

Колледж космического машиностроения и технологий

**Отчёт по модульному экзамену**

**Тема: «Разработка мобильного приложения «Часы»»**

Выполнил студент

\_­\_\_\_\_\_\_\_\_\_\_\_\_\_Герасимов Д.А.

Группа П1-18

\_\_\_\_\_\_\_\_\_\_(Дата сдачи работы)

Принял преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гусятинер Л.Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Попов В.Н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Оценка)

**Королев 2021 г.**

Содержание

[1. Текстовое описание задачи 3](#_Toc76364907)

[2. Диаграмма прецедентов 4](#_Toc76364908)

[3. Диаграмма классов 5](#_Toc76364909)

[4. Определения классов 6](#_Toc76364910)

[5. Отладка одного из модулей при разработке 7](#_Toc76364911)

[6. Тестовые наборы и тестирование одного из модулей 8](#_Toc76364912)

[7. Сложность O для одного из методов 11](#_Toc76364913)

1. Текстовое описание задачи

Приложение состоит из 3-х главных активностей – будильник, таймер и секундомер.

Будильник выполняет следующие действия:

1. Добавление будильника
2. Сброс будильника
3. Срабатывание будильника
4. Выключение / Включение будильника

Таймер выполняет следующие действия:

1. Ввод таймера
2. Запуск таймера
3. Остановка таймера
4. Сброс таймера
5. Срабатывание таймера

Секундомер выполняет следующие действия:

1. Запуск секундомера
2. Остановка секундомера
3. Сброс секундомера
4. Диаграмма прецедентов

В данном разделе содержится диаграмма прецедентов для приложения, на которой показаны возможные функциональные отношения.

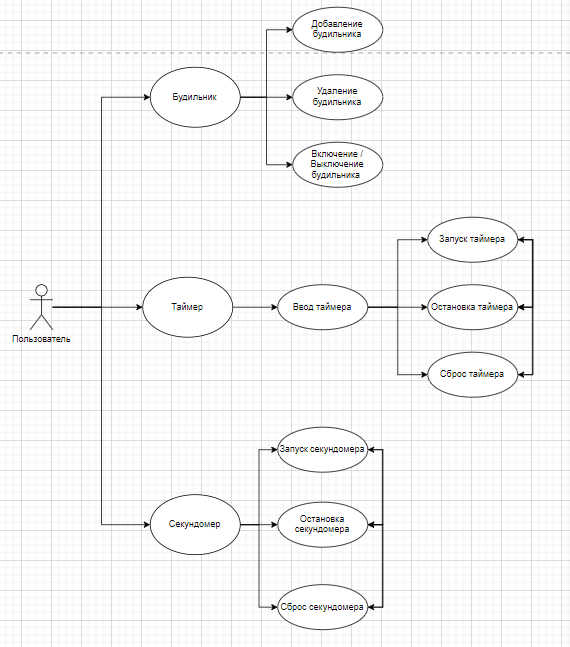


Рисунок 1. Диаграмма прецедентов

1. Диаграмма классов

В данном разделе находится диаграмма классов для приложения.

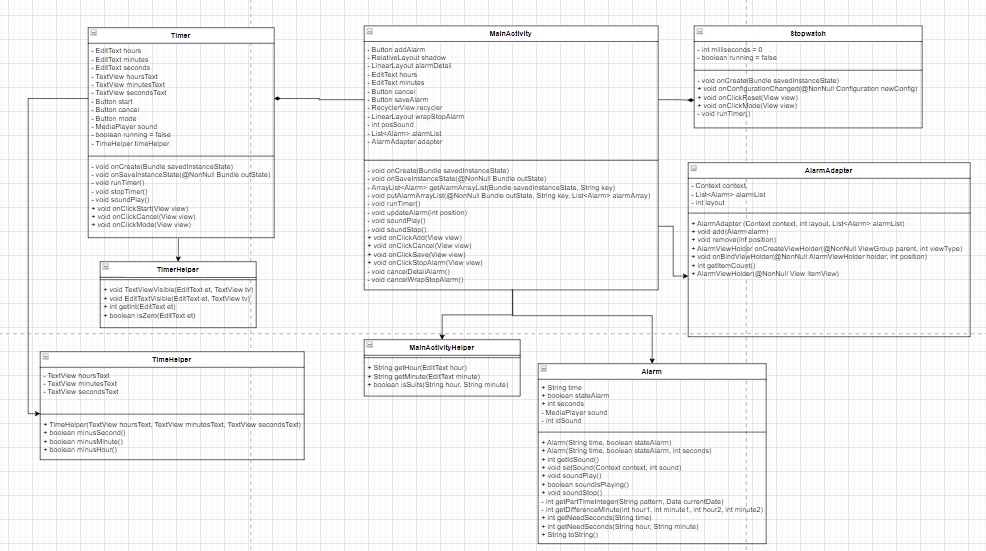


Рисунок 2. Диаграмма классов

1. Определения классов

Каждый класс в программе выполняет свое действие. Вот их определения:

1. **MainActivity** – класс отвечающий за функциональность будильника
2. **MainActivityHelper** – класс помогающий классу MainActivity
3. **Alarm** – класс представляющий каждый будильник по отдельности в списке будильников
4. **AlarmAdapter** – класс работающий со всеми будильниками на экране
5. **Timer** - класс отвечающий за функциональность таймера
6. **TimerHelper** – класс помогающий классу Timer
7. **TimeHelper** – класс работающий с вычислением времени
8. **Stopwatch** – класс отвечающий за функционал секундомера
9. Отладка одного из модулей при разработке

В ходе написания проекта при попытке запустить программу были получены ошибки:

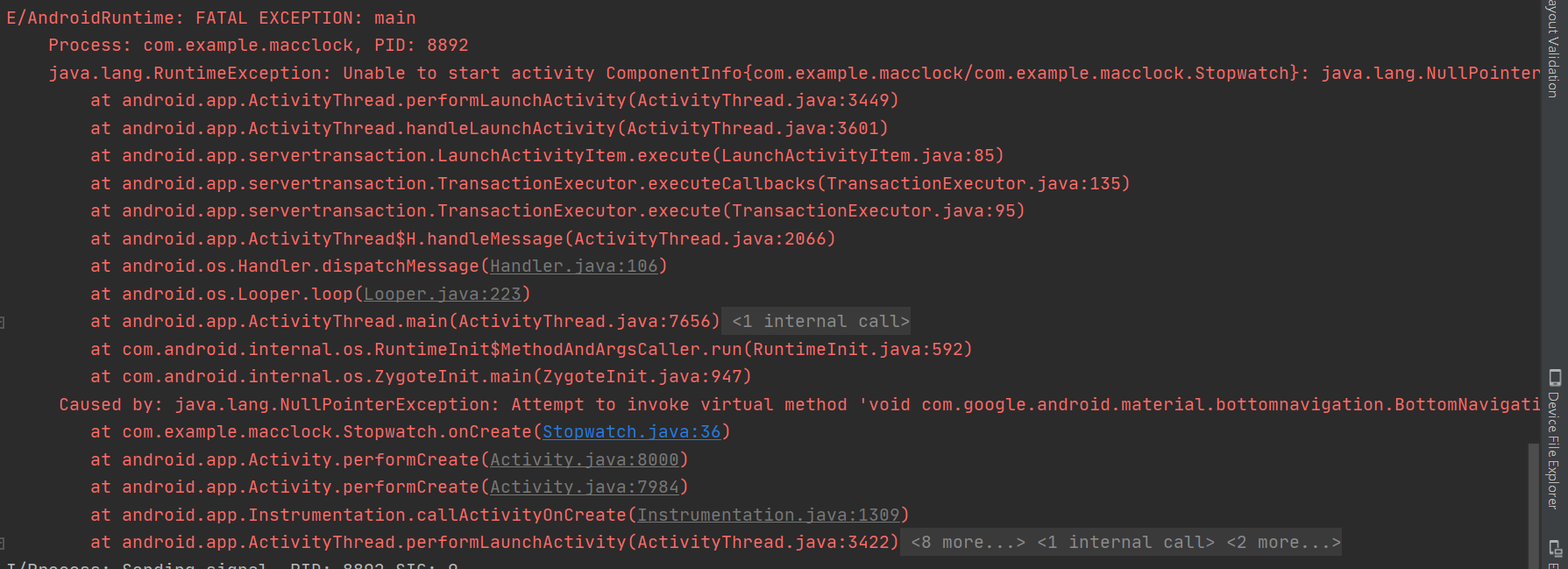


Рисунок 3. Сообщение об ошибках

После отладки ошибок не было:

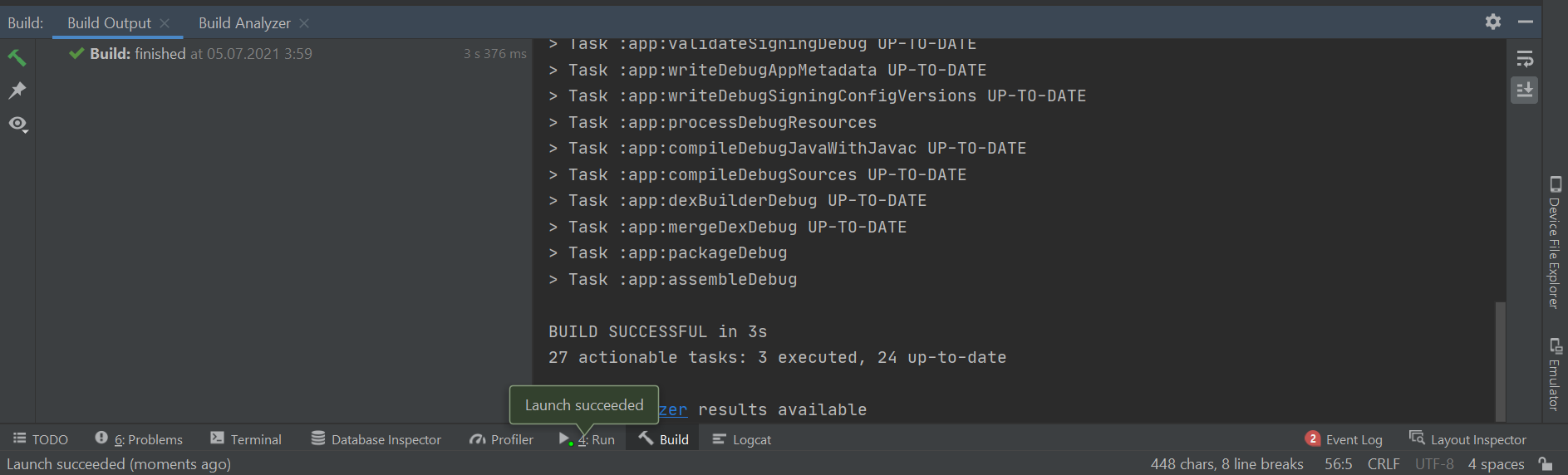


Рисунок 4. Сообщения после отладки программы

1. Тестовые наборы и тестирование одного из модулей

Возьмём модуль – будильник. При запуске будильника открывается такое пустое окно.

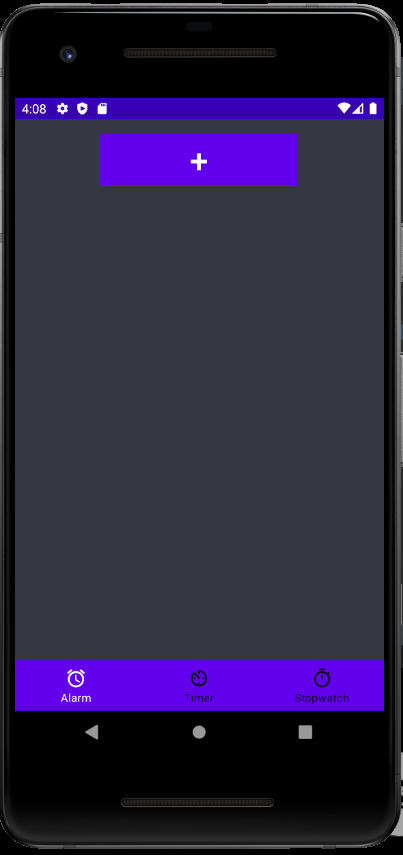


Рисунок 5. Будильник - стартовое окно

После запуска нажмем на кнопку «+». Эта кнопка отвечает за добавление будильника.

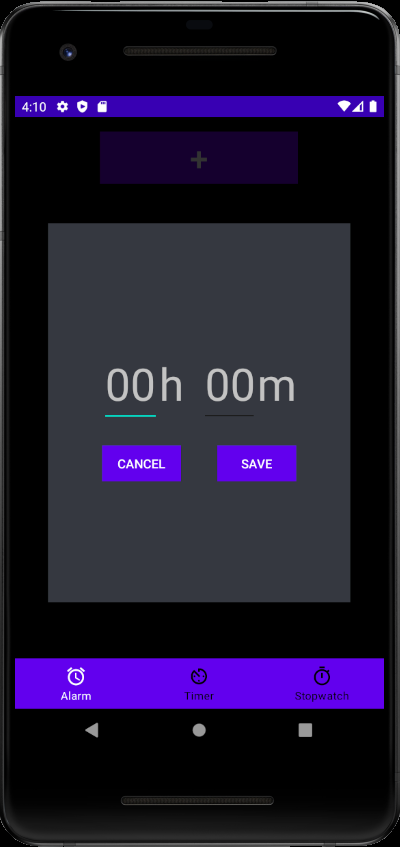


Рисунок 6. Будильник - статус добавления будильника

После ввода будильника и нажатия на кнопку «Save». Создастся новый будильник в активном состоянии.

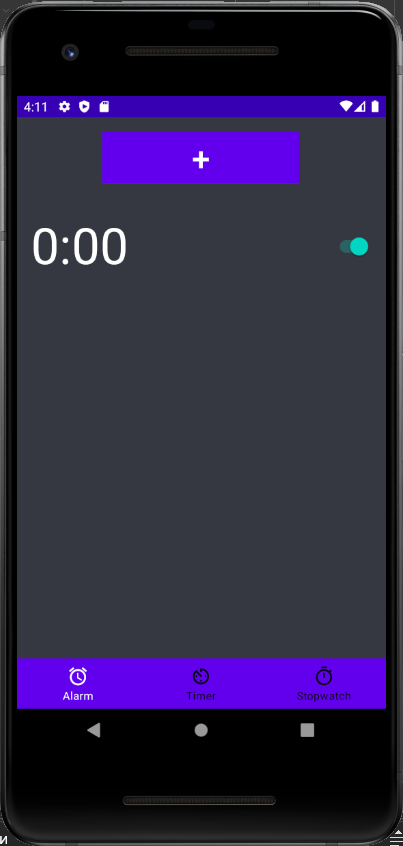


Рисунок 7. Будильник - после добавления будильника

После того, как время встанет вспять, сработает будильник. На фоне будет играть песня. После того, как вы нажмете на кнопку «Да», песня прекратится.

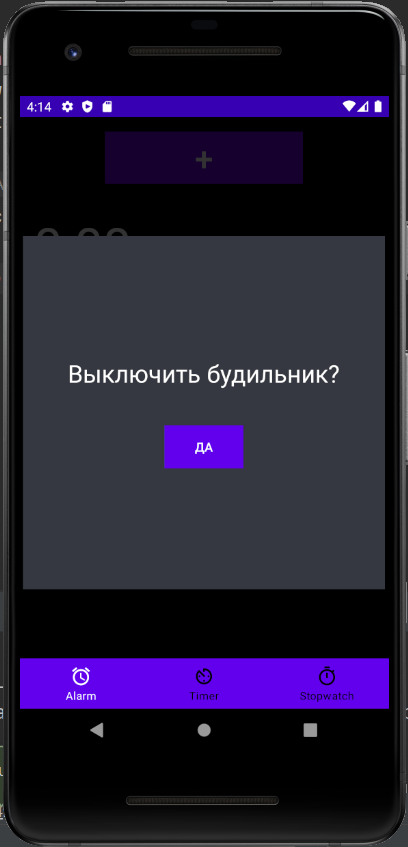


Рисунок 8. Будильник - проигрывание будильника

1. Сложность O для одного из методов

Сложность для данного метода – O(n). Мы рассматриваем цикл, т.к. данная часть метода возрастает быстрее всего. В этом случае – это цикл. Поэтому сложность такова.

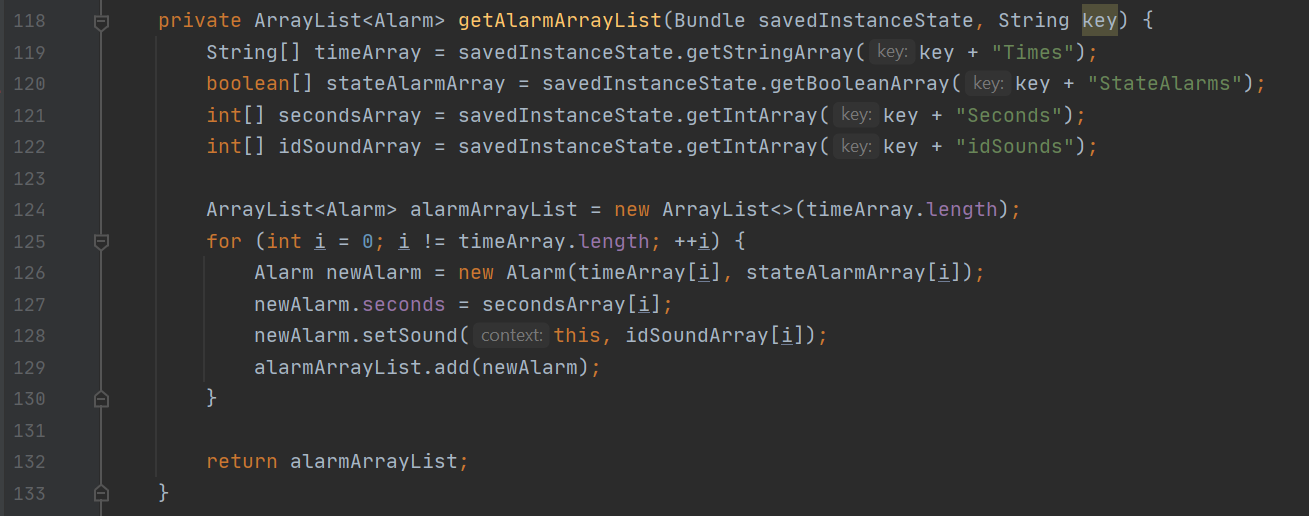


Рисунок 9. Метод возвращающий список будильников