



Android

разработка

Старт марафона 3.03.2022 12:00

Добро пожаловать на курс!



Table of Contents

Правила и порядок участия:	3
1. Introduction	4
2. Classes	5
3. Conventions	6
4. Collections	7
5. Properties	8
6. Builders	9
7. Generics	10
8. Kotlin Coroutine	11
9. TableView	13
10. Activity and Layouts	14
11. Save States and used menu, live data, view model ..	15
12. Fragments, fragment-lifecycle and navigation	16
13. Styles, xml, RecyclerView, WebView	18
14. Многопоточность	19
15. Room, SQLite, Content provider	20
16. Диалоговые окна, intents	21
17. Пользовательские представления и события касания ..	23
18. Работа с сетью, REST API, json	24
19. WorkManager.Service	25
20. Ознакомление с RxJava, RxKotlin	26
Тестовое задание по желанию	28
Ресурсы	29



Правила и порядок участия:

Каждое задание заливаем на GITLAB.

1 задание = 1 репозиторий (репозиторий должен быть открытым) .

Далее заполняем форму, куда вставляем ссылку на GitLab:

[Ссылка на форму](#)

Сдача задания вовремя: 5 баллов

Сдача задания позже срока: 2 балла

+ выставление баллов от 0 до 5 за каждое задание в зависимости от качества кода.

Окончание марафона – 4.04.2022

Подведение итогов и награждение – 18.04.2022



1. Introduction

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Introduction» в [Kotlin Koans](#)

Deadline - 4.03.2022 23:59



2. Classes

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Classes» в [Kotlin Koans](#)

Deadline - 6.03.2022 12:00



3. Conventions

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Conventions» в [Kotlin](#)
[Koans](#)

Deadline - 7.03.2022 23:59



4. Collections

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Collections» в [Kotlin Koans](#)

Deadline - 9.03.2022 12:00



5. Properties

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти весь блок «Properties» в [Kotlin Koans](#)

Deadline - 10.03.2022 23:59



6. Builders

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Introduction» в [Kotlin Koans](#)

Deadline - 12.03.2022 12:00



7. Generics

Ссылка на видео: [How to Kotlin](#)

Домашнее задание: - пройти блок «Introduction» в [Kotlin Koans](#)

Deadline - 13.03.2022 23:59



8. Kotlin Coroutine

Ссылка на видео: [Корутины в Kotlin](#)

Домашнее задание:

I. Сделать программу, которая запускает в фоновом потоке метод печати «World» с задержкой в 1 сек. В основном потоке запускаем печать «Hello,» с задержкой в 2 сек., блокируем поток.

II. Написать две функции с задержкой, которые будут возвращать 2 числа. В main написать блок кода, который будет суммировать вызов этих 2 функций, сначала написать последовательный вызов функций, вывести сумму и прева работы, потом написать блок с асинхронным вызовом и сравнить время работы, обосновать время.

III. Дополнить код, чтоб программа выводила след текст в консоль.

```
////////////////////////////////////  
«I'm sleeping 0 ...  
I'm sleeping 1 ...  
I'm sleeping 2 ...  
main: I'm tired of waiting!  
I'm running finally  
main: Now I can quit.»  
////////////////////////////////////
```



```
fun main(args: Array<String>) = runBlocking<Unit> {  
    val job = launch {  
        try {  
            repeat() { i ->  
  
                }  
        } finally {  
  
        }  
    }  
}
```

Deadline - 15.03.2022 12:00



9. TableView

Ссылка на видео: [Android App Development Tutorial for Beginners](#)

Домашнее задание: создать приложение, в котором будет `imageView` и `button`. При нажатии на кнопку будет появляться случайная картинка на `imageView`, 4-7 любых картинок можно скачать с интернета.

Подсказка: обработку изображений можно сделать в виде `switch case` (when в kotlin), изображения лучше всего поместить в `drawable` или `res`, откуда производить загрузку изображения с изменением `background imageView`. Для изображений использовать функцию `random`.

Дополнительно: добавить рандомное изменение цвета кнопки, то есть её `rgb` компонент.

Deadline - 16.03.2022 23:59



10. Activity and Layouts

Ссылки на видео:

1. [Intents and starting activities](#)
2. [Создание Activity, передача данных, возврат](#)

Домашнее задание: Создать 3 экрана

1 экран `SplashActivity`: экран с какой-то картинкой-заставкой в 3 секунды, реализовать с помощью `FrameLayout`.

