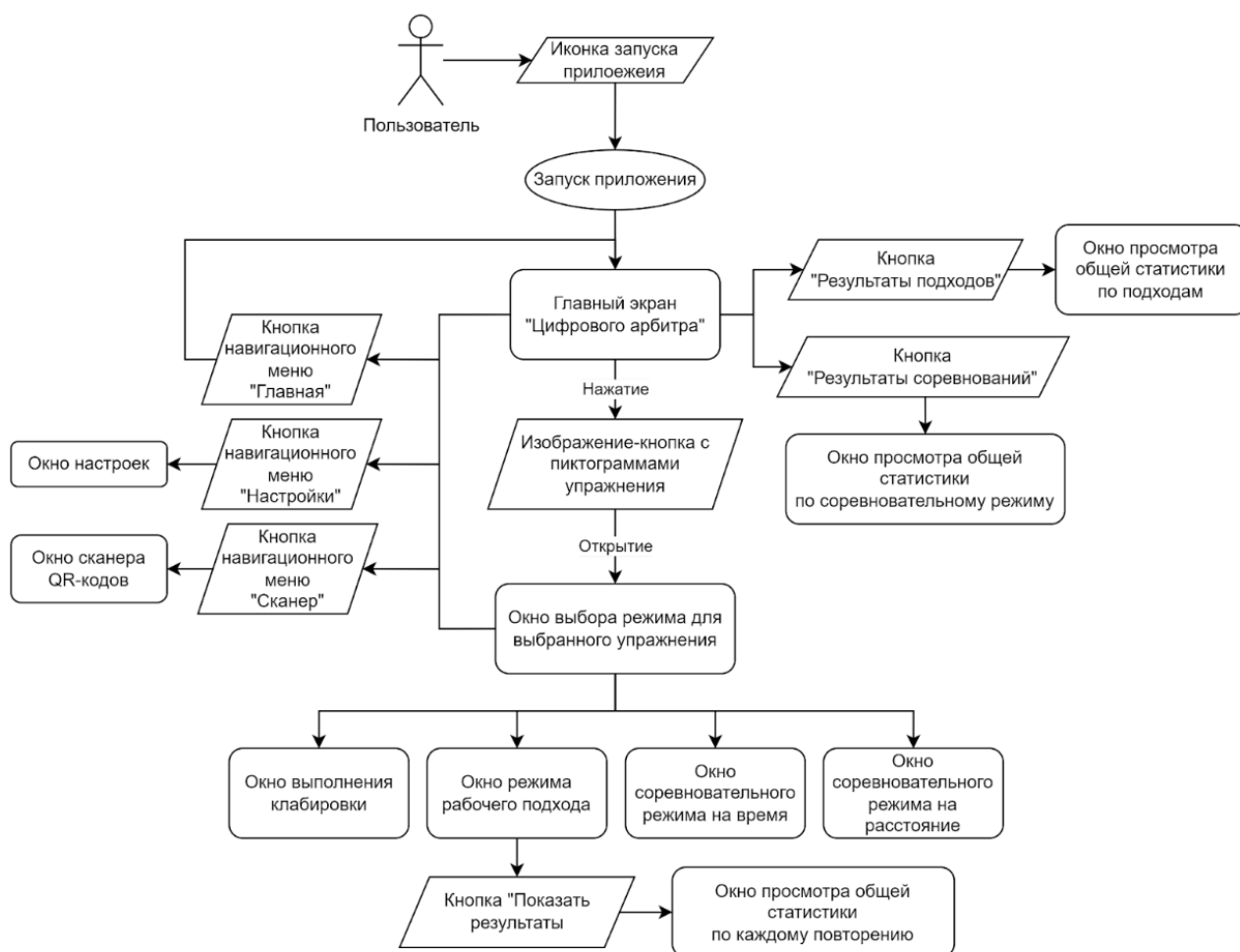


Рис. 1: Диаграмма переходов интерфейса.

Построение пользовательского интерфейса основывается на следующих принципах:

1. *Наглядность.* Пользователь должен интуитивно понимать назначение каждого элемента интерфейса и методику взаимодействия с ним. Кнопки подразумевают однократные нажатия, текстовые поля ввода предполагают ввод текста, согласно соответствующей подсказки (числа или текст);
2. *Функциональность.* Пользовательский интерфейс и его элементы должны осуществлять функции, заложенные в нем разработчиками;
3. *Минимализм.* Задачей пользовательского интерфейса обеспечить максимальную функциональность приложения с использованием минимального количества элементов интерфейса в соответствующем дизайне.

Пример пользовательского интерфейса мобильного приложения:

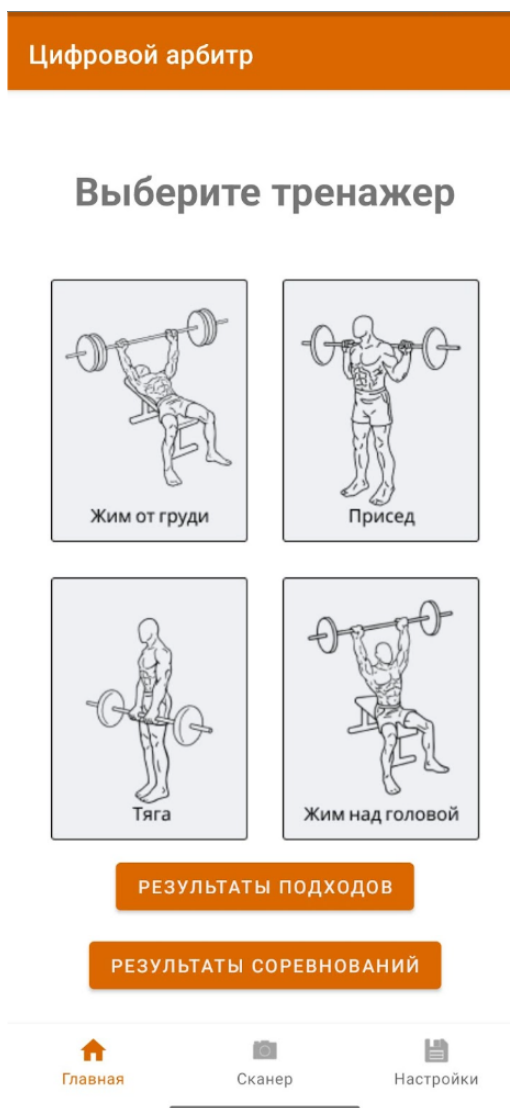


Рис. 2: Главный экран приложения.



