

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 — Программная инженерия

Отчет о проектной работе по курсу «Разработка приложений для мобильных ОС»

ПРИЛОЖЕНИЕ «TODO»

Выполнил:

Куусела Демид Александрович

студент 2 курса группы 22207

Д. А. Куусела \_\_\_\_\_  
*подпись*

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Требования к приложению</b>	<b>3</b>
<b>2 Проектирование приложения</b>	<b>3</b>
2.1 MainActivity.kt . . . . .	3
2.2 MainFragment.kt . . . . .	3
2.3 CustomAdapter.kt . . . . .	4
<b>3 Реализация приложения</b>	<b>4</b>
<b>Заключение</b>	<b>5</b>

# Введение

Цель проекта: разработать мобильное приложение для управления задач.

Задачи проекта:

1. Организовать хранение данных о задачах
2. Организовать хранение данных о задачах при выключенном приложении
3. Разработать интерфейс приложения
4. Разработать функции для загрузки информации в главный фрагмент

## 1 Требования к приложению

1. Возможность просматривать текущие задачи
2. Возможность добавлять новые задачи
3. Возможность удалять задачи

## 2 Проектирование приложения

### 2.1 MainActivity.kt

Главный Kotlin файл, создаёт и запускает Activity.

### 2.2 MainFragment.kt

Kotlin файл, подгружает данные с помощью функции `loadTasks`, навешивает "прослушку" на кнопку добавления новой задачи. Также в `MainFragment` есть несколько функций:

- `addTask` — преобразование содержимого поля ввода в строку, добавление новой задачи в список, очищение поля ввода;
- `saveTasks` — сохранение текущих задач в `SharedPreferences`;
- `loadTasks` — подгрузка данных из `SharedPreferences`.

## 2.3 CustomAdapter.kt

Kotlin файл, адаптер между моделью и отображением (view). Функции:

- addTask — добавление задачи в модель и вызов сохранения данных;
- deleteTask — удаление задачи из модели и вызов сохранения данных;
- getTasksList — возвращает список моделей текущих задач.

## 3 Реализация приложения

В итоге приложение содержит:

- 4 модуля
- 3 Kotlin класса с 6 функциями
- 136 строк Kotlin кода и 91 строка XML кода

## Заключение

В результате разработано приложение для просмотра, добавления и удаления текущих задач. Реализованы все запланированные функции.

Скриншот приложения:

