**Государственное общеобразовательное учреждение Республики Коми**

**“Физико-математический лицей интернат”**

**Приложение для Android. Ежедневник-планировщик.**

Автор: Мельников Игорь Андреевич, 10Б класс

Руководитель: Зимина А. С., учитель информатики,

ГОУ РК “ФМЛИ”

Сыктывкар

2023 г.

**Оглавление**

[**Введение**](#_6agr01xv0zfn) **3**

[**Глава 1. Теоретическая часть**](#_ismny2xjx8ac) **6**

[1.1 Язык программирования Kotlin](#_7umgemm4h3tw) 6

[1.2 Среда разработки Android Studio](#_yvs474irjx8w) 6

[1.3 Интерфейс](#_7w3a8ano1g23) 7

[1.4 Система контроля версий](#_ep8uxhwyut97) 7

[1.5 База данных Sqlite](#_gximf8hdbd0g) 7

[1.6 Сборка приложения](#_ueum1z40lbxv) 7

[**Глава 2. Практическая часть**](#_bautp77qaujk) **8**

[2.1 Изучение языка программирования Kotlin](#_3d6msnvfcjoh) 8

[2.2 Изучение среды разработки Android Studio](#_2dtrk1qayb9p) 8

[2.3 Система контроля версий GitHub](#_a3vf6mfxecmt) 8

[2.4 План работы приложения](#_5rfpyagrdpla) 9

[2.5 Разработка дизайна](#_c2q1zmn2c7r5) 9

[2.6 Логика приложения](#_87vizv554vlm) 12

[2.7 Размещение в публичный доступ](#_7px0mezfuqg) 13

[**Заключение**](#_ebprd185pbdt) **14**

[**Список литературы**](#_1dfrjmnsqx5o) **15**

[**Приложения**](#_ojz0fe5r8qom) **16**

# **Введение**

В наше время смартфоны являются важной частью жизни человека, в нем хранятся его личные данные, с помощью него человек всегда остается на связи, поэтому телефон всегда рядом со своим обладателем. Кроме того, одной из тенденций настоящего является постоянно возрастающий темп жизни, поэтому нам приходится очень тщательно следить за временем.

Если постоянно держать в своей голове массу разнообразной информации и различных задач, то часто можно попросту забыть обо всем на свете и запутаться в собственных устремлениях. Множество людей весьма поверхностно представляет себе планы на следующий день, неделю, месяц, и в результате эффективность их действий оставляет желать лучшего. Время проходит, а какие-либо значимые успехи в различных областях жизни так и не наблюдаются. Бывает и так, что вновь возникшая идея путает старые планы, и не дает до конца довести начатое дело. Стоит признать, что с подобными проблемами часто сталкивается большинство современных людей. А чтобы такого не происходило, необходимо систематизировать получаемые знания и начать постоянно планировать свою жизнь. В этом поможет такой практичный инструмент, как ежедневник.

**Актуальность:** Благодаря записи различной поступающей информации, она лучше откладывается и дольше запоминается у вас в голове. Просматривание списка дел на день дает возможность не отвлекаться на лишнее и не тратить силы впустую, что позволяет сэкономить массу энергии и вашего личного времени.

**Тип проекта** - Практико-ориентированный

**Цель:** сделать приложение для android, в которое можно будет записывать свое расписание дел, а также напоминать про ближайшие активности.

**Задачи:**

1. Выбрать и изучить язык программирования, на котором будет написано приложение.
2. Выбрать и изучить среду разработки.
3. Разработать приложение.
4. Выпуск приложения в открытый доступ.

**Предполагаемым продуктом** моей работы является приложение для OC Android.

**Ресурсы:**

1. Информационные (Интернет-ресурсы)
2. Материальные (компьютер, телефон)
3. Трудовые (я)

**План работы:**

1. Выбор темы (февраль)
2. Планирование деятельность (февраль)
3. Изучения языка программирования Kotlin и работу в Android Studio (февраль - апрель)
4. Создать примерный интерфейс приложения (март)
5. Разработка приложения (март - апрель)
6. Создание текстовой части проекта (апрель)
7. Презентация проекта (апрель)
8. Рефлексия (апрель)

Чтобы разобраться в теме я использовал материалы по изучению языка Kotlin (видеоролики на Youtube, статьи различных специалистов, бесплатные курсы), среды разработки Android Studio (видеоролики на Youtube, статьи в Интернете), а также такие форумы как, StackOverflow, Cyberforum, Habr.

**Практическая значимость** проекта заключается в том, что мое приложение может помочь лучше планировать свое время.

# Глава 1. Теоретическая часть

## **1.1 Язык программирования Kotlin**

**Kotlin** - [статически типизированный](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), объектно-ориентированный [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), работающий поверх [Java Virtual Machine](https://ru.wikipedia.org/wiki/Java_Virtual_Machine) (форматирует Java код в байт-код) и разрабатываемый компанией [JetBrains](https://ru.wikipedia.org/wiki/JetBrains). Также компилируется в [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) и в [исполняемый код](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D0%BD%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) ряда платформ через инфраструктуру [LLVM](https://ru.wikipedia.org/wiki/LLVM) (набор компиляторов для высокоуровневых языков программирования) .

Авторы ставили целью создать язык более лаконичный и типобезопасный, чем [Java](https://ru.wikipedia.org/wiki/Java), и более простой, чем [Scala](https://ru.wikipedia.org/wiki/Scala_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Следствием упрощения по сравнению со Scala стали также более быстрая [компиляция](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и лучшая поддержка языка в [ID](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)E. Язык полностью совместим с Java, что позволяет Java-разработчикам постепенно перейти к его использованию; в частности, язык также встраивается [Android](https://ru.wikipedia.org/wiki/Android), что позволяет для существующего Android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком.

## **1.2 Среда разработки Android Studio**

**Android Studio** – интегрированная среда разработки для работы с платформой Android. Основана на программном обеспечении IntelliJ IDEA от компании JetBrains. Официальное средство разработки Android приложений, главная причина, по которой был выбран именно Android Studio

## 1.3 Интерфейс

**XML -** Расширяемый язык разметки, подобный HTML, применяемый для создания интерфейса приложения.

## 1.4 Система контроля версий

**GitHub** - веб-сервис для хостинга IT-проектов и их совместной разработки

## 1.5 База данных Sqlite

**Sqlite** - встраиваемая СУБД, реализующая легковесную дисковую базу данных, не требующую отдельного серверного процесса и позволяющую получить доступ к БД с использованием языка запросов SQL.

## 1.6 Сборка приложения

**Gradle** — [система автоматической сборки](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8), построенная на принципах [Apache Ant](https://ru.wikipedia.org/wiki/Apache_Ant) и [Apache Maven](https://ru.wikipedia.org/wiki/Apache_Maven), но предоставляющая [DSL](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) на языках [Groovy](https://ru.wikipedia.org/wiki/Groovy) и [Kotlin](https://ru.wikipedia.org/wiki/Kotlin).

# Глава 2. Практическая часть

## 2.1 Изучение языка программирования Kotlin

Для разработки каких-либо программ в Android Studio необходимо знание языков программирования Kotlin или Java. Выбирая между двумя языками, мною были просмотрены некоторые статьи с сайтов по IT и ролики на youtube, и я остановился на языке Kotlin, так как большая часть программистов говорили, что этот язык более лаконичный, удобный и современный.

В изучении языка я бесплатные курсы с youtube (см. список литературы 1, 2(бд)), также официальную документацию языка Kotlin (см. список литературы 3).

## 2.2 Изучение среды разработки Android Studio

Для обучения с работой в Android Studio я смотрел статьи из интернета и официальную документацию (см. список литературы 4), а также для закрепления материала создавал крайне простые интерфейсы.

## 2.3 Система контроля версий GitHub

В создании своего проекта я использовал систему контроля версий GitHub, для удобства работы с ней я использовал официальное десктопное приложение - GitHub Desktop.

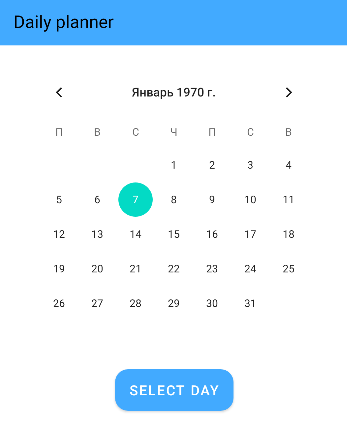
## 

## 2.4 План работы приложения

В первую очередь надо было придумать, как пользователь будет взаимодействовать с приложением. Было решено сделать 3 экрана.

1. **Главный экран** - пользователь выбирает день, в который он хочет внести свои дела.
2. **Экран работы с делами** - пользователь может добавить какое-либо дело, нажав на кнопку с иконкой плюс или посмотреть расписание, уже занесенных им дел.
3. **Редактирование или добавление записи** - пользователь добавляет занятость или редактирует её.

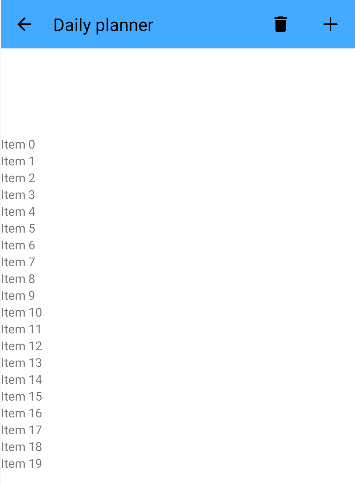
## 2.5 Разработка дизайна

Следующим шагом была разработка дизайна приложения. Он состоит из экранов Android Layout. Весь дизайн был создан с помощью инструментов Android Studio.

**1) Главный экран**

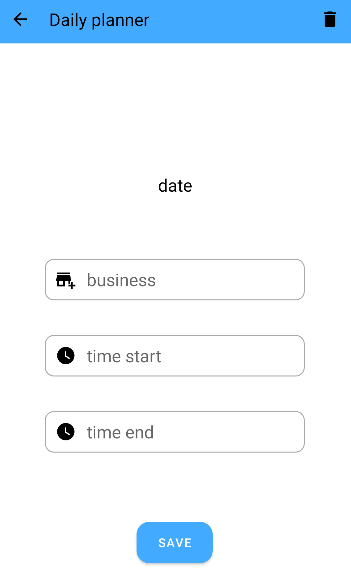
На этом экран с помощью инструмента CalendarView пользователь выбирает интересующий его день.

2) **Экран работы с делами**

После выбора дня пользователь видит этот экран. 

Для отображения дневных задач был использован объект RecyclerView, представляющий собой список элементов CardView (см.приложение). Пользователь может добавить новую занятость, нажав на кнопку с иконкой плюс, удалить все дневные активности, нажав на кнопку с иконкой мусорки или вернуться на главное меню нажав на кнопку с иконкой стрелочки назад.

3) **Экран редактирования записи**



Экран состоит из 3 полей EditText, в которые пользователь вводит название дела, время начала, время окончания, кнопки с текстом save, при нажатии на которую, запись сохраняется в базу данных и этот экран закрывается. Если в экране работы с делами пользователь нажмет на один из элементов RecyclerView, то будет видна кнопка с иконкой мусорки, удаляющей выбранную активность.

## 2.6 Логика приложения

Мое приложение состоит из трех Activity

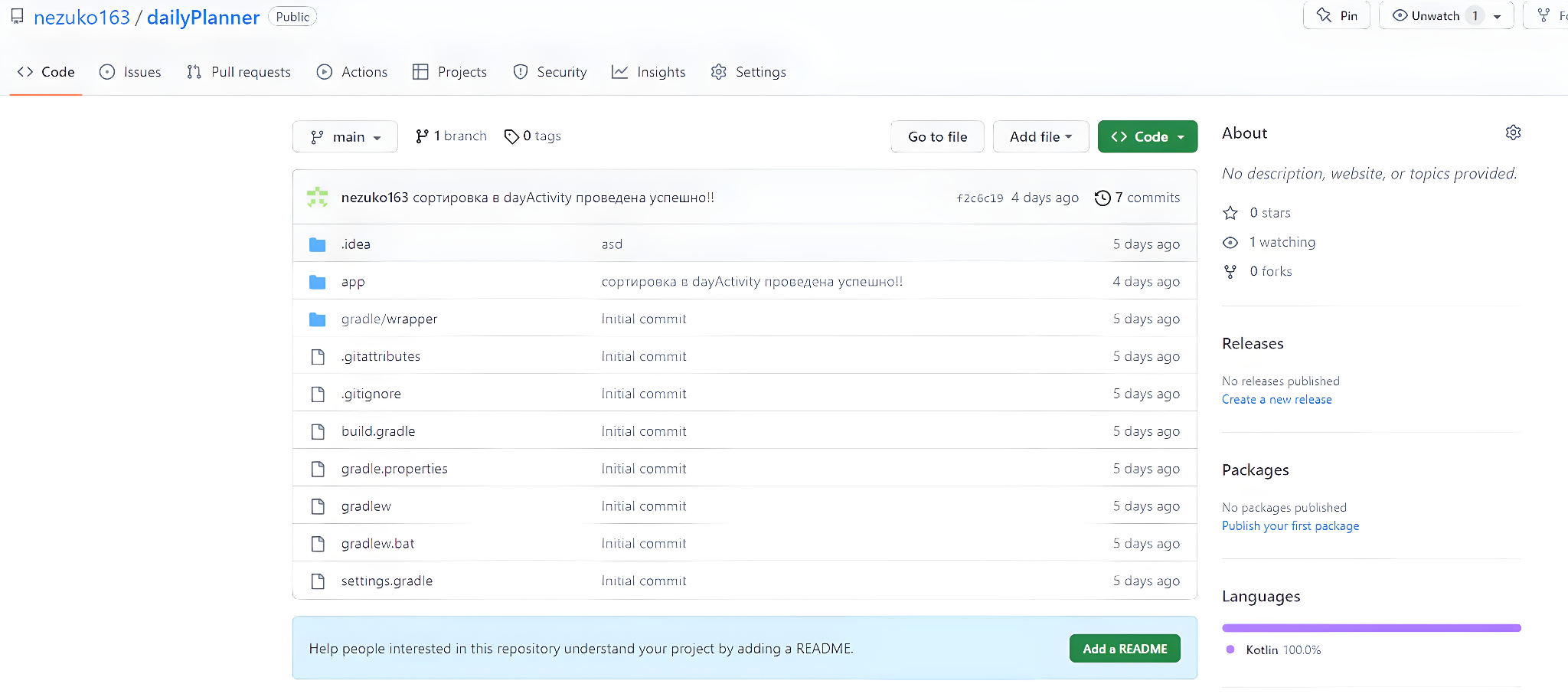
1) MainActivity - привязана к главному экрану, обрабатывает выбор дня и переходит в DayActivity.

2) DayActivity - привязана к экрану с делами, выводит список дел на выбранный день, обрабатывает нажится на кнопку с иконкой плюс (переход на следующее Activity AddNewThing), на кнопку с иконкой мусорки (удаление занятий в выбранном дне), на элемент RecyclerView (переход на следующее Activity AddNewThing), также принимает коллбек с AddNewThing и записывает новую активность в базу данных.

3) AddNewThing - привязана к экрану редактрования, считывает данные с EditText’ов и при нажатии на кнопку с текстом save или на кнопку с иконкой мусорки отправляет коллбек в DayActivity с добавлением / редактированием / удалением записи.

## 2.7 Размещение в публичный доступ

Весь готовый проект, включая этот документ можно найти в моём профиле на гитхаб. Название - dailyPlanner.



https://github.com/nezuko163/dailyPlanner

# Заключение

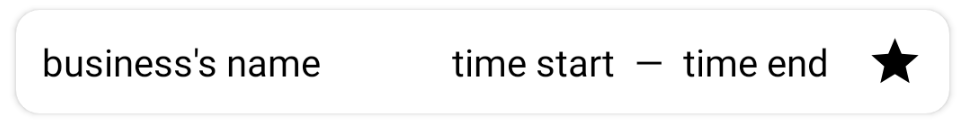
В результате своей работы я изучил основы языка программирования Kotlin, среду разработки Android Studio. Поставленная цель была достигнута. Полученное приложение является простым в обращении, не требует подключения к интернету. Скачать приложение можно с гитхаба, ознакомившись с файлом readme.

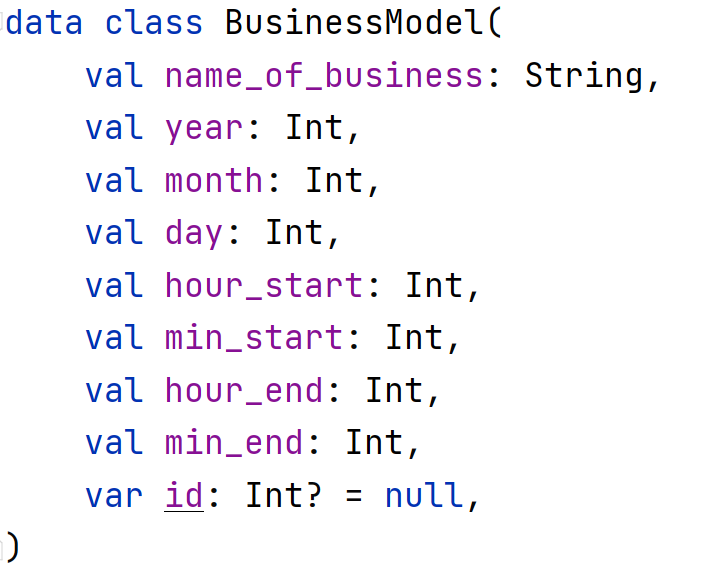
# Список литературы

1. Курсы для изучения языка Kotlin. <https://www.youtube.com/playlist?list=PLmjT2NFTgg1fdHN-9Wn4XYr-IOuadxMm5>
2. Использование sqlite в Kotlin. <https://www.youtube.com/playlist?list=PLSrm9z4zp4mGK0g_0_jxYGgg3os9tqRUQ>
3. Официальная документация Kotlin. <https://kotlinlang.org/>
4. Официальная документация Android Studio. <https://developer.android.com/studio>
5. Первые шаги с GitHub. <https://habr.com/ru/companies/yandex_praktikum/articles/700708/>
6. StackOverflow. <https://stackoverflow.com/>
7. GeeksForGeeks. <https://www.geeksforgeeks.org/>
8. RecyclerView. <https://habr.com/ru/articles/258195/>
9. Памятка по Sqlite. <https://habr.com/ru/articles/564390/>
10. Обучение языку Kotlin. <https://developer.alexanderklimov.ru/android/kotlin/>

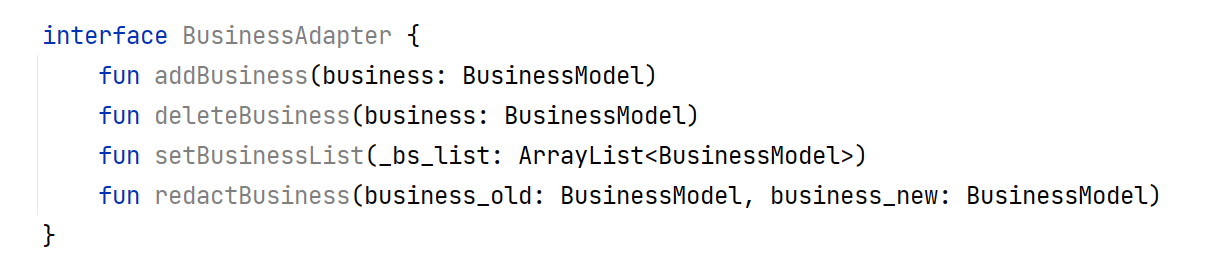
# Приложения

Это ConstraintLayout, состоящий из CardView, использовающийся как элемент RecyclerView, в который вместо заданных текстовых значений подставляются значения, которые ввёл пользователь





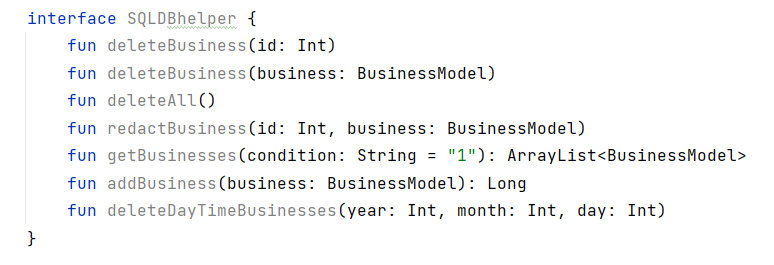
Интерфейс класса BusinessAdapter



Список дел отображается на элементе RecyclerView отображается с помощью специального наследуемого класса BusinessAdapter, который взаимодействует с айтемами RecyclerView. В LayoutManager RecyclerView сообщается данного класса, таким образом происходит привязка.



Для удобства работы с базой данных я создал отдельный класс, вот его интерфейс.



Файл DayActivity. В нём создаётся лаунчер, который будет запускать AddNewThing активити, а когда пользователь завершит работу с ним (нажмёт на кнопку save или удалить запись), то в это активити передасться результат, с нужной инструкцией.