2 экран `MainActivity`: с 5 `edit textview`: фамилия, имя, отчество, возраст и описание хобби человека, реализовать с применением `ConstraintLayout`.

3 экран `InfoActivity`: в зависимости от возраста человека выводить различный текст, в который подставляются его данные, реализовать с помощью `ScrollView`, переход на экран 3 осуществляется по нажатию с экрана 2.

Deadline - 18.03.2022 12:00



11. Save States and used menu, live data, view model

Ссылки на видео:

1. [Saving data with Shared Preferences](#)
2. [Menu, Toolbar в Android](#)
3. [The ViewModel & LiveData](#)

Домашнее задание: Использовать приложение из урока 10. Добавить возможность сохранения состояния вводимых полей при повороте экрана. Сделать кнопку меню в ActionBar. С элементами выбора:

- 1) Сохранить состояния вводимых полей (кэш) с помощью sharedpreference
- 2) Изменение background ConstraintLayout на рандомную фотографию, похожее задание делалось в 9 уроке.
- 3) Возможность перейти экран, с кнопкой и текстовым полем, то есть создать ещё 1 activity(добавить к текущему проекту). Реализовать с помощью Live Data и ViewModel.

Deadline - 19.03.2022 23:59



12. Fragments, fragment-lifecycle and navigation

Ссылки на видео:

1. [Fragment](#)

2. [Navigation Component](#)

Домашнее задание:

1. Сделать 4 фрагмента

1 фрагмент — LoginFragmet

2 фрагмент — RegistrationFragment

3 фрагмент — HomeFragment

4 фрагмент — WebFragment

2. Сделать вёрстку в activity. Сделать Frame под фрагмент. И добавить BottomNavigationView

В папке menu сделать xml файл с 3 items для нашей навигации между фрагментами.

В папке navigation создать xml файл navigation_controller, в нём добавить возможность навигации между LoginFragmet и RegistrationFragment, должны быть зациклины, от LoginFragmet сделать навигацию к фрагменту HomeFragment. После чего должен стать видимым BottomNavigationView. В фрагментах регистрации и логина сделать кнопки для переходов, в логине у вас будет 2 кнопки.



Deadline - 21.03.2022 12:00



13. Styles, xml, RecyclerView, WebView

Ссылки на видео:

1. [RecyclerView](#)
2. [WebView](#)
3. [Styles and Themes](#)

Домашнее задание: Использовать приложение с предыдущего урока.

В HomeFragment добавить RecyclerView, выведет в него список из строк. Добавить возможность с какой-то периодичностью менять стиль у элементов view. Стили создать на своё усмотрение, основные параметры: шрифт, размер и цвет.

В фрагмент Web добавить WebView и вывести туда страницу поискового запроса google. Добавить возможность сохранить url пользователя при заходе. Так добавить возможность навигации по сайтам, которые пользователь посетил, то есть переопределить кнопку назад и возвращать на предыдущие страницы.

Deadline - 22.03.2022 23:59



14. Многопоточность

Ссылки на видео:

[1. Многопоточность](#)

[2. Getting started with Firebase on Android](#)

Домашнее задание: Использовать предыдущий проект урока, сделать авторизацию для пользователя, сделать какой-то заранее известный данные, можно допустим сохранить локально в json, а потом доставить оттуда данные, и сверять, при регистрации добавлять туда данные, для более продвинутых можно изучить инструмент firebase, тогда будет лучше проект. При регистрации запросить данные, допустим имя, номер, дату рождения, пол (на своё усмотрение).

Дополнительно: реализовать логику запоминания токена пользователя, чтоб ему не приходиться авторизацию каждый раз.

Допустим можно хранить его локально.

Реализовать фрагмент профиля пользователя и выход с аккаунта.

Deadline – 24.02.2022 12:00



15. Room, SQLite, Content provider

Ссылка на видео:

[1.Room](#)

[2.Content Provider](#)

Домашнее задание: Реализовать заметки. Сделать приложение, которое будет с помощью Content provider получать контакты пользователей на телефоне, добавить возможность добавлять заметки по каждому контакту, заметки сохранять в room, добавить возможность их редактировать и удалять. Проработать момент, если контакт был удалён, чтобы данные обновлялись при запуске в room.

Deadline - 25.03.2022 23:59



16. Диалоговые окна, intents

Ссылка на видео:

[1. Dialog Fragment](#)

[2. Intents](#)

Домашнее задание: Сверстать экран, на котором будет текст преступления и кнопки для изменения вызова диалоговых окон. Первая кнопка для изменения времени, вторая для изменения даты.

Схему взаимодействия см. рис. 1-2

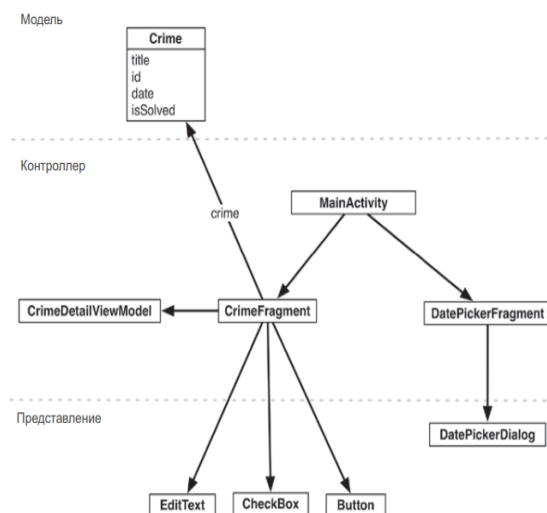


Рисунок 1 – Диаграмма объектов для фрагментов с хостом MainActivity



Рисунок 2 – Взаимодействие между CrimeFragment и DatePickerFragment

Deadline – 27.03.2022 12:00



17. Пользовательские представления и события касания

Ссылка на видео: [AvatarImageView](#)

Домашнее задание: заверстать экран с кастомным view элементом. На котором можно возможность рисовать различные фигуры, их перетаскивать и очищать холст. Сделать возможность рандомного появления цвета у элементов.

Deadline - 28.03.2022 23:59



18. Работа с сетью, REST API, json

Ссылка на видео: [KOTLIN Retrofit Tutorial](#)

Ссылка на доп материалы:

1. <https://jsonplaceholder.typicode.com/>
2. <https://jsonplaceholder.typicode.com/posts>

Домашнее задание: сделать список RecyclerView со списком сообщений, в списке должен быть title и номер сообщения. Сделать возможность нажатия элементов, чтоб открывалось окно редактирования сообщения и title, т. е. Либо просмотр, либо изменение полей и с последующим вызовом post запроса, если применить изменения. Использовать Retrofit или Okhttp для REST API.

Deadline - 30.03.2022 12:00



19. WorkManager.Service

Ссылки на видео:

1. [WorkManager](#)

2. [SERVICES](#)

Домашнее задание: Сделать приложение с REST API, добавить сетевые запросы в service и выводить в консоль json, добавить запросы в WorkManager, сделать там задержку, проверит , что всё работает, при свёрнутом приложении.

На экран добавить кнопку и RecyclerView, при нажатии кнопки добавить возможность отобразить данные.

Deadline - 31.03.2022 23:59



20. Ознакомление с RxJava, RxKotlin

Ссылка на видео: [RxJava](#)

Домашнее задание:

Как создать Observable?

Как обернуть обычный код в реактивный, так, чтобы можно было подписаться на Observable?

Заверстать простое приложение на 2 фрагментах. В 1 фрагменте создать TextView которое будет выводить текст и кнопку start, которая будет отвечать за подписку observer на observable. Во втором фрагменте сделать 2 кнопки, первая кнопка будет отвечать за подрузку текста по Rest API, 2 кнопка будет забирать рандомные данные с json файла, который будет хранится папке res,value. Json можете сформировать как угодно, оставить поле id, для идентификации строки. Также добавить textview, которое будет говорить нам о успешной выгрузке данных или неудачной. Во втором фрагменте создать observable с нашими данными. В 1 фрагменте написать observer, который будет подписан на наш observable. В observer сделать задержку в 0.5 сек на методе onNext() По истечению всех событий будем писать в textview, что данные иссякли или ошибку.



Наблюдатель должен уведомить подписчика, что данные изменились и в `textView` должны отобразиться новые данные уже. Навигацию фрагментов можно настроить через `BottomNavigationView`.

Для `RestApi` использовать ссылку из урока про `RestApi`.

Deadline - 2.04.2022 12:00



Тестовое задание по желанию

1. Ссылка на дизайн
2. Ссылка для REST API – [ТЫК](#)

Домашнее задание: Лёгкое тестовое задание на Rest API и применение фрагментов, и сохранением state при повороте экрана.

Студенты могут закрепить свои знания + применить различный стек технологий на своё усмотрение.

Deadline – 3.04.2022 23:59



Ресурсы:

Книга:

<https://swiftbook.ru/content/languageguide/closures/>

<https://www.raywenderlich.com/>

<https://www.hackingwithswift.com>