|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА — Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №10**

по дисциплине

«Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-33-21 | Привезенцев Д.А. |
| **Проверил:**  Ассистент | Новичков Д.Е. |

Москва 2023 г.

**СОДЕРЖАНИЕ**

[1. ЗАДАНИЕ 1 3](#_Toc136851026)

[1.1 Условие задания 3](#_Toc136851027)

[1.2 Решение задания 3](#_Toc136851028)

[2. ЗАДАНИЕ 2 4](#_Toc136851029)

[2.1 Условие задания 4](#_Toc136851032)

[2.2 Решение задания 4](#_Toc136851033)

[ВЫВОД 7](#_Toc136851034)

# ЗАДАНИЕ 1

## Условие задания

В разрабатываемом приложении необходимо реализовать многопоточную задачу, обоснованную предметной областью, основанную на классе ExecutorService.

## Решение задания

Для решения данной задачи был добавлен функционал, позволяющий выводить сообщение о работе в виде Toast уведомления. Вид приложения во время многопоточной работы представлен на Рисунке 1.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 1 — Сообщение о начале работы

# ЗАДАНИЕ 2



## Условие задания

В разрабатываемом приложении необходимо реализовать многопоточную задачу, обоснованную предметной областью, основанную на классе Worker.

## Решение задания

Для решения данной задачи был реализован класс MyWorker, реализующий многопоточную задачу, которая выводит состояние задачи с помощью Toast и по прошествии 10 секунд выводит результат. Итоговый вид приложения показан на Рисунках 2-4.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 2 — запуск задачи

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3 — выполнение задачи

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 4 — завершение задачи

# ВЫВОД

В данном проекте были решены две задачи, связанные с многопоточностью. Первая задача была решена с использованием класса ExecutorService. Был создан пул потоков, и при нажатии на кнопку запускалась задача в отдельном потоке, которая имитировала получение сообщения от сервера и выводила его в виде всплывающего сообщения. Вторая задача была реализована с помощью класса Worker из библиотеки androidx.work. Созданный класс выполнял фоновую задачу с задержкой и возвращал результат успешного или неуспешного выполнения. Оба решения позволяют выполнять длительные задачи в фоновом режиме, улучшая отзывчивость приложения и освобождая главный поток от нагрузки.