### 1.3 Архитектура приложения

Общее представление архитектуры выглядит следующим образом:

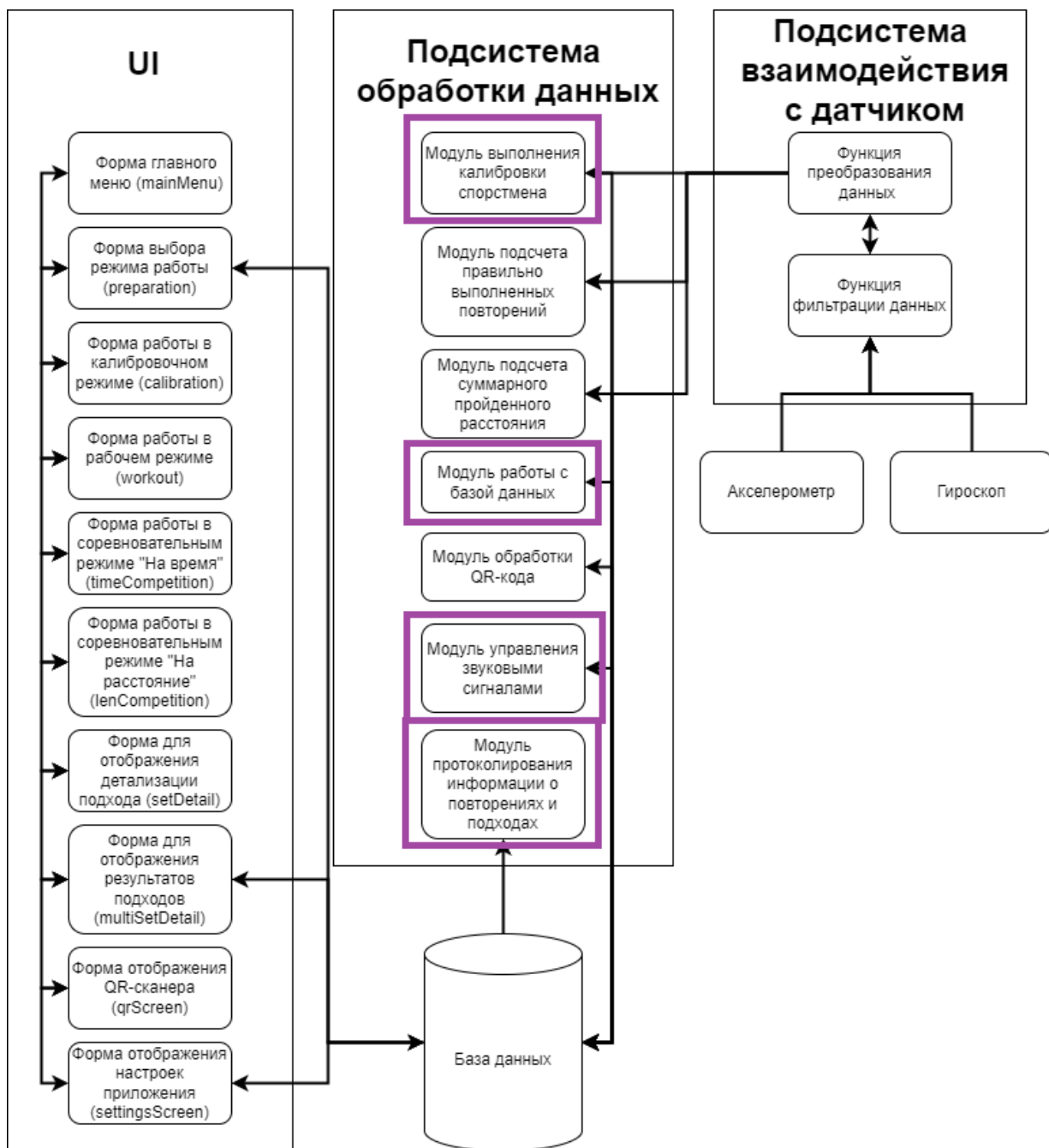


Рис. 3: Архитектура приложения.

Приложение включает в себя следующие модули:

1. **Модуль выполнения калибровки спортсмена.** После преобразования данных, получаемых с акселерометра, на вход функции калибровки непрерывно поступают значения угла  $p_i$ , где  $i$  - соответствует каждому последующему интервалу времени. Если получаемые подряд  $n$  значений  $p_i$  отличаются между собой не более чем на

константу , то считается, что пользователь зафиксировал тренажер в верхнем или нижнем положении.

При достижении пользователем указанного количества калибровочных повторений, вычисляются калибровочные значения путем усреднения полученных углов для нижнего и верхнего положений. Полученные углы для нижнего и верхнего положений заносятся в базу данных.

- 2. Модуль подсчета правильно выполненных повторений.** Из базы данных извлекаются калибровочные значения углов *upper* (верхнее положение) и *lower* (нижнее положение) После преобразования значений, получаемых с акселерометра, на вход функции рабочего подхода непрерывно поступают значения угла  $p_i$ . Если получаемые подряд  $k$  значений  $p_i$  достигают калибровочных значений на 95%, то повторение считается засчитанным.

Каждые правильно выполненные повторения суммируются, а также для них фиксируется время выполнения, а также пройденное руками атлета расстояние.

- 3. Модуль подсчета суммарного пройденного расстояния.** Для расчета пройденного расстояния руками спортсмена, необходимо измерить длину рычага от места установки датчика (смартфона) до оси вращения рамы тренажера. На основе полученных для каждого повторения углов  $\alpha_1$  (в крайней верхней точке) и  $\alpha_2$  (в крайней нижней точке) и длины рычага  $L$ , вычисляется путь  $S = (\pi * \beta) / (180 * L)$ , где  $\beta = |\alpha_1 - \alpha_2|$ ,  $\alpha_1$  и  $\alpha_2$  вычисляются как среднее значение среди всех углов, достигших 95% от калибровочного значения для каждого повторения.

- 4. Модуль работы с базой данных.** Для хранения параметров тренажеров (длина рычага), а также значения калибровочной амплитуды для каждого типа тренажеров используется локальная база данных с SQL-подобным синтаксисом.

- 5. Модуль обработки qr кода.** Для определения типа тренажера, используется система распознавания QR кода. После сканирования система определяет уникальный идентификатор тренажера и подгружает информацию для работы с ним.

- 6. Модуль управления звуковыми сигналами.** Работа на тренажере сопровождается настраиваемой системой звуковых оповещений, информирующих пользователя о начале и окончании подхода, засчитанных повторениях и процессе калибровки.

- 7. Модуль протоколирования информации о повторениях и подходах.** Ин-

формация о каждом выполненном подходе отображается в специальной таблице и характеризуется следующей информацией: количество повторений, пройденное расстояние, затраченное время. После выполнения подхода, атлет может посмотреть детализацию каждого повторения по времени и расстоянию.

## 1.4 Тестируемые модули

Будут тестироваться следующие модули:

1. Модуль выполнения калибровки спортсмена.
2. Модуль работы с базой данных.
3. Модуль управления звуковыми сигналами.
4. Модуль протоколирования информации о повторениях и подходах.

Остальная часть модулей тестируется другим разработчиком приложения, поскольку разработка ведется в команде. Тестируемый функционал разделен равномерно между разработчиками.

## 2 Стратегия тестирования

### 2.1 Тестовый стенд

Для экспериментального исследования и апробации разработанных алгоритмов использовался тренажер компании “МВ Barbell” типа “Жим лежа от груди” с регулируемым весом. Дополнительно для отладки и тестирования использовались тренажеры компании МВ Barbell рычагового типа для выполнения упражнений: “жим лежа”, “присед”, “становая тяга” и “жим над головой”, установленные на набережной Онежского Озера. На них с помощью фиксатора в определенном положении зафиксирован смартфон с предварительно установленным ПО для смартфона. Точка крепления фиксатора находится на заранее измеренном расстоянии от оси вращения рамы тренажера.



Рис. 4: Используемый тренажер для проведения экспериментов. Фиксатор для смартфона установлен с правой стороны на подвижной раме.

## 2.2 Тестируемый функционал

Необходимо протестировать следующие функции приложения:

1. **Функция калибровки** - функция, отвечающая за режим калибровки атлета, в котором он выполняет подход без веса с фиксацией рук(или других частей тела в зависимости от тренажера) в верхнем и нижнем положениях. Полученные данные используются в дальнейшем для подсчета правильно выполненных повторений.
2. **Функция определения времени прохождения заданного расстояния руками спортсмена** - функция, определяющая время, за которое спортсмен смог достичь необходимого расстояния руками при выполнении повторений. Для данного режима используются данные калибровки для определения правильности выполнения повторения.
3. **Функция отображения статистики по результатам рабочего подхода для каждого повторения** - функция, отвечающая за отображение статистики рабочего подхода. Формирует таблицу, где для каждого повторения отображается: порядковый номер; градус, на котором был зафиксирован тренажер в верхнем положении; градус, на котором был зафиксирован тренажер в нижнем положении; дистанция, пройденная руками атлета за данное повторение; время, затраченное атлетом на данное повторение.
4. **Функция воспроизведения звукового сопровождения** - функция, отвечающая за воспроизведение звукового сопровождения для действий спортсмена. Отвечает за воспроизведение инструкций для выбранных режимов, а также сопровождает действия спортсмена соответствующими сигналами во время выполнения упражнения: начало подхода, фиксация в верхнем/нижнем положениях, окончание повторения.

Тестирование остальной части функционала является задачей другого разработчика.

В связи с этим, в данной работе остальные функции тестироваться не будут.

### 2.2.1 Функция калибровки

```
fun calibrate(count: Int, pitch: Double)
...
return Double upper_position, Double lower_position
```

На вход алгоритма постоянно поступают значения углов, соответствующих положению подвижной рамы тренажера в текущий момент времени. Существует переменная  $i$ , которая является счетчиком удовлетворяющих условию углов, а также переменная  $CurCalib$ , которая содержит в себе сумму значений, удовлетворяющих условию углов. Угол удовлетворяет условиям, если его значение не отличается от значения предыдущего угла более, чем на  $\epsilon$ , иначе переменная  $i$  приравнивается к нулю. Если переменная  $i$  превысило некоторое значение  $n$ , то считается, что спортсмен зафиксировал тренажер в определенном положении, после чего вычисляется среднее значение по углам:  $CurCalib / i$ . Поочередно подсчитываются 2 калибровочных значения - для верхнего и нижнего положений. В зависимости от положения, среднее значение по углам прибавляется к переменным  $upCalib$  |  $downCalib$ . При каждом изменении данных переменных также срабатывают счетчики  $upCount$  |  $downCount$  соответственно. В зависимости от указанного пользователем кол-ва повторений, данные значения фиксируются до тех пор, пока необходимое кол-во повторений не будет выполнено. После чего вычисляются средние значения углов по всем повторениям для верхнего ( $upCalib / upCount$ ) и нижнего ( $downCalib / downCount$ ) положений, после чего эти данные заносятся в базу данных.

#### Входные данные:

- Угол положения подвижной рамы тренажера в данный момент (Тип данных: Double);
- Количество калибровочных повторений.(Тип данных: Int).

#### Выходные данные:

- Калибровочное значение для верхнего положения (Тип данных: Double);
- Калибровочное значение для нижнего положения (Тип данных: Double).

### 2.2.2 Функция определения времени прохождения заданного расстояния руками спортсмена

```
fun timeCompetition(pitch: Double, distance: Int, weight: Double)
...
return Double time, Long date
```

На вход функции поступает значение расстояния в метрах, которое задает пользователь. Пользователь выполняет упражнение, во время которого каждое измерение угла расположения рамы тренажера запоминается в переменную текущего угла *current\_pitch*, откуда затем значение используется для вычисления расстояния, пройденного за единицу измерения: пройденное руками расстояния определяется как длина дуги  $\alpha$ , пройденная руками, выраженная через пройденный угол  $\beta$ :

$$\alpha = (\pi * L) / 180 * \beta$$

Далее все полученные расстояния за каждую единицу измерения суммируются. Измерения времени останавливаются после того, как достигнуто заданное пользователем расстояние.

#### Входные данные:

- Угол положения подвижной рамы тренажера в данный момент (Тип данных: Double);
- Расстояние, которое необходимо пройти руками в метрах (Тип данных: Int);
- Вес (Тип данных: Double).

#### Выходные данные:

- Время в секундах (Тип данных: Double);
- Дата выполнения упражнения (Тип данных: Long).



### 2.2.3 Функция отображения статистики по результатам рабочего подхода для каждого повторения

```
fun showSetData(rep_num: Int, time: Double, upper_fix: Double, lower_fix: Double, rep_distance: Double)
...
return Form SetData
```

Функция отображает статистику текущего выполненного спортсменом рабочего подхода. В ней отображаются результаты каждого повторения. Для каждого повторения указываются номер повторения, затраченное время, углы нижней и верхней фиксации а также пройденное расстояние.

#### Входные данные:

- Номер повторения(Тип данных: Int);
- Время, затраченное спортсменом на данное повторение(Тип данных: Double);
- Угол, на котором спортсмен зафиксировал тренажер в нижнем положении(Тип данных: Double);
- Угол, на котором спортсмен зафиксировал тренажер в верхнем положении(Тип данных: Double);
- Расстояние, пройденное руками спортсмена за данное повторение(Тип данных: Double).

#### Выходные данные:

- Форма для отображения детализации подхода(setDetail).

#### 2.2.4 Функция воспроизведения звукового сопровождения

```
fun soundCompanion(signal: Int, volume: Int)
...
return Sound sound
```

Функция реализует озвучку в связи с выполняемыми пользователем действиями в виде ассистента, озвучивающего действия пользователя и подсказки при использовании различных окон приложения. Озвучка может функционировать в 2-х режимах:

- «Детальная озвучка» - все инструкции полностью проигрываются;
- «Короткая озвучка» - все инструкции сокращены и рассчитаны на опытных пользователей, которые уже работали с приложением в режиме «Детальной озвучки»

##### **Входные данные:**

- Действие пользователя - каждому типу действий соответствует свой идентификатор.
- Громкость звука

##### **Выходные данные:**

- Воспроизведение звукового сигнала.

## 2.3 Стратегия блочного тестирования

Тестирование будет проводиться путем вызова функций и сравнения результата вызова функции с ожидаемыми выходными данными.

Тестирование будет проводиться при помощи библиотеки модульного тестирования JUnit.

Блочное тестирование должно быть применено к следующим модулям:

- Модуль выполнения калибровки спортсмена;
- Модуль определения времени прохождения заданного расстояния руками спортсмена;
- Модуль управления звуковыми сигналами;
- Модуль протоколирования информации о повторениях и подходах.

## 2.4 Стратегия интеграционного тестирования

При интеграционном тестировании проверяется взаимодействие частей программы между собой, корректное получение исходных данных и отправка результатов. Модули вызываются последовательно.

Интеграционное тестирование будет проходить с помощью Android-фреймворка Espresso.

Тестированию подлежат следующие взаимодействия модулей:

- Модуль выполнения калибровки спортсмена -> база данных;
- Модуль определения времени прохождения заданного расстояния руками спортсмена -> база данных
- База данных -> модуль определения времени прохождения заданного расстояния руками спортсмена
- База данных -> модуль отображения детализации подхода.

## 2.5 Стратегия аттестационного тестирования

Аттестационное тестирование будет проводиться вручную согласно сценариям тестирования. Необходимо проверить следующий функционал:

1. При фиксации пользователем тренажера должны фиксироваться калибровочные углы, на основе измерения которых должна рассчитываться калибровочная амплитуда;
2. При выполнении спортсменом подхода на время для заданного расстояния должно фиксироваться время при достижении спортсменом необходимого расстояния;
3. После выполнения подхода информация о каждом повторении должна отображаться в соответствующей форме;
4. Действия спортсмена должны сопровождаться звуковыми сигналами или инструкциями

## 2.6 Стратегия специального тестирования

При специальном тестировании проверяется реакция приложения на следующие нестандартные условия:

- **Реакция на внешние вибрации** - под внешними вибрациями подразумевается внешнее воздействие на тренажер, которое приводит к возникновению вибраций. Например, если атлет сбросит тренажер, возникнет удар об упоры, что вызовет вибрации, что не должно сказываться на результатах спортсмена.
- **Реакция на изменение скорости выполнения** - так как определение положения рук атлета определяется с помощью акселерометра, измеряющего ускорения, необходимо проверить, влияет ли изменение скорости выполнения упражнения на точность определения положения рук атлета.
- **Реакция на нагрузку системы** - проверка реакции системы на нагрузку, система должна выдавать идентичные результаты при максимальной нагрузке на систему.

## 2.7 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.

## 3 Детальный план тестирования

### 3.1 Блочное тестирование

#### 3.1.1 Функция калибровки

1. **Тест Б1 [позитивный]:** Проверка фиксирования начального положения
  - *Описание:* Проверить фиксирование начального положения тренажера.
  - *Объект тестирования:* Calibration.kt, fun calibrate
  - *Входные данные:* 45 градусов.
  - *Ожидаемый результат:* upper\_position = 45.
2. **Тест Б2 [позитивный]:** Проверка фиксирования конечного положения
  - *Описание:* Проверить фиксирование конечного положения тренажера.
  - *Объект тестирования:* Calibration.kt, fun calibrate
  - *Входные данные:* 25 градусов.
  - *Ожидаемый результат:* lower\_position = 25.
3. **Тест Б3 [негативный]:** Передача данных в режим калибровки
  - *Описание:* Проверить реакцию системы на пустые данные калибровки.
  - *Объект тестирования:* Calibration.kt, fun calibrate
  - *Входные данные:* Пустая строка в имени пользователя.
  - *Ожидаемый результат:* error "Некорректный формат имени пользователя".
4. **Тест Б4 [позитивный]:** Повторная калибровка
  - *Описание:* Проверить возможность повторной калибровки после начальной настройки.
  - *Объект тестирования:* Calibration.kt, fun calibrate
  - *Входные данные:* show Calibration.
  - *Ожидаемый результат:* Система позволяет выполнить повторную калибровку.



### 3.1.2 Функция определения времени прохождения расстояния

#### 1. Тест Б5 [позитивный]: Точность измерения времени

- *Описание:* Проверить точность измерения времени прохождения заданного расстояния.
- *Объект тестирования:* TimeCompetition.kt, fun timeCompetition
- *Входные данные:* pitch = [23.4, 47.3, 24.7, 48.1] distance = 1, weight = 50.
- *Ожидаемый результат:* time = 3.146

#### 2. Тест Б6 [негативный]: Негативный тест на измерение времени

- *Описание:* Проверить реакцию системы на отсутствие движения.
- *Объект тестирования:* TimeCompetition.kt, fun timeCompetition
- *Входные данные:* pitch = [] distance = 1, weight = 50.
- *Ожидаемый результат:* error "Автоматическое отключение подхода после длительного бездействия".

### 3.1.3 Функция отображения статистики

#### 1. Тест Б7 [позитивный]: Отображение статистики по подходам

- *Описание:* Проверить отображение статистики по каждому рабочему подходу.
- *Объект тестирования:* ShowData.kt, fun showSetData
- *Входные данные:* rep\_num = [1,2,3,4,5], time = [1.64, 1.71, 1.83, 1.74, 1.63], upper\_fix = [47.99,47.10,47.99,47.06,47.55], lower\_fix = [23.65,23.49,23.74,23.66,23.88], rep\_distance = [0.780,0.753,0.770,0.755,0.764].
- *Ожидаемый результат:* show showSetData.

### 3.1.4 Функция воспроизведения звукового сопровождения

#### 1. Тест Б8 [позитивный]: Корректность звукового сигнала при начале калибровки

- *Описание:* Проверить, что звуковой сигнал при начале калибровки воспроизводится корректно.
- *Объект тестирования:* SoundCompanion.kt, fun soundCompanion
- *Входные данные:* signal = 1.

- *Ожидаемый результат:* Sound calibration\_start.

2. **Тест Б9 [позитивный]:** Звуковое сопровождение при ошибке

- *Описание:* Проверить воспроизведение звукового сигнала при возникновении ошибки.
- *Объект тестирования:* SoundCompanion.kt, fun soundCompanion
- *Входные данные:* signal = -1
- *Ожидаемый результат:* Sound error\_inactivity.

3. **Тест Б10 [позитивный]:** Настройка громкости звукового сопровождения

- *Описание:* Проверить возможность настройки громкости звукового сопровождения.
- *Объект тестирования:* SoundCompanion.kt, fun soundCompanion
- *Входные данные:* volume = 70.
- *Ожидаемый результат:* Sound test\_volume

## 3.2 Описание интеграционных тестов

### 3.2.1 Тест И1 [позитивный] (Интеграция: калибровка -> БД)

- *Описание:* Проверка сохранения данных калибровки в базе данных.
- *Объект тестирования:* Calibration.kt, DB.kt
- *Входные данные:* upper\_position = 45, lower\_position = 25
- *Передаваемые данные:* upper = 45, lower = 25, name = Евгений, exersice = benchPress.
- *Ожидаемый результат:* Запись калибровки в БД успешно создана: upper = 45, lower = 25, name = Евгений, exersice = benchPress.

### 3.2.2 Тест И2 [негативный] (Интеграция: калибровка -> БД)

- *Описание:* Проверка обработки ошибок при передаче некорректных данных калибровки в базу данных.
- *Объект тестирования:* Calibration.kt, DB.kt
- *Входные данные:* upper\_position = 45, lower\_position = ,
- *Передаваемые данные:* upper = 45, lower = , name = Евгений, exersice = benchPress.
- *Ожидаемый результат:* error "Некорректные данные при добавлении в базу данных".

### 3.2.3 Тест И3 [позитивный] (Интеграция: измерение времени -> БД)

- *Описание:* Проверка передачи и сохранения данных о времени выполнения упражнения в базе данных.
- *Объект тестирования:* TimeCompetition.kt, DB.kt
- *Входные данные:* pitch =[23.4, 47.3, 24.7, 48.1], distance = 1, weight = 50.
- *Передаваемые данные:* time = 3.146
- *Ожидаемый результат:* Запись в БД успешно сохранена: time = 3.146, pitch =[23.4, 47.3, 24.7, 48.1], distance = 1, weight = 50.

### 3.2.4 Тест И4 [негативный] (Интеграция: измерение времени -> БД)

- *Описание:* Проверка обработки ошибок при передаче некорректных данных о времени выполнения упражнения в базу данных.
- *Объект тестирования:* TimeCompetition.kt, DB.kt
- *Входные данные:* pitch = [23.4, 47.3, 24.7, 48.1] distance = 1, weight = 50.
- *Передаваемые данные:* time = 0.144
- *Ожидаемый результат:* error "Слишком короткое время выполнения"

### 3.2.5 Тест И5 [позитивный] (Интеграция: БД -> отображение детализации подхода)

- *Описание:* Проверка отображения данных из базы данных в модуле детализации подхода.
- *Объект тестирования:* DB.kt, ShowData.kt
- *Входные данные:* show SetData.
- *Передаваемые данные:* rep\_num = [1,2,3,4,5], time = [1.64, 1.71, 1.83, 1.74, 1.63], upper\_fix = [47.99,47.10,47.99,47.06,47.55], lower\_fix = [23.65,23.49,23.74,23.66,23.88], rep\_distance = [0.780,0.753,0.770,0.755,0.764].
- *Ожидаемый результат:* отображение окна детализации следующими данными:  
Повторение - 1, время - 1.64с, верхний угол - 47.99, нижний угол - 23.65, дистанция - 0.780м  
Повторение - 2, время - 1.71с, верхний угол - 47.10, нижний угол - 23.49, дистанция - 0.753м  
Повторение - 3, время - 1.83с, верхний угол - 47.99, нижний угол - 23.74, дистанция - 0.770м  
Повторение - 4, время - 1.74с, верхний угол - 47.06, нижний угол - 23.66, дистанция - 0.755м  
Повторение - 5, время - 1.63с, верхний угол - 47.55, нижний угол - 23.88, дистанция - 0.764м

### 3.2.6 Тест И6 [негативный] (Интеграция: БД -> отображение детализации подхода)

- *Описание:* Проверка обработки ошибок при некорректных данных в базе данных для модуля детализации подхода.
- *Объект тестирования:* DB.kt, ShowData.kt
- *Входные данные:* show SetData.
- *Передаваемые данные:* rep\_num = [1,2,3,4,5], time = [1.64, 1.71, 1.83, 1.74, 1.63], upper\_fix = [47.99,47.10,47.99,47.06], lower\_fix = [23.65,23.49,23.74,23.66,23.88], rep\_distance = [0.780,0.753,0.770,0.755,0.764].
- *Ожидаемый результат:* error "Данные некорректны или неполны"

### 3.2.7 Тест И7 [позитивный] (Интеграция: калибровка -> отображение результатов)

- *Описание:* Проверка корректности отображения результатов калибровки в пользовательском интерфейсе.
- *Объект тестирования:* Calibration.kt, ShowData.kt
- *Входные данные:* upper\_position = 45, lower\_position = 25
- *Передаваемые данные:* upper = 45, lower = 25, name = Евгений, exercise = benchPress.
- *Ожидаемый результат:* Отображение результатов калибровки со следующими данными:  
Верхний угол - 45  
Нижний угол - 25

### 3.2.8 Тест И8 [негативный] (Интеграция: калибровка -> отображение результатов)

- *Описание:* Проверка обработки ошибок при некорректных данных калибровки для отображения в пользовательском интерфейсе.
- *Объект тестирования:* Calibration.kt, ShowData.kt
- *Входные данные:* upper\_position = 45, lower\_position =

- *Передаваемые данные:* upper = 45, lower =, name = Евгений, exercise = benchPress.
- *Ожидаемый результат:* error "Данные калибровки некорректны".

### 3.2.9 Тест И9 [положительный] (Интеграция: измерение времени -> отображение результатов)

- *Описание:* Проверка отображения времени выполнения упражнения в пользовательском интерфейсе.
- *Объект тестирования:* TimeCompetition.kt, ShowData.kt
- *Входные данные:* pitch = [23.4, 47.3, 24.7, 48.1], distance = 1, weight = 50.
- *Передаваемые данные:* time = 3.146
- *Ожидаемый результат:* Отображения результатов со следующими данными:  
Время выполнения упражнения - 3.146с

### 3.2.10 Тест И10 [отрицательный] (Интеграция: измерение времени -> отображение результатов)

- *Описание:* Проверка обработки ошибок для времени выполнения для отображения в пользовательском интерфейсе.
- *Объект тестирования:* TimeCompetition.kt, ShowData.kt
- *Входные данные:* pitch = [], distance = 1, weight = 50.
- *Передаваемые данные:* time =
- *Ожидаемый результат:* error "Данные некорректны или неполны".

### 3.2.11 Тест И11 [положительный] (Интеграция: БД -> калибровка)

- *Описание:* Проверка извлечения данных калибровки из базы данных.
- *Объект тестирования:* Calibration.kt, DB.kt
- *Входные данные:* Запрос на извлечение результатов калибровки.
- *Передаваемые данные:* upper = 45, lower = 25, name = Евгений, exercise = benchPress.

- *Ожидаемый результат:* Отображение результатов калибровки со следующими данными:

Верхний угол - 45

Нижний угол - 25

### 3.2.12 Тест И12 [негативный] (Интеграция: БД -> калибровка)

- *Описание:* Проверка обработки ошибок данных калибровки из базы данных.
- *Объект тестирования:* Calibration.kt, DB.kt
- *Входные данные:* Запрос на извлечение результатов калибровки.
- *Передаваемые данные:* upper = 45, lower = , name = Евгений, exercise = benchPress.
- *Ожидаемый результат:* error "Данные некорректны или неполны".

### **3.3 Аттестационное тестирование**

#### **3.3.1 Тест А1 [позитивный] (ФТ 1 Калибровка углов)**

- *Описание:* Проверка корректности фиксации калибровочных углов и расчета калибровочной амплитуды.
- *Начальное состояние системы:* Запущен режим калибровки
- *Процесс:* Пользователь фиксирует тренажер в различных положениях.
- *Ожидаемый результат:* Калибровочные углы фиксируются, и на их основе рассчитывается калибровочная амплитуда.

#### **3.3.2 Тест А2 [негативный] (ФТ 1 Калибровка углов при отсутствии фиксации)**

- *Описание:* Проверка обработки ошибок при отсутствии фиксации тренажера спортсменом.
- *Начальное состояние системы:* Запущен режим калибровки
- *Процесс:* Пользователь не фиксирует тренажер в верхнем или нижнем положении.
- *Ожидаемый результат:* Система выдает ошибку "Отсутствие фиксации тренажера"

#### **3.3.3 Тест А3 [позитивный] (ФТ 1 Точность измерения калибровочной амплитуды)**

- *Описание:* Проверка точности измерения калибровочной амплитуды.
- *Начальное состояние системы:* Запущен режим калибровки.
- *Процесс:* Пользователь выполняет калибровку с физическим ограничителем амплитуды (ограничитель не позволяет поднять или опустить тренажер за установленные рамки).
- *Ожидаемый результат:* Калибровочная амплитуда измеряется с высокой точностью.



### **3.3.4 Тест А4 [позитивный] (ФТ 2 Измерение времени выполнения подхода)**

- *Описание:* Проверка измерения времени при выполнении спортсменом подхода на заданное расстояние.
- *Начальное состояние системы:* Запущен соревновательный режим на время.
- *Процесс:* Спортсмен выполняет подход на время для заданного расстояния.
- *Ожидаемый результат:* Время фиксируется при достижении спортсменом необходимого расстояния.

### **3.3.5 Тест А5 [негативный] (ФТ 2 Измерение времени при недостижении расстояния)**

- *Описание:* Проверка обработки ситуации, когда спортсмен не достигает заданного расстояния во время подхода.
- *Начальное состояние системы:* Запущен соревновательный режим на время.
- *Процесс:* Спортсмен выполняет подход, но не достигает заданного расстояния.
- *Ожидаемый результат:* Система не фиксирует время или выдает предупреждение о недостижении расстояния.

### **3.3.6 Тест А6 [позитивный] (ФТ 2 Точность измерения времени выполнения подхода)**

- *Описание:* Проверка точности измерения времени выполнения подхода.
- *Начальное состояние системы:* Запущен соревновательный режим на время.
- *Процесс:* Спортсмен выполняет подход на время для заданного расстояния.
- *Ожидаемый результат:* Время выполнения подхода соответствует реальному времени выполнения.

### **3.3.7 Тест А7 [позитивный] (ФТ 3 Отображение информации о повторениях)**

- *Описание:* Проверка отображения информации о каждом повторении после выполнения подхода.
- *Начальное состояние системы:* Запущен режим рабочего подхода.

- *Процесс:* Спортсмен выполняет подход с несколькими повторениями.
- *Ожидаемый результат:* Информация о каждом повторении корректно отображается в соответствующей форме.

### **3.3.8 Тест А8 [негативный] (ФТ 3 Отсутствие информации о повторениях)**

- *Описание:* Проверка обработки ситуации, когда информация о повторениях не отображается.
- *Начальное состояние системы:* Запущен режим рабочего подхода.
- *Процесс:* Спортсмен выполняет подход с нулем повторений и завершает его досрочно.
- *Ожидаемый результат:* Система выдает ошибку или предупреждение об отсутствии информации о повторениях.

### **3.3.9 Тест А9 [позитивный] (ФТ 3 Корректность отображения информации о повторениях)**

- *Описание:* Проверка корректности отображения информации о повторениях после выполнения подхода.
- *Начальное состояние системы:* Запущен соревновательный режим на время.
- *Процесс:* Спортсмен выполняет подход с несколькими повторениями.
- *Ожидаемый результат:* Время выполнения и расстояние соответствуют реальным времени выполнения и расстоянию.

### **3.3.10 Тест А10 [позитивный] (ФТ 3 Адаптивность отображения информации о повторениях)**

- *Описание:* Проверка адаптивности интерфейса при отображении информации о различном количестве повторений.
- *Начальное состояние системы:* Запущен режим рабочего подхода.
- *Процесс:* Спортсмен выполняет 50 различных подходов.
- *Ожидаемый результат:* Информация о повторениях адаптивно отображается в интерфейсе независимо от их количества.

### **3.3.11 Тест А11 [позитивный] (ФТ 4 Звуковое сопровождение действий)**

- *Описание:* Проверка сопровождения действий спортсмена звуковыми сигналами или инструкциями.
- *Начальное состояние системы:* Запущен режим калибровки
- *Процесс:* Спортсмен поочередно фиксирует тренажер в верхнем и нижнем положениях
- *Ожидаемый результат:* Действия спортсмена сопровождаются соответствующими звуковыми сигналами или инструкциями.

### **3.3.12 Тест А12 [негативный] (ФТ 4 Отсутствие звукового сопровождения)**

- *Описание:* Проверка обработки ситуации, когда звуковое сопровождение не происходит.
- *Начальное состояние системы:* Запущен режим калибровки.
- *Процесс:* Спортсмен выполняет действия на тренажере, но громкость телефона убавлена до нуля.
- *Ожидаемый результат:* Система выдает ошибку или предупреждение об отсутствии звукового сопровождения.

### **3.3.13 Тест А13 [позитивный] (ФТ 4 Соответствие звукового сопровождения действиям)**

- *Описание:* Проверка соответствия звукового сопровождения действиям спортсмена.
- *Начальное состояние системы:* Запущен режим рабочего подхода.
- *Процесс:* Спортсмен поочередно фиксирует тренажер в нижнем и верхнем положении пять раз.
- *Ожидаемый результат:* Звуковое сопровождение сопровождает каждую фиксацию тренажера, а также издает сигнал, оповещающий о завершении подхода по достижению пяти повторений.

## 3.4 Специальное тестирование

### 3.4.1 Тест С1 [позитивный] (Точность калибровки при различных условиях)

- *Описание:* Проверка точности калибровки тренажера при различных условиях использования.
- *Начальное состояние системы:* Запущен режим калибровки.
- *Процесс:* Пользователь выполняет три различных калибровки с физическим ограничителем амплитуды(ограничитель не позволяет поднять или опустить тренажер за установленные рамки):
  1. калибровка выполняется в благоприятных условиях
  2. калибровка выполняется при слабых вибрациях тренажера
  3. калибровка выполняется при сильных вибрациях тренажера
- *Ожидаемый результат:* Результаты калибровки имеют незначительные отличия и соответствуют действительности.

### 3.4.2 Тест С2 [позитивный] (Точность измерения времени при различных скоростях)

- *Описание:* Проверка точности измерения времени выполнения подхода при различных скоростях выполнения упражнений.
- *Начальное состояние системы:* Запущен режим рабочего подхода.
- *Процесс:* Пользователь выполняет три различных рабочих подхода с физическим ограничителем амплитуды(ограничитель не позволяет поднять или опустить тренажер за установленные рамки):
  1. рабочий подход выполняется медленно
  2. рабочий подход выполняется с обычной скоростью
  3. рабочий подход выполняется максимально быстро
- *Ожидаемый результат:* Время выполнения подхода измеряется точно при любой скорости выполнения.

### 3.4.3 Тест С3 [негативный] (Некорректное измерение времени при сбоях)

- *Описание:* Проверка обработки ошибок при некорректном измерении времени из-за сбоев в системе.

- *Начальное состояние системы:* Запущен режим калибровки.
- *Процесс:* Во время выполнения подхода происходит сбой в системе измерения времени.
- *Ожидаемый результат:* Система выдает ошибку и не сохраняет некорректно измеренное время.

#### **3.4.4 Тест С4 [негативный] (Некорректное отображение информации при перегрузке системы)**

- *Описание:* Проверка обработки ошибок при некорректном отображении информации о повторениях из-за перегрузки системы.
- *Начальное состояние системы:* Запущен режим рабочего подхода.
- *Процесс:* Система перегружена, и информация о повторениях отображается некорректно.
- *Ожидаемый результат:* Система выдает ошибку или предупреждение о некорректном отображении информации.

#### **3.4.5 Тест С5 [позитивный] (Синхронизация звукового сопровождения с действиями)**

- *Описание:* Проверка синхронизации звукового сопровождения с действиями спортсмена.
- *Начальное состояние системы:* Запущен режим калибровки.
- *Процесс:* Пользователь выполняет калибровку с физическим ограничителем амплитуды (ограничитель не позволяет поднять или опустить тренажер за установленные рамки)
- *Ожидаемый результат:* Звуковое сопровождение синхронизировано с действиями пользователя.

### 3.5 Пример кода блочного теста Б1 - Проверка фиксирования начального положения

```
import org.junit.Assert.assertEquals
import org.junit.Test

class CalibrationTest {

    @Test
    fun testCalibrate() {
        // Создаем экземпляр класса Calibration
        val calibration = Calibration()

        // Вызываем метод calibrate с входными данными 45 градусов
        calibration.calibrate(45)

        // Проверяем, что upper_position установлено в 45
        assertEquals(45, calibration.upperPosition)
    }
}
```

### 3.6 Пример кода интеграционного теста И5 - Интеграция: БД -> отображение детализации подхода

```
import org.junit.Assert.assertEquals
import org.junit.Before
import org.junit.Test
import org.mockito.Mockito.mock
import org.mockito.Mockito.`when`

class IntegrationTest {

    private lateinit var calibration: Calibration
    private lateinit var showData: ShowData

    @Before
    fun setUp() {
        // Инициализация и мокирование объектов
        calibration = mock(Calibration::class.java)
        showData = ShowData(calibration)
    }

    @Test
    fun testCalibrationDataIntegration() {
        // Настройка моков
        `when`(calibration.upperPosition).thenReturn(45)
        `when`(calibration.lowerPosition).thenReturn(25)

        // Вызов метода для отображения данных
        val result = showData.displayCalibrationResults("Евгений", "benchPress")

        // Проверка результата
        assertEquals("Верхний угол - 45\nнижний угол - 25", result)
    }
}
```

### 3.7 Журнал тестирования

Номер теста	Дата проведения	Результат	Отчет об ошибке
Тест Б1	15.11.2023	Успешно	
Тест Б2	15.11.2023	Успешно	
Тест Б3	15.11.2023	Неудачно	Не выводится необходимая ошибка
Тест Б4	15.11.2023	Успешно	
Тест Б5	15.11.2023	Успешно	
Тест Б6	16.11.2023	Успешно	
Тест Б7	16.11.2023	Успешно	
Тест Б8	16.11.2023	Успешно	
Тест Б9	16.11.2023	Успешно	
Тест Б10	16.11.2023	Успешно	

Таблица 1: Блочное тестирование



Номер теста	Дата проведения	Результат	Отчет об ошибке
Тест И1	18.11.2023	Успешно	
Тест И2	18.11.2023	Неудачно	Не выводится необходимая ошибка
Тест И3	18.11.2023	Успешно	
Тест И4	18.11.2023	Успешно	
Тест И5	18.11.2023	Неудачно	Не отображается последнее выполненное повторение
Тест И6	18.11.2023	Успешно	
Тест И7	20.11.2023	Успешно	
Тест И8	20.11.2023	Успешно	
Тест И9	20.11.2023	Успешно	
Тест И10	20.11.2023	Успешно	
Тест И11	20.11.2023	Успешно	
Тест И12	20.11.2023	Успешно	

Таблица 2: Интеграционное тестирование

Номер теста	Дата проведения	Результат	Отчет об ошибке
Тест А1	24.11.2023	Успешно	
Тест А2	24.11.2023	Успешно	
Тест А3	24.11.2023	Успешно	
Тест А4	24.11.2023	Успешно	
Тест А5	24.11.2023	Успешно	
Тест А6	24.11.2023	Успешно	
Тест А7	27.11.2023	Успешно	
Тест А8	27.11.2023	Успешно	
Тест А9	27.11.2023	Успешно	
Тест А10	27.11.2023	Неудачно	На экране отображаются только первые 32 повторения
Тест А11	27.11.2023	Успешно	
Тест А12	27.11.2023	Успешно	
Тест А13	27.11.2023	Успешно	

Таблица 3: Аттестационное тестирование

<b>Номер теста</b>	<b>Дата проведения</b>	<b>Результат</b>	<b>Отчет об ошибке</b>
Тест С1	04.12.2023	Успешно	
Тест С2	04.12.2023	Успешно	
Тест С3	04.12.2023	Успешно	
Тест С4	04.12.2023	Успешно	
Тест С5	04.12.2023	Успешно	

Таблица 4: Специальное тестирование

### 3.8 Журнал найденных ошибок

<b>№ Отчета</b>	1
<b>Дата</b>	15.11.2023
<b>№ Теста</b>	БЗ
<b>Условия возникновения</b>	Попытка запустить режим калибровки при незаполненном поле "Имя пользователя"
<b>Ожидаемый Результат</b>	error "Некорректный формат имени пользователя"
<b>Фактический Результат</b>	ошибка не выводится
<b>Приоритет</b>	Некритичная

<b>№ Отчета</b>	2
<b>Дата</b>	18.11.2023
<b>№ Теста</b>	И2
<b>Условия возникновения</b>	Попытка передать данные некорректного формата в базу данных
<b>Ожидаемый Результат</b>	error "Некорректные данные при добавлении в базу данных"
<b>Фактический Результат</b>	Не выводится необходимая ошибка
<b>Приоритет</b>	Некритичная

<b>№ Отчета</b>	3
<b>Дата</b>	18.11.2023
<b>№ Теста</b>	И5
<b>Условия возникновения</b>	Атлет выполняет подход и нажимает кнопку "Детализация подхода"
<b>Ожидаемый Результат</b>	Отображаются все повторения в подходе
<b>Фактический Результат</b>	Не отображается последнее выполненное повторение
<b>Приоритет</b>	Некритичная

<b>№ Отчета</b>	4
<b>Дата</b>	27.11.2023
<b>№ Теста</b>	A10
<b>Условия возникновения</b>	Спортсмен выполнил более 32-х повторений за один подход
<b>Ожидаемый Результат</b>	Информация о повторениях адаптивно отображается в интерфейсе независимо от их количества
<b>Фактический Результат</b>	На экране отображаются только первые 32 повторения
<b>Приоритет</b>	Некритичная

## 4 Покрывтие кода тестами

В рамках процесса верификации ПО "Цифровой арбитр" особое внимание было уделено покрытию кода тестами. Целью данного этапа является обеспечение высокого уровня качества и надежности программного продукта путем тщательной проверки всех ключевых функций и модулей системы.

### 4.1 Покрывтие тестируемых функций

Тестирование охватывало следующие основные функции приложения:

- Функция индивидуальной калибровки под каждого спортсмена - **покрытие  $\approx 94\%$** .
- Детализация подхода в терминах пройденного расстояния, затраченного времени и количества повторений - **покрытие  $\approx 86\%$**
- Звуковое сопровождение при работе на тренажере - **покрытие  $\approx 82\%$** .
- Функции определения времени прохождения заданного расстояния руками спортсмена - **покрытие  $\approx 87\%$**  .

Для каждой из этих функций были разработаны и выполнены блочные, интеграционные и аттестационные тесты. Это позволило оценить корректность работы каждой функции в изоляции, а также в контексте взаимодействия с другими компонентами системы.

### 4.2 Инструменты измерения покрытия

Для измерения покрытия кода тестами использовался инструмент *JaCoCo*, который предоставляет детальный отчет о покрытии кода на уровне строк и ветвлений.

### 4.3 Заключение

Таким образом, покрытие кода тестами в проекте "Цифровой арбитр" было выполнено с высоким уровнем детализации и охватило все ключевые аспекты функциональности приложения. Общий уровень покрытия кода составил примерно  $87\%$ . Это обеспечивает уверенность в том, что приложение будет работать корректно и эффективно в реальных условиях эксплуатации.

## 5 Заключение

В ходе тестирования приложения "Цифровой арбитр" были проведены комплексные тесты, включая блочное, интеграционное, аттестационное и специальное тестирование. Основной целью тестирования было обеспечение надежности, эффективности и точности работы приложения в условиях реального использования.

### **Основные результаты тестирования:**

*Блочное тестирование:* Показало высокую степень надежности отдельных модулей приложения. Большинство тестов были успешно пройдены, что подтверждает корректность работы отдельных функций приложения.

*Интеграционное тестирование:* Выявило хорошую совместимость и взаимодействие между различными модулями приложения. Незначительные проблемы, выявленные в ходе тестирования, были оперативно исправлены.

*Аттестационное тестирование:* Подтвердило готовность приложения к эксплуатации. Приложение успешно прошло все критические тесты, что говорит о его надежности и эффективности в реальных условиях использования.

*Специальное тестирование:* Особое внимание было уделено проверке точности измерений и устойчивости приложения к различным условиям эксплуатации. Результаты тестирования подтвердили высокую точность и надежность приложения в разнообразных сценариях использования.

### **Общие выводы:**

Приложение "Цифровой арбитр" демонстрирует высокую степень надежности, точности и функциональности. Большинство тестов были успешно пройдены, а найденные ошибки были оперативно устранены. Это позволяет рекомендовать приложение к использованию в спортивных залах для повышения эффективности и точности оценки выполнения упражнений.

Таким образом, результаты тестирования подтверждают готовность приложения "Цифровой арбитр" к широкому внедрению и использованию в целях автоматизации процесса оценки выполнения спортивных упражнений.