Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Институт математики и информационных систем

Факультет компьютерных и физико-математических наук

Кафедра прикладной математики и информатики

**О Т Ч Е Т**

о прохождении учебной практики

на базе кафедры прикладной математики и информатики

ФГБОУ ВО «Вятcкий государственный университет»

(наименование базы практики)

в период с \_08.02.2021\_ по **\_**02.05.2021**\_**

Выполнил

студент гр. ФИб-3302-51-00 Изображение выглядит как очки

Автоматически созданное описание Савин Д. А. \_\_\_\_\_\_\_\_\_\_

(шифр) (подпись) (Ф. И. О.) (дата)

Руководитель практики от вуза

к.ф.-м.н, доцент\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ Шалагинова Н.В. \_\_\_\_\_\_\_\_\_\_

(уч. степень, должность) (подпись) (Ф. И. О.) (дата)

Практика защищена

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Киров 2021

**Содержание**

[Введение 3](#_Toc70623715)

[Описание индивидуального задания 4](#_Toc70623716)

[Описание разработанного приложения 5](#_Toc70623717)

[Заключение 6](#_Toc70623718)

[Список литературы 7](#_Toc70623719)

[Приложение 1. Листинг программы 8](#_Toc70623720)

# Введение

Учебная практика проходила в ФГБОУ ВО «Вятский государственный университет», на базе кафедры фундаментальной информатики и прикладной математики факультета компьютерных и физико-математических наук с 08.02.2021 г. по 02.05.2021 г., количество недель: 12. Установочная конференция состоялась 08.02.2021 г.

Целью учебной практики является закрепление и углубление теоретической подготовки, приобретение практических навыков и компетенций в сфере профессиональной деятельности, в основном путём самостоятельного решения предусмотренного программой индивидуального задания.

Форма проведения учебной практики: компьютерная (разработка мобильного приложения).

В ходе прохождения практики необходимо реализовать мобильное приложение «VoiceAssistent» для операционной системы Android и дополнить его новыми возможностями, используя данные web-ресурсов.

Язык программирования: kotlin.

# Описание индивидуального задания

1. Постановка задачи. Описание требований к функционалу приложения.
2. Описание выбранных сетевых ресурсов (api для получения данных, структуры html-страницы для парсинга).
3. Ссылка на систему контроля версий Git.

# Описание разработанного приложения

1. Руководство пользователя (описание запросов к голосовому помощнику).

|  |  |
| --- | --- |
| Запрос Пользователя | Ответ Помощника/Варианты ответа |
| Привет, Hi, Hello, Здравствуйте | Привет, Здравствуйте |
| Что делаешь? | Отвечаю на вопросы :), Ожидаю. |
| Как дела? | Отлично :), Нормально. |
| Который час?, Время | Выводит текущее время в формате: hh.mm.ss |
| