



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Разработка программных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Создание программного приложения в среде Android Studio «Умный Будильник»

Студент: Замилацкая Диана Владимировна

Группа: ИКБО-13-18

Работа представлена к защите 15.05.2020 (дата)

/Замилацкая Д.В./
(подпись и ф.и.о. студента)

Руководитель: к.т.н., старший преподаватель Куликов Александр Анатольевич

Работа допущена к защите 15.05.2020 (дата)

_____/Куликов А.А./
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/ 21.06.2020, к.т.н., старший преподаватель Куликов Александр Анатольевич/

_____/ 21.06.2020, ассистент Рачков Андрей Владимирович/

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших
защиту)

Москва 2020

И. о. заведующего кафедрой
инструментального и прикладного
программного обеспечения (ИиППО)
Института информационных технологий (ИТ)
Болбакову Роману Геннадьевичу

От студента Замилацкой


Дианы Владимировны
ФИО
ИКБО-13-18
группа
2 курс
курс

Контакт: 8(909) 997-15-57,
darkshadow0368@gmail.com

Заявление

Прошу утвердить мне тему *курсовой работы* по дисциплине «*Разработка программных приложений*» образовательной программы бакалавриата 09.03.04 (*Программная инженерия*). Тема: Создание программного приложения в среде Android Studio «Умный Будильник».

Приложение: лист задания на КР/КП в 2-ух экземплярах на двухстороннем листе (проект)

Подпись студента	 подпись	/Замилацкая Д.В./ ФИО
Дата	<u>02.05.2020</u>	
Подпись руководителя	 подпись	/ст. преп. Куликов А.А./ Должность, ФИО
Дата	<u>02.05.2020</u>	



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ на выполнение курсовой работы

по дисциплине: Разработка программных приложений
по профилю: Разработка программных продуктов и проектирование информационных систем

Студент: Замилацкая Диана Владимировна

Группа: ИКБО-13-18

Срок представления к защите: 15.05.2020

Руководитель: к.т.н., старший преподаватель Куликов Александр Анатольевич

Тема Создание программного приложения в среде Android Studio «Умный Будильник»

Исходные данные: документация по языку Java, ГОСТ 19.105-78, ГОСТ 19.201-78, статьи по работе с Android Studio

Перечень вопросов, подлежащих разработке, и обязательного графического материала:
1. Провести анализ предметной области разработки программных приложений, 2. Определить перечень функциональных требований к программе, 3. Выбрать инструментальные средства для разработки программы, 4. Спроектировать модули программного приложения. 5. Реализовать модули программного приложения 6. Представить пояснительную записку.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

И. о. зав. кафедрой ИиППО: _____ / Болбаков Р. Г. /, «_____» _____ 2020 г.

Задание на КР выдал: _____ / Куликов А.А./, «_____» _____ 2020 г.

Задание на КР получил: Stk / Замилацкая Д.В./, «2» мая 2020 г.

АННОТАЦИЯ

Курсовая работа содержит 40 страниц, 20 иллюстраций, 2 приложения: первое с техническим заданием, второе с листингами файлов компановки. При написании работы использовалось 5 источников.

Тема работы - Создание программного приложения в среде Android Studio «Умный Будильник».

Объект исследования – программное приложение для среды Android, написанное на языке Java.

Цель работы – освоить на практике основные подходы программирования для разработки приложений для Android.

Основное назначение разработанной программы – предоставить пользователю возможность заводить будильники, для отключения которых необходимо решить математическую задачу. Таким образом исключается возможность бессознательного отключения будильника.

Данная программа имеет учебный характер, создание коммерческой версии не планируется.

Исходные тексты разработанного в течение КР ПП представлены в гит-репозитории, доступном по ссылке:
<https://github.com/DianaZam/MathAlarmClock>

Ключевые слова: ANDROID, JAVA, ALARMMANAGER, ПРОГРАММНОЕ ПРИЛОЖЕНИЕ, БУДИЛЬНИК.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	4
СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ	7
ВВЕДЕНИЕ	8
1 ОБЩИЕ СВЕДЕНИЯ.....	9
1.1 Обозначение и наименование программы.....	9
1.2 Программное обеспечение, необходимое для функционирования программы.....	9
1.3 Языки программирования, на которых написана программа	9
2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	10
3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	11
3.1 Декомпозиция	11
3.2 Описание зависимостей.....	12
3.3 Описание интерфейсов.....	12
3.4 Описание деталей.....	13
3.4.1 Класс MainActivity	13
3.4.2 Класс AlarmBaseAdapter.....	14
3.4.3 Класс AlarmBroadcastReceiver	15
3.4.4 Класс AlarmService	16
3.4.5 Класс AlarmTimePicker.....	16
3.4.6 Класс AlarmEditDialog.....	16
3.4.7 Класс AlarmDialog.....	17
3.4.8 Класс AlarmSQLiteOpenHelper	18
3.4.9 Класс AlarmData.....	19

3.5 Описание графического интерфейса.....	20
4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	26
5 ВЫЗОВ И ЗАГРУЗКА	27
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	29
Приложение А	30
Приложение Б	34

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ

КР – курсовая работа;

ТЗ – техническое задание;

ОС – Операционная система;

ОПП – Объектно-ориентированное программирование;

ПП – Программное приложение;

БД – база данных;

UML – Unified Modeling Language.

ВВЕДЕНИЕ

Известно, что будильником называют часы, в заданный момент времени подающие звуковой сигнал. Русское название «будильник» указывает на основное назначение таких часов – пробуждение человека. Также будильник может применяться как «напоминатель» или таймер.

Будильники стали незаменимой частью жизни современного человека, поэтому выбранная тема – «Создание программного приложения в среде Android Studio «Умный Будильник»» – является актуальной.

В данной курсовой работе приведена разработка ПП «Умный Будильник» в среде Android Studio. Данная тема была выбрана из-за необходимости создания приложения, которое предоставляло бы возможность заводить будильники, для отключения которых необходимо было бы решить математическую задачу, чтобы исключить возможность бессознательного отключения будильника.

Объект исследования – программное приложение для среды Android, написанное на языке Java.

Цель работы – освоить на практике основные подходы программирования для разработки приложений для Android.

В результате выполнения курсовой работы была получена и отработана следующая компетенция – ПК-13 – готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности

Выполнение курсовой работы должно опираться на положения СМКО МИРЭА 7.5.1/04.И.05-18 «Инструкция по организации и проведению курсового проектирования» от 30.05.2018.

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Обозначение и наименование программы

Название данного приложения – «Умный будильник».

1.2 Программное обеспечение, необходимое для функционирования программы

Для функционирования приложения необходимо установить программный продукт на персональное устройство, функционирующее на базе ОС Andoid версии 4.4 или выше.

1.3 Языки программирования, на которых написана программа

Данная программа написана на языке программирования Java.

Java – объектно-ориентированный высокоуровневый язык программирования, который предоставляет широкий спектр возможностей для написания ПО для серверов, мобильных устройств и персональных компьютеров.

Java официально поддерживается Google, компанией-разработчиком операционной системы Android. Значительная часть приложений для мобильных устройств реализованы именно с помощью данного языка программирования.

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Приложение «Умный будильник» представляет собой будильник. Приложению требуется обеспечивать пользователя возможностью добавлять будильники на определённое время, устанавливать мелодию будильника, делать его одноразовым или с ежедневным повтором, редактировать и удалять будильники. Количество будильников не ограничено. Выключение работающего будильника должно выполняться после решения пользователем простого математического примера.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Декомпозиция

Исходный код приложения представлен в пакете `ru.riddlezone.mathalarmclock`, в котором содержатся следующие классы:

- `MainActivity` – класс активности;
- `AlarmBaseAdapter` – класс адаптера;
- `AlarmBroadcastReceiver` – класс обработки сообщений;
- `AlarmService` – класс сервиса;
- `AlarmTimePicker` – класс диалога выбора времени;
- `AlarmEditDialog` – класс диалога изменения будильника;
- `AlarmDialog` – класс диалога будильника;
- `AlarmSQLiteOpenHelper` – класс создания базы данных;
- `AlarmData` – экземпляр будильника.

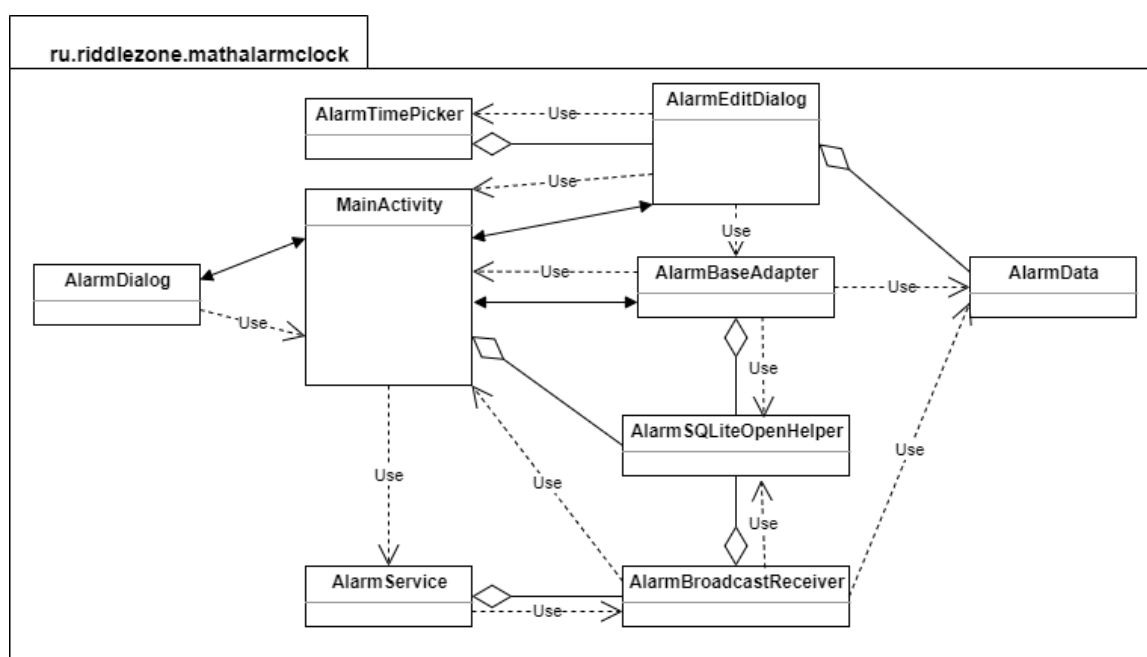


Рисунок 3.1 – UML диаграмма пакета `ru.riddlezone.mathalarmclock`

UML диаграмма пакета представлена на рисунке 3.1. На диаграмме классы представлены без описания их внутреннего строения, так как оно будет подробно рассмотрено в подпункте 3.4.

3.2 Описание зависимостей

Визуальное представление связей классов внутри пакета `ru.riddlezone.mathalarmclock` было представлено выше на рисунке 3.1.

Что касается внешних относительно пакета `ru.riddlezone.mathalarmclock` классов, ниже приведу наиболее значимые их использования:

- `MainActivity` наследует от `AppCompatActivity`, использует `ListView` (прокручиваемый список) для создания списка будильников.
- `AlarmBaseAdapter` наследует от `BaseAdapter` (для создания собственного адаптера, заполняющего список `ListView`) и использует `Cursor` (для работы с записями в базе данных).
- `AlarmBroadcastReceiver` наследует от `BroadcastReceiver` (для обработки событий), использует `Cursor`, `AlarmManager` (для генерации событий).
- `AlarmService` наследует от `Service`.
- `AlarmTimePicker` наследует от `DialogFragment`, реализует `TimePickerDialog.OnTimeSetListener`, использует `Calendar`, `Date`, `GregorianCalendar` для работы с временем и виджет `TimePicker` для выбора времени.
- `AlarmEditDialog` наследует от `AlertDialog`, использует `MediaPlayer`.
- `AlarmDialog` наследует от `AlertDialog`.
- `AlarmSQLiteOpenHelper` наследует от `SQLiteOpenHelper` (для создания базы данных и работы с ней), использует `SQLiteDatabase` и `Cursor`.
- `AlarmData` использует `Calendar`, `Date`, `GregorianCalendar` для работы с временем.

3.3 Описание интерфейсов

В приложении используются 5 `.xml` файлов из папки `/res/layout` и 1 из папки `/res/menu`.

Активити MainActivity использует макет активити activity_main.xml (приложение Б, листинг Б.1), который в свою очередь ссылается на макет content_main.xml (приложение Б, листинг Б.2), и макет меню menu_main.xml (приложение Б, листинг Б.3).

Адаптер AlarmBaseAdapter использует макет alarm_item.xml (приложение Б, листинг Б.4).

Диалог AlarmDialog использует макет alarm_dialog.xml (приложение Б, листинг Б.5).

Диалог AlarmEditDialog использует макет alarm_edit.xml (приложение Б, листинг Б.6).

3.4 Описание деталей

Далее будут представлены описания классов из пакета ru.riddlezone.mathalarmclock. Исходный текст классов можно увидеть в приложении Б.

3.4.1 Класс MainActivity

На рисунке 3.2 представлена UML-диаграмма класса MainActivity.

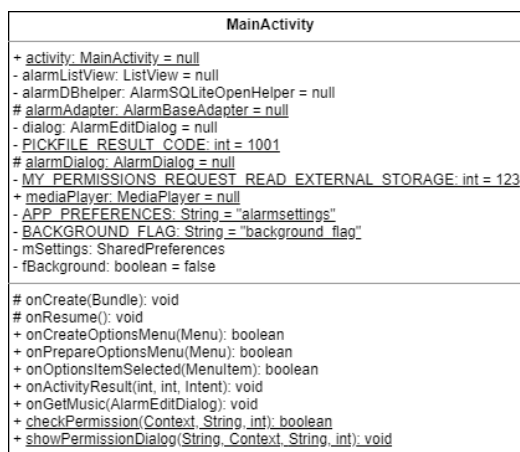


Рисунок 3.2 – UML диаграмма класса MainActivity

Метод onCreate() вызывается при создании основного Activity, в нем происходит проверка разрешения на допуск к хранилищу, подключение к базе данных, создание адаптера и заполнение списка будильников.

Метод onResume() вызывается каждый раз, когда активность идёт на переднем плане. Он вызывает AlertDialog, если работает MediaPlayer и диалог ещё не вызван.

Метод onCreateOptionsMenu() вызывается системой при создании меню.

Метод onPrepareOptionsMenu() обрабатывает нажатие на action bar item.

Метод onOptionsItemSelected() - метод обработки нажатий пунктов меню.

Метод onActivityResult(), если передан результат активности по выбору мелодии, передаёт ссылку на мелодию в диалог создания/редактирования будильника.

Метод onGetMusic() запускает намерение для выбора мелодии из хранилища устройства.

Метод checkPermission() проверяет наличие разрешения на доступ к внешнему хранилищу.

Метод showPermissionDialog() создаёт диалоговое окно для запроса разрешения на доступ к внешнему хранилищу.

3.4.2 Класс AlarmBaseAdapter

На рисунке 3.3 представлена UML-диаграмма класса AlarmBaseAdapter.

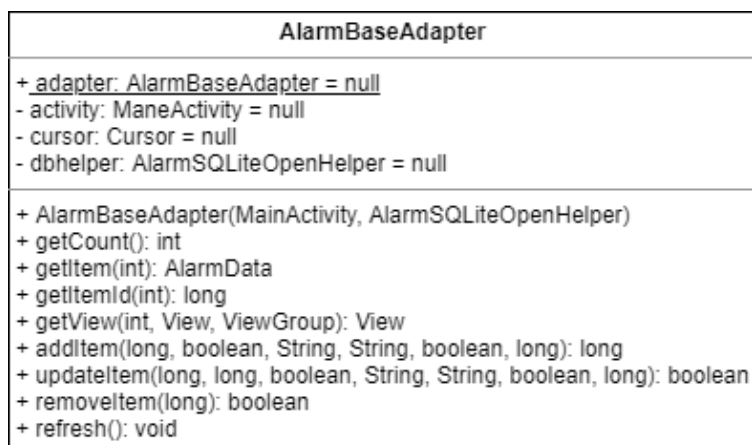


Рисунок 3.3 – UML диаграмма класса AlarmBaseAdapter

Метод getCount() возвращает количество будильников в таблице БД.

Метод getItem() возвращает экземпляр будильника по номеру позиции.

Метод getItemId() возвращает ID будильника по номеру позиции.

Метод `getView()` выводит будильник в список, а так же устанавливает обработчики нажатия кнопок и изменения индикаторов.

Метод `addItem()` добавляет запись будильника в таблицу БД.

Метод `updateItem()` изменяет запись будильника в таблице БД.

Метод `removeItem()` удаляет запись будильника из таблицы БД.

Метод `refresh()` обновляет список будильников.

3.4.3 Класс `AlarmBroadcastReceiver`

На рисунке 3.4 представлена UML-диаграмма класса `AlarmBroadcastReceiver`.

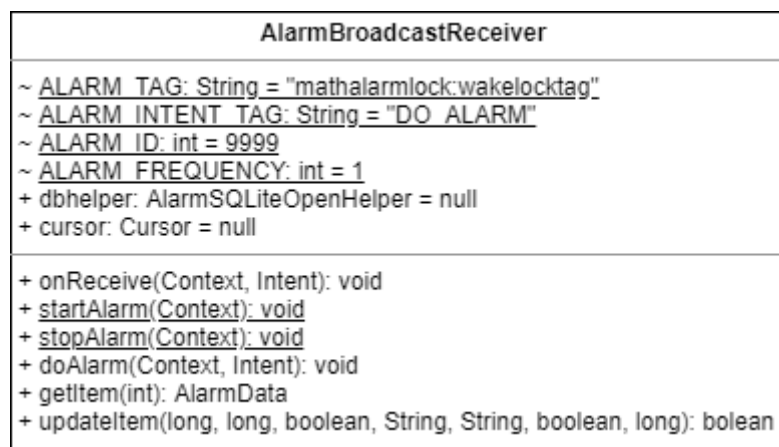


Рисунок 3.4 – UML диаграмма класса `AlarmBroadcastReceiver`

Метод `onReceive()` вызывается при получении широковещательного намерения и вызывает `doAlarm()`.

Метод `startAlarm()` запускает обработку службы сообщений.

Метод `stopAlarm()` останавливает обработку службы сообщений.

Метод `doAlarm()` обрабатывает будильники. Если найден будильник, который должен сработать в данное время, то запускается проигрыватель с выбранной мелодией и вызывается `MainActivity`.

Метод `getItem()` возвращает экземпляр будильника по номеру позиции в таблице.

Метод `updateItem()` изменяет запись будильника в таблице БД.

3.4.4 Класс AlarmService

На рисунке 3.5 представлена UML-диаграмма класса AlarmService.

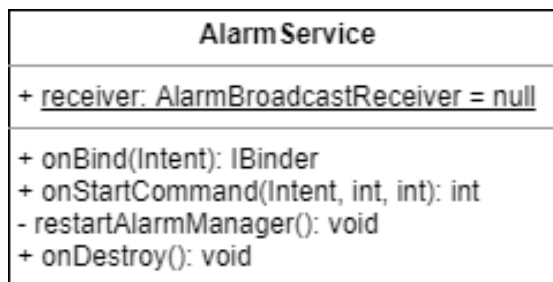


Рисунок 3.5 – UML диаграмма класса AlarmService

Метод onStartCommand() вызывает restartAlarmManager().

Метод restartAlarmManager() создаёт экземпляр AlarmBroadcastReceiver и запускает обработку службы сообщений при помощи startAlarm().

Метод onDestroy() останавливает обработку службы сообщений при помощи AlarmBroadcastReceiver.stopAlarm().

3.4.5 Класс AlarmTimePicker

На рисунке 3.6 представлена UML-диаграмма класса AlarmTimePicker.

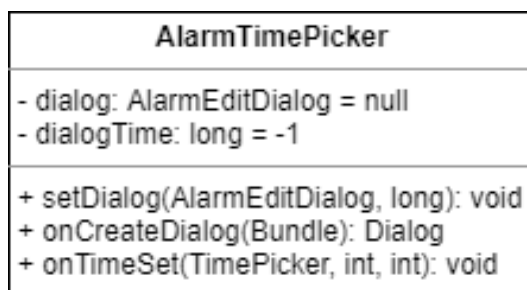


Рисунок 3.6 – UML диаграмма класса AlarmTimePicker

Метод setDialog() устанавливает значения диалога, откуда был вызван диалог выбора времени.

Метод onCreateDialog() создаёт диалог выбора времени.

Метод onTimeSet() обрабатывает результат выбора времени, то есть передаёт результат в диалог создания/изменения параметров будильника.

3.4.6 Класс AlarmEditDialog

На рисунке 3.7 представлена UML-диаграмма класса AlarmEditDialog.

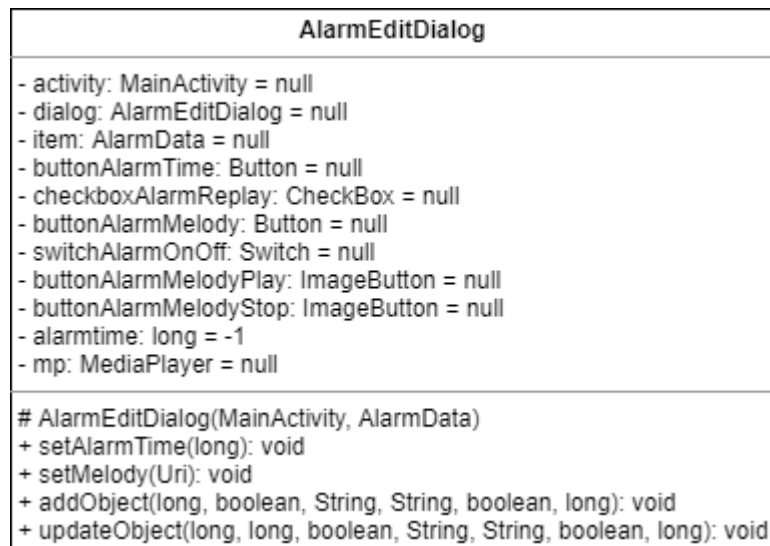


Рисунок 3.7 – UML диаграмма класса AlarmEditDialog

Конструктор AlarmEditDialog() создаёт диалоговое окно создания/изменения будильника.

Метод setAlarmTime() устанавливает текст кнопки выбора времени.

Метод setMelody() устанавливает текст кнопки выбора мелодии и показывает или скрывает кнопку проигрывания мелодии.

Метод addObject() добавляет новый будильник, вызывая AlarmBaseAdapter.adapter.addItem().

Метод updateObject() изменяет существующий будильник, вызывая AlarmBaseAdapter.adapter.updateItem().

3.4.7 Класс AlarmDialog

На рисунке 3.8 представлена UML-диаграмма класса AlarmDialog.

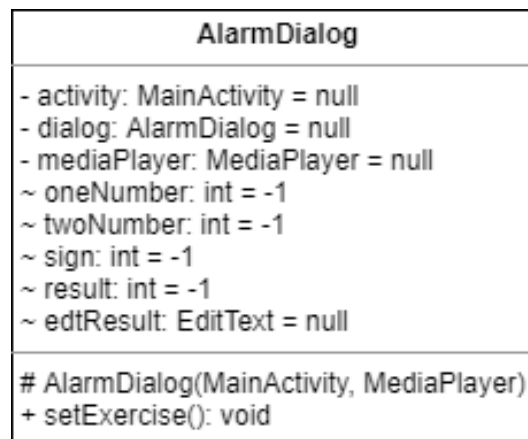


Рисунок 3.8 – UML диаграмма класса AlarmDialog

Конструктор AlarmDialog() создаёт диалоговое окно выключения будильника.

Метод setExercise() задаёт случайный простой математический пример.

3.4.8 Класс AlarmSQLiteOpenHelper

На рисунке 3.9 представлена UML-диаграмма класса AlarmSQLiteOpenHelper.

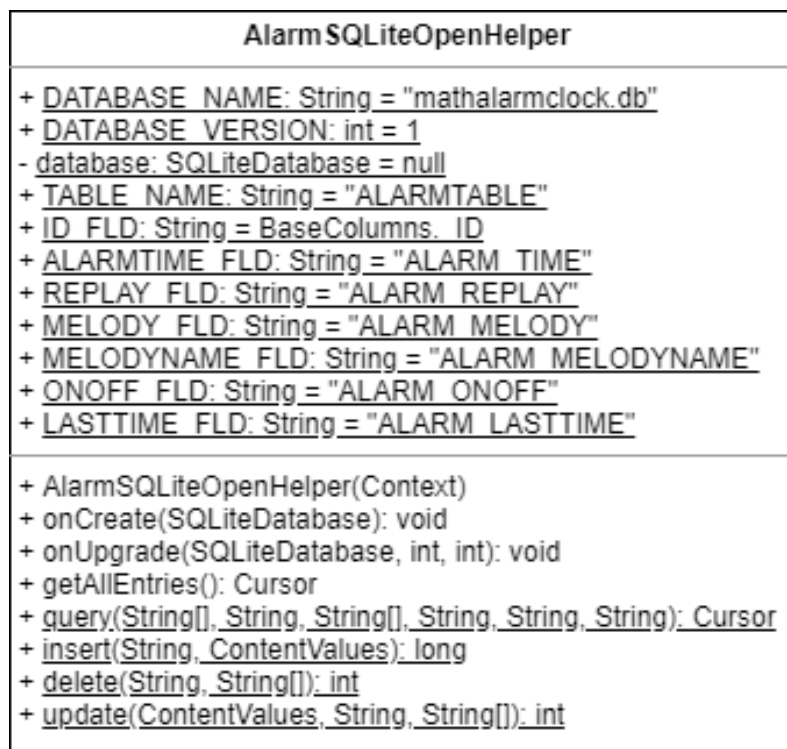


Рисунок 3.9 – UML диаграмма класса AlarmSQLiteOpenHelper

Конструктор AlarmSQLiteOpenHelper() создаёт экземпляр БД mathalarmclock.db.

Метод onCreate() создаёт таблицу ALARMTABLE.

Метод onUpgrade() пересоздаёт таблицу ALARMTABLE.

Метод getAllEntries() возвращает курсор на все записи в таблице ALARMTABLE.

Метод query() выполняет запрос SELECT.

Метод insert() выполняет запрос INSERT.

Метод delete() выполняет запрос DELETE.

Метод update() выполняет запрос UPDATE.

3.4.9 Класс AlarmData

На рисунке 3.10 представлена UML-диаграмма класса AlarmData.

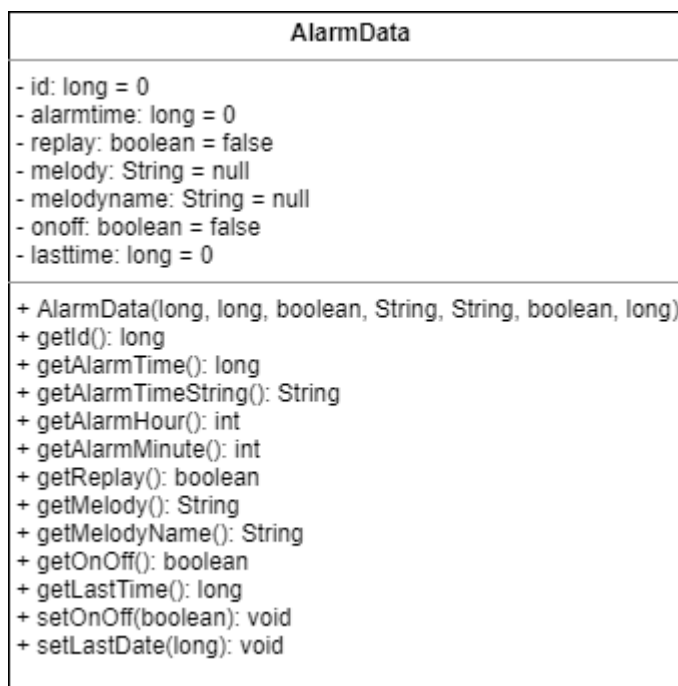


Рисунок 3.10 – UML диаграмма класса AlarmData

Конструктор `AlarmData()` создаёт экземпляр будильника.

Метод `getId()` – геттер, возвращающий ID будильника.

Метод `getAlarmTime()` – геттер, возвращающий время будильника.

Метод `getAlarmTimeString()` возвращает время в формате HH:MM.

Метод `getAlarmHour()` возвращает час будильника.

Метод `getAlarmMinute()` возвращает минуты будильника.

Метод `getReplay()` – геттер, возвращающий индикатор повтора.

Метод `getMelody()` – геттер, возвращающий путь к мелодии.

Метод `getMelodyName()` – геттер, возвращающий название мелодии.

Метод `getOnOff()` – геттер, возвращающий индикатор активности.

Метод `getLastTime()` – геттер, возвращающий время последнего срабатывания.

Метод `setOnOff()` – сеттер, устанавливающий значение индикатора активности.

Метод `setLastDate()` – сеттер, устанавливающий время последнего срабатывания.

3.5 Описание графического интерфейса

Графический интерфейс приложения представлен на рисунках ниже.

При первом запуске приложения появляется просьба предоставить доступ к хранилищу (рис. 3.11).

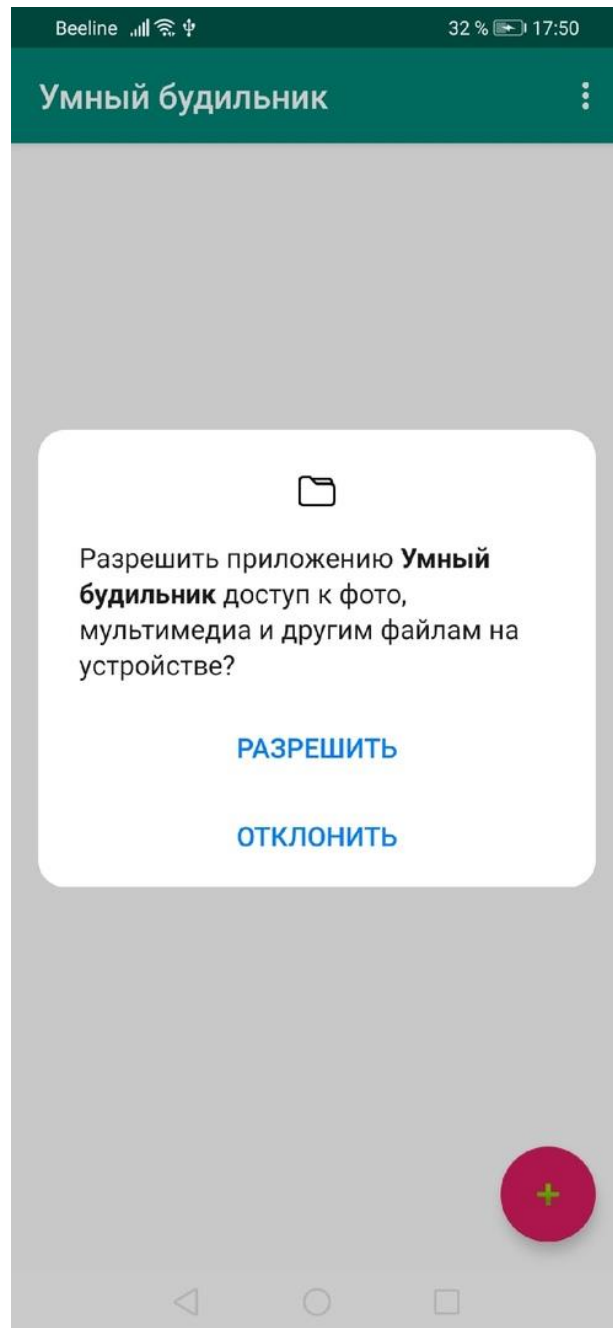


Рисунок 3.11 – Разрешение доступа

При запуске открывается MainActivity со списком установленных будильников. Если ни одного будильника ещё не создано или все будильники удалены, список пуст. Интерфейс представлен на рисунке 3.12.

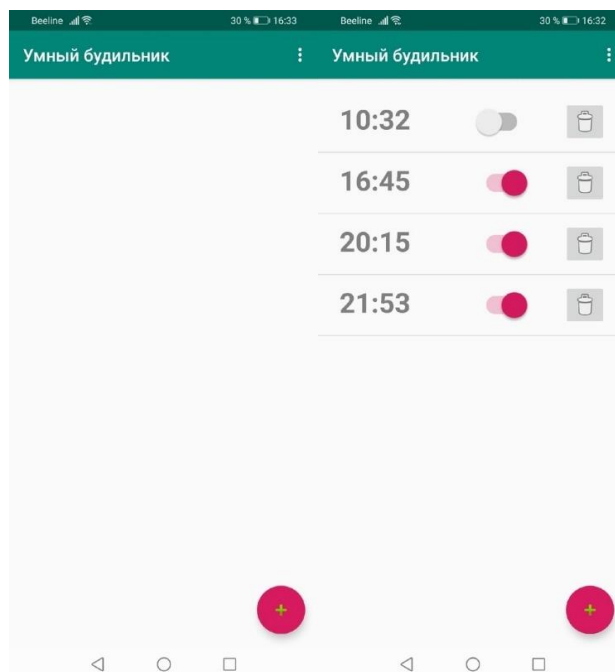


Рисунок 3.12 – MainActivity

При нажатии на кнопку экшн-бара появляется индикатор включения/отключения сервиса обработки будильников (рис. 3.13).

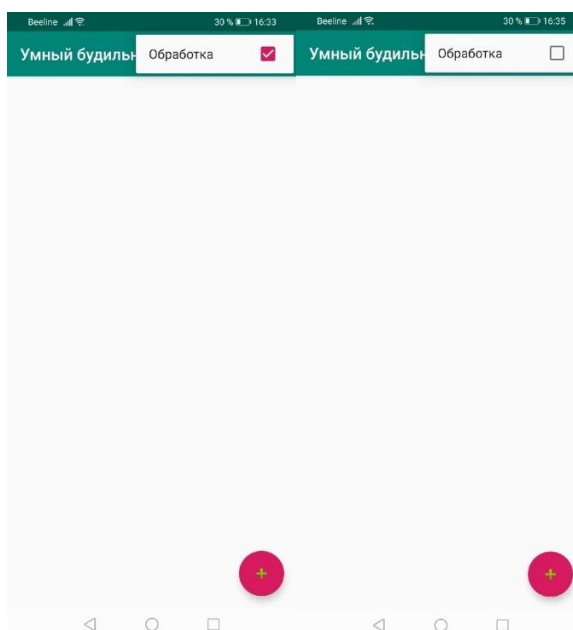


Рисунок 3.13 – Индикатор обработки

При нажатии на FloatingActionButton появляется диалоговое окно создания будильника (рис. 3.14). Здесь можно задать время срабатывания будильника, является ли он ежедневным, включён (активен) ли он, а также выбрать мелодию. После выбора мелодии становится видимой кнопка произведения мелодии.

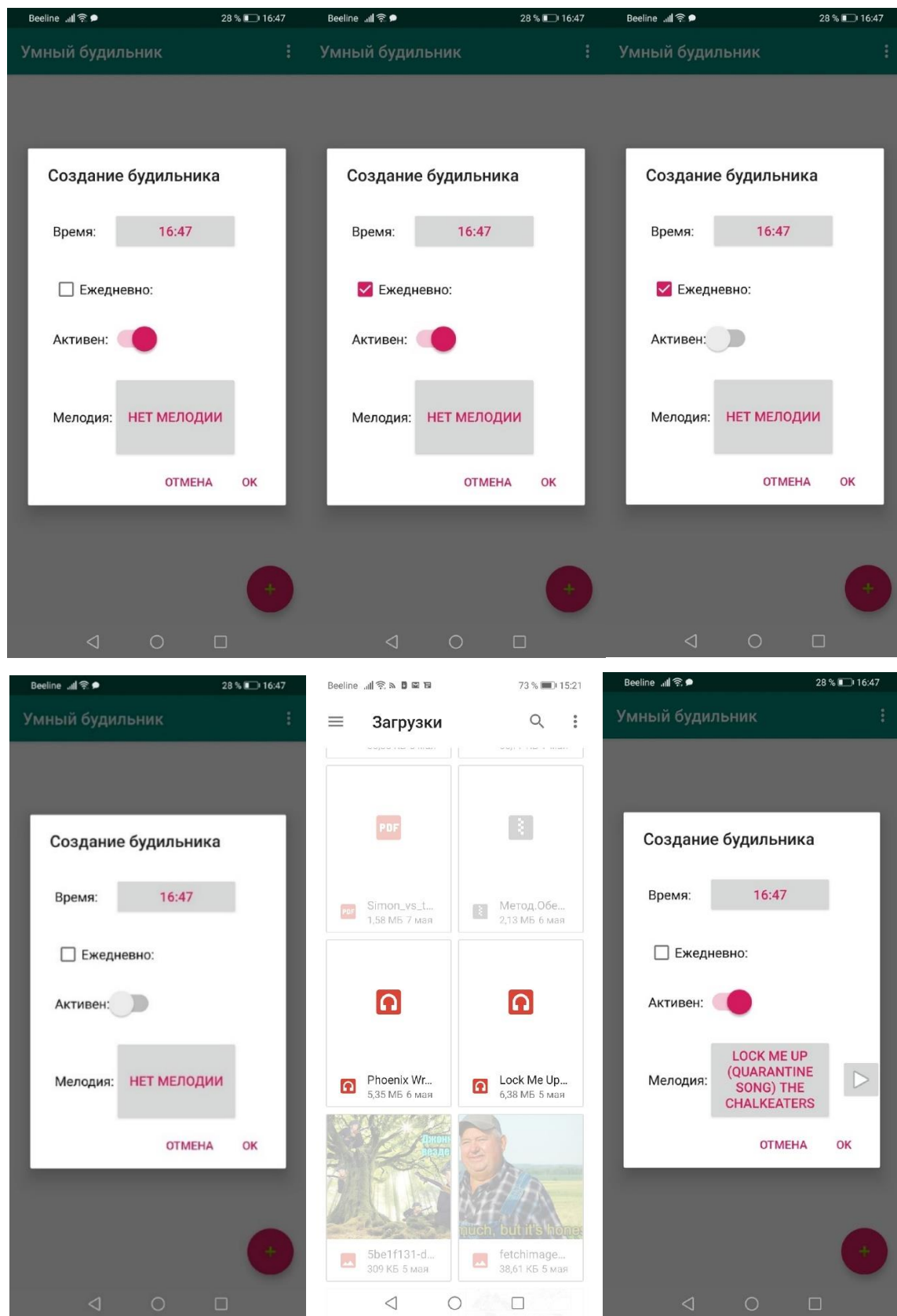


Рисунок 3.14 – Создание будильника

При нажатии на кнопку выбора времени появляется диалог выбора времени (рис. 3.15).



Рисунок 3.15 – Диалог выбора времени

После создания будильника он появляется в списке (рис. 3.16).

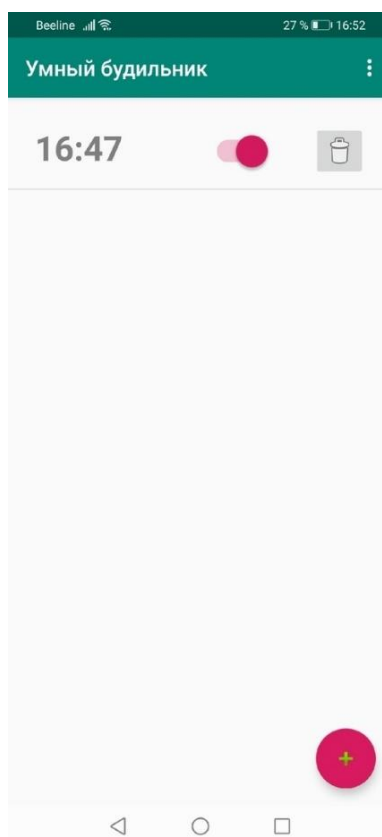


Рисунок 3.16 – Обновлённый после создания будильника список

При нажатии на запись из списка открывается диалог изменения будильника (рис. 3.17).

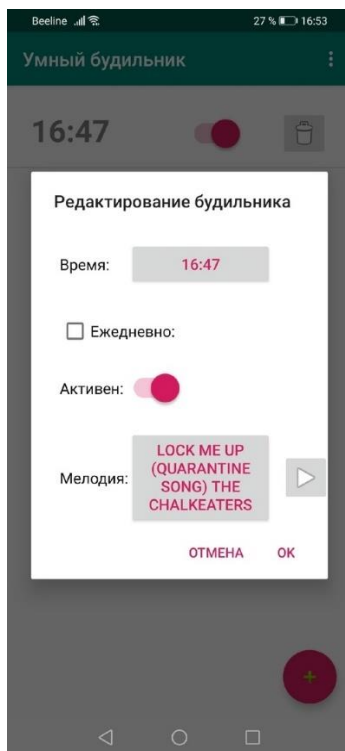


Рисунок 3.17 – Диалог изменения будильника

При нажатии на кнопку удаления будильника появляется диалог удаления будильника (рис. 3.18).

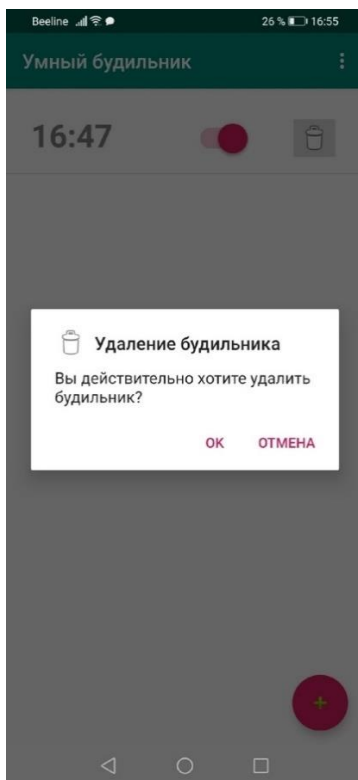


Рисунок 3.18 – Диалог удаления будильника

После обновления будильника обновляется список (рис. 3.19).

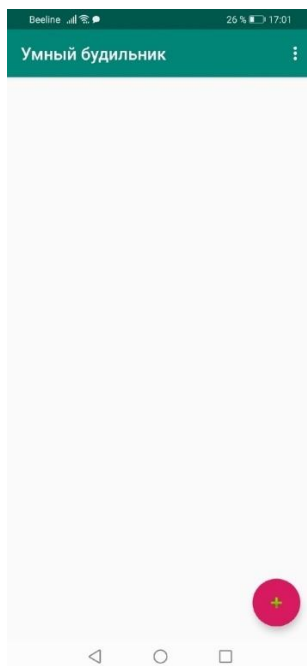


Рисунок 3.19 – Обновлённый после изменения будильника список

При срабатывании будильника начинает играть мелодия, запускается MainActivity (при свёрнутом приложении) и, если будильник не ежедневный, индикатор активности автоматически переключается. Также открывается диалог отключения будильника. При вводе неверного результата, диалог заменяется новым, с иным примером. После ввода верного результата диалог закрывается и мелодия перестаёт играть. Описанный процесс представлен на рисунке 3.20.

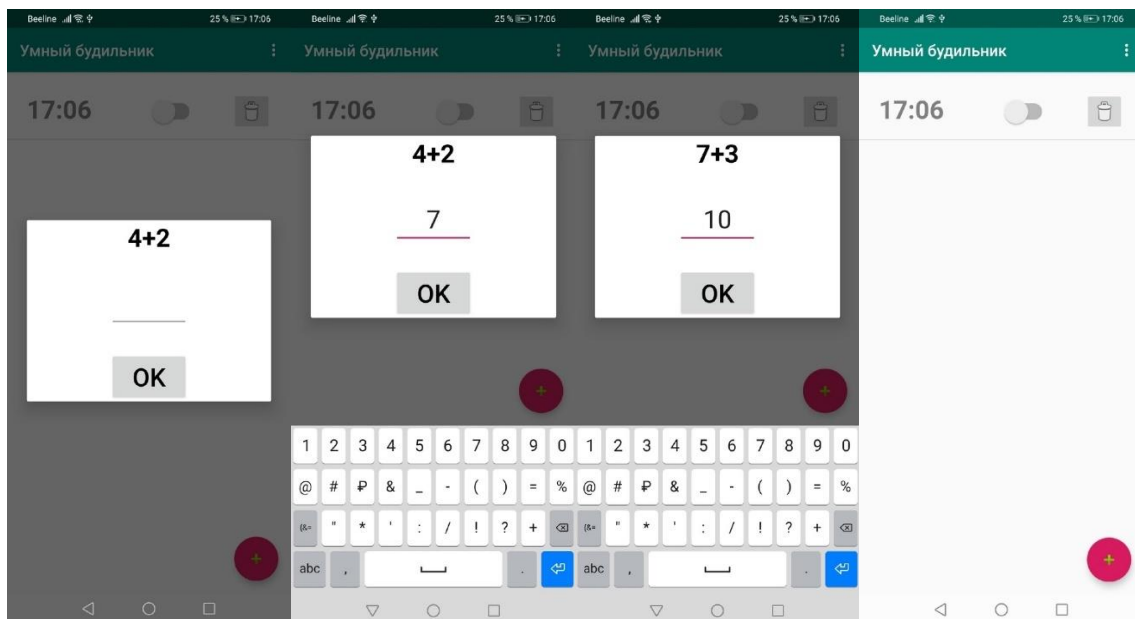


Рисунок 3.20 – Выключение будильника

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для корректного функционирования программы необходимо около 2 МБ свободного места в внутреннем хранилище мобильного устройства, а также предоставление доступа к хранилищу. На некоторых устройствах (например, смартфонах huawei) рекомендуется вручную запретить оптимизацию батареи для этого приложения.

5 ВЫЗОВ И ЗАГРУЗКА

Объем программы, содержащей 7 записей в базе данных, составляет примерно 3 МБ. А для начала работы программы достаточно установить её на мобильное устройство с помощью открытия файла MathAlarm.apk, содержащего архив с программой.

ЗАКЛЮЧЕНИЕ

Исходный код приложения, разработанного в ходе КР, можно увидеть в гит-репозитории, доступном по ссылке:
<https://github.com/DianaZam/MathAlarmClock>

По итогам разработки курсовой работы были выполнены следующие цели:

- Для приложения был спроектирован и разработан интерфейс, который доступно демонстрирует возможности, предоставляемые пользователю.
- Все функции приложения, указанные в техническом задании, реализованы в полной мере и работают без ошибок.

Исходя из представленных тезисов, поставленную в начале данной работы цель можно считать выполненной.

В результате выполнения курсовой работы была получена и отработана следующая компетенция – ПК-13 – готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лафоре, Р. Структуры данных и алгоритмы в Java / Роберт Лафоре.- 2-е изд., перераб. и доп.- СПб.: Питер, 2013.- 704 с. (Первое обращение – 15.03.20)
2. Алгоритмы: построение и анализ / Томас Кормен Х. [и др.].— 2-е изд.- М.: Издательский дом “Вильямс”, 2005.- 1296 с. (Первое обращение – 15.03.20)
3. Уоррен-Младший, Г. Алгоритмические трюки для программистов / Генри Уоррен-Младший.— 2-е изд.— М.: ООО “И. Д. Вильямс”, 2014.— 512 с. (Первое обращение – 15.03.20)
4. Макконелл, Дж. Основы современных алгоритмов / Джон Макконелл.- 2-е изд., перераб - М.: Техносфера, 2004.- 368 с. (Первое обращение – 15.03.20)
5. Android. Программирование для профессионалов Харди Брайан. [и др.].— 2-е изд.- СПб Питер, 2016.— 640 с. (Первое обращение – 15.03.20)

Приложение А

Техническое задание

А.1 Введение

Составленное техническое задание по дисциплине «Разработка программных приложений» является документом к курсовой работе, который отображает этапы разработки, в том числе стадии проектирования и документирования программы.

А.1.1 Наименование программы

Название данного приложения – «Умный будильник».

А.1.2 Краткая характеристика области применения программы

Программа предназначена для помощи в пробуждении и отслеживании времени при помощи будильников.

А.2 Основание для разработки

Основанием для разработки является курсовая работа по дисциплине «Разработка программных приложений», предусмотренная учебным планом программы бакалавриата 09.03.04 «Программная инженерия» с учетом специфики профиля «Разработка программных приложений и проектирование информационных систем».

А.3 Назначение для разработки

Создание приложения, способного дать пользователю возможность работать с будильниками.

А.4 Требования, предъявляемые к программе

Приложение должно обеспечивать пользователя возможностью добавлять будильники на определённое время, устанавливать мелодию будильника, делать его одноразовым или с ежедневным повтором, редактировать и удалять будильники. Количество будильников не ограничено. Выключение работающего будильника должно выполняться после решения пользователем простого математического примера.

А.4.1 Взаимодействие продукта с другими продуктами и компонентами

В приложении не реализовано взаимодействие с другими продуктами.

А.4.2 Функциональные требования

Приложение должно реализовывать следующие функции:

1. Создание неограниченного количества будильников, информация о которых должна храниться в таблице базы данных.
2. Удаление любого существующего будильника.
3. Изменение любого существующего будильника.
4. Установка времени срабатывания будильника.
5. Установка мелодии будильника из аудиофайлов, хранящихся на устройстве.
6. Задание ежедневного или одноразового повтора будильника.
7. Задание параметра активности будильника (если будильник создан, но выключен, при наступлении момента срабатывания ничего не произойдёт).
8. Выключение работающего будильника по средству решения простого математического примера.

А.4.3 Требования к внешним интерфейсам

Для корректной работы приложения необходимо, чтобы устройство имело доступ к хранилищу.

A.4.4 Требования к производительности

Приложение должно работать стабильно, таким образом, чтобы действия пользователя не вызвали аварийного завершения программы.

A.4.5 Нефункциональные требования

Требования к доступности и безопасности приложения не предъявляются.

A.5 Требования к программной документации

1. Пояснительная записка, оформленная в соответствии с локально-нормативными актами РТУ МИРЭА (СМКО МИРЭА 7.5.1/04.И.05-18);
2. Отчётная документация, составленная в соответствии с ГОСТ.

В процессе создания приложения вся проделанная работа документируется, должны быть отражены детали разработки. Всё вышеперечисленное должно быть отражено в пояснительной записке, которая прилагается к работе.

A.6 Техничко-Экономические показатели

Техничко-экономические требования к работе не предъявляются.

A.7 Стадии и этапы разработки

Этапы разработки:

3. Исследование концепции: 10.02.20 – 10.02.20.
4. Выработка требования и составление ТЗ: 11.02.20 – 11.02.20.
5. Проектирование: 13.02.20 – 15.02.20.
6. Реализация компонент, написание логики программы: 15.02.20 – 21.03.20.
7. Интеграция компонент, написание графического интерфейса: 21.03.20 – 05.04.20.

8. Отладка созданного приложения: 01.04.20 – 15.04.20.
9. Оформление отчетной документации: 15.03.20 – 10.05.20.
10. Представление работы к защите: 15.05.2020.

Дальнейшее сопровождение программы после сдачи и защиты курсового проекта не предусмотрено.

А.8 Порядок контроля и приема работы

Прием работы осуществляется путем защиты курсовой работы у преподавателя, назначенного руководителем КР. Защита осуществляется в соответствии с установленным графиком защиты курсовых работ.

Приложение Б

Листинг файлов компановки

Листинг Б.1 – *activity_main.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />
    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:srcCompat="@android:drawable/ic_input_add" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Листинг Б.2 – *content_main.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context=".MainActivity"
tools:showIn="@layout/activity_main">

<ListView
    android:id="@+id/alarmListView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:paddingTop="@dimen/alarmlistview_padding_top"
    android:paddingBottom="@dimen/alarmlistview_padding_bottom"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Листинг Б.3 – menu_main.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="ru.riddlezone.mathalarmclock.MainActivity">
    <item
        android:id="@+id/background"
        android:orderInCategory="100"
        android:title="@string/action_background"
        android:checkable="true"
        app:showAsAction="never" />
</menu>

```

Листинг Б.4 – alarm_item.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/alarm_item_row"
    android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:paddingTop="@dimen/alarmitemview_padding_top"
        android:paddingBottom="@dimen/alarmitemview_padding_bottom">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/alarm_item"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="@dimen/alarmlistview_item_margin_left">

            <TextView
                android:id="@+id/alarm_item_time"
                android:layout_width="@dimen/alarmlistview_alarmtime_width"
                android:layout_height="match_parent"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintLeft_toLeftOf="parent"
                android:text=""
                android:textSize="@dimen/alarmlistview_alarmtime_font_size"
                android:textColor="@color/colorGray"
                android:textStyle="bold"
                android:gravity="left"/>

            <Switch
                android:id="@+id/alarm_item_switch_alarmonoff"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:scaleX="1.5"
                android:scaleY="1.5"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintLeft_toRightOf="@+id/alarm_item_time"

            android:layout_marginLeft="@dimen/alarmlistview_alarmonoff_margin_left"/>

            <ImageButton
                android:id="@+id/alarm_item_delete"
                android:layout_width="@dimen/alarmlistview_delete_width"
                android:layout_height="@dimen/alarmlistview_delete_height"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintRight_toRightOf="parent"

```

```

        android:layout_marginRight="@dimen/alarmlistview_delete_margin_right"
            android:src="@android:drawable/ic_menu_delete"/>

    </androidx.constraintlayout.widget.ConstraintLayout>

</LinearLayout>

```

Листинг Б.5 – *alarm_dialog.xml*

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="500dp"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/alarmdialog_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        android:text=""
        android:textSize="@dimen/alarmdialog_font_size"
        android:textStyle="bold"
        android:textColor="@color/colorBlack"/>

    <EditText
        android:id="@+id/alarmdialog_editttext"
        android:layout_width="@dimen/alarmdialog_button_width"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/alarmdialog_text"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginTop="@dimen/editalarm_label_margin_top"
        android:textSize="@dimen/alarmdialog_font_size"
        android:inputType="number"
        android:ems="10"
        android:gravity="center"/>

```

```

<Button
    android:id="@+id/button_alarmdialog_ok"
    android:layout_width="@dimen/alarmdialog_button_width"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/alarmdialog_edittext"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="@dimen/editalarm_label_margin_top"
    android:text="OK"
    android:textSize="@dimen/alarmdialog_font_size"
    android:textColor="@color/colorBlack"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Листинг Б.6 – alarm_edit.xml

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/label_alarmtime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="@+id/button_alarmtime"
        app:layout_constraintBottom_toBottomOf="@+id/button_alarmtime"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="@dimen/editalarm_label_margin_left"
        android:text="@string/label_alarmtime"
        android:textSize="@dimen/editalarm_label_font_size"
        android:gravity="left"
        android:textColor="@color/colorBlack"/>

    <Button
        android:id="@+id/button_alarmtime"
        android:layout_width="@dimen/editalarm_button_width"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"

```

```

        app:layout_constraintLeft_toRightOf="@+id/label_alarmtime"
        android:layout_marginTop="@dimen/editalarm_label_margin_top"
        android:layout_marginLeft="@dimen/editalarm_button_margin_left"
        android:text=""
        android:textSize="@dimen/editalarm_button_font_size"
        android:textColor="@color/colorRed"/>

<CheckBox
    android:id="@+id/checkBox_alarmReplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/button_alarmtime"
    app:layout_constraintLeft_toLeftOf="@+id/label_alarmtime"
    android:layout_marginTop="@dimen/editalarm_label_margin_top"
    android:text="@string/label_alarmreplay"
    android:textSize="@dimen/editalarm_label_font_size"
    android:textColor="@color/colorBlack"/>

<TextView
    android:id="@+id/label_alarmonoff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toTopOf="@+id/switch_alarmonoff"
    app:layout_constraintBottom_toBottomOf="@+id/switch_alarmonoff"
    app:layout_constraintLeft_toLeftOf="@+id/label_alarmtime"
    android:text="@string/label_alarmonoff"
    android:textSize="@dimen/editalarm_label_font_size"
    android:textColor="@color/colorBlack"/>

<Switch
    android:id="@+id/switch_alarmonoff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:scaleX="1.5"
    android:scaleY="1.5"
    app:layout_constraintTop_toBottomOf="@+id/checkBox_alarmReplay"
    app:layout_constraintLeft_toLeftOf="@+id/button_alarmtime"
    android:layout_marginTop="@dimen/editalarm_label_margin_top"/>

<TextView
    android:id="@+id/label_alarmmelody"
    android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="@+id/button_alarmmelody"
        app:layout_constraintBottom_toBottomOf="@+id/button_alarmmelody"
        app:layout_constraintLeft_toLeftOf="@+id/label_alarmtime"
        android:text="@string/label_alarmmelody"
        android:textSize="@dimen/editalarm_label_font_size"
        android:textColor="@color/colorBlack"/>

<Button
    android:id="@+id/button_alarmmelody"
    android:layout_width="@dimen/editalarm_button_width"
    android:layout_height="@dimen/editalarm_button_height"
    app:layout_constraintTop_toBottomOf="@+id/switch_alarmonoff"
    app:layout_constraintLeft_toLeftOf="@+id/button_alarmtime"
    android:layout_marginTop="@dimen/editalarm_label_margin_top"
    android:text=""
    android:textSize="@dimen/editalarm_button_font_size"
    android:textColor="@color/colorRed"/>

<ImageButton
    android:id="@+id/button_alarmmelody_play"
    android:layout_width="@dimen/editalarm_play_button_width"
    android:layout_height="wrap_content"
    android:src="@android:drawable/ic_media_play"
    android:layout_marginLeft="@dimen/editalarmplay_button_margin_left"
    android:text=""
    app:layout_constraintBottom_toBottomOf="@+id/button_alarmmelody"
    app:layout_constraintLeft_toRightOf="@+id/button_alarmmelody"
    app:layout_constraintTop_toTopOf="@+id/button_alarmmelody" />

<ImageButton
    android:id="@+id/button_alarmmelody_stop"
    android:layout_width="@dimen/editalarm_play_button_width"
    android:layout_height="wrap_content"
    android:layout_marginLeft="@dimen/editalarmplay_button_margin_left"
    android:src="@android:drawable/ic_media_pause"
    android:text=""
    app:layout_constraintBottom_toBottomOf="@+id/button_alarmmelody"
    app:layout_constraintLeft_toRightOf="@+id/button_alarmmelody"
    app:layout_constraintTop_toTopOf="@+id/button_alarmmelody" />
</androidx.constraintlayout.widget.ConstraintLayout>

```