**ФГБОУ ВО   
Уфимский университет науки и технологий**

**Кафедра ВМиК**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 90 |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

Программирование мобильного приложения, использующего сенсоры

**ОТЧЕТ**

**по лабораторной работе**

**по** Программированию  
мобильных приложений

(*наименование дисциплины*)

|  |
| --- |
| Лабораторная работа 1 |
| (обозначение документа) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  | Фамилия, И., О. | Подпись | Дата | Оценка |
| номер |  |
|  |  |
| Студент | | | Студент\_ФИО |  |  |  |
| Преподаватель | | | Преподаватель\_ФИО |  |  |  |
| Принял | | |  |  |  |  |

**Уфа 2024 г****.**

Содержание

[1 Цель работы 3](#_Toc182145781)

[2 Практическая часть 4](#_Toc182145782)

[2.1 Выбор приложения 4](#_Toc182145783)

[2.2 Разработка кода 4](#_Toc182145784)

[2.3 Примеры работы приложения 8](#_Toc182145785)

[3 Вывод 10](#_Toc182145786)

[4 Список литературы 11](#_Toc182145787)

# Цель работы

Целью работы является разработка мобильного приложения, которое будет использовать сенсоры.

# Практическая часть

## Выбор приложения

В ходе лабораторной работы нам необходимо реализовать мобильное приложение, использующее сенсоры, пусть это будет «уровень».

## Разработка кода

Будем разрабатывать приложение на языке программирования «Kotlin», так как разработка классических мобильных приложений ведется на ЯП «Java», а «Kotlin» – кроссплатформенный, статически типизированный, объектно-ориентированный язык программирования, работающий поверх «Java», но при этом его [«Kotlin»] синтаксис легче.

Для начала создадим новый проект «Empty Views Activity», как показано на рисунке 2.1 и рисунке 2.2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.1 – Создание проекта (выбор шаблона)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.2 – Создание проекта (выбор ЯП)

Далее переходим в «activity\_main.xml», где настроим интерфейс программы: добавим один «TextView» и три «LinearLayout» (рисунок 2.3). «TextView» необходим, чтобы выводить градусы отклонения, а «LinearLayout» нужен, чтобы показывать это отклонение графически.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.3 – Разработка интерфейса

Теперь реализуем код приложения в «MainActivity». Сначала создадим массивы, в которых будем хранить значения, передаваемые сенсорами «Accelerometer» и «Magnetic Field» (рисунок 2.4).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.4 – Создание массивов

Далее подключим сенсоры, как изображено на рисунке 2.5.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 2.5 – Подключение сенсоров

Затем передадим полученные сенсорами значения в созданные ранее массивы (рисунок 2.6).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 2.6 – Передача значений сенсоров в массивы

Наконец приведем полученные значения в градусную меру относительно горизонтальной поверхности. Цвет центрального «LinearLayout» будем изменять на зеленый, если получен «0», иначе – на красный, как приведено на рисунок 2.7.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.7 – Нормализация данных

В конце зарегистрируем слушателей, чтобы они могли получать данные от сенсоров в реальном времени (рисунок 2.8).



Рисунок 2.8 – Регистрация слушателей

## Примеры работы приложения

На рисунке 2.9 представлена ситуация, когда телефон расположен параллельно по отношению к горизонтальной поверхности, а на рисунке 2.10 – непараллельно.

Изображение выглядит как снимок экрана, текст, График

Автоматически созданное описание

Рисунок 2.9 – Пример параллельного расположения

Изображение выглядит как снимок экрана, текст, линия, Шрифт

Автоматически созданное описание

Рисунок 2.10 - Пример непараллельного расположения

# Вывод

В ходе лабораторной работы мы разработали мобильное приложение «уровень», которое использует сенсоры.

# Список литературы

1. https://ru.wikipedia.org/wiki/Kotlin
2. https://www.youtube.com/watch?v=H92xX4uiu4M
3. https://www.youtube.com/watch?v=BM6J6AhEl0Q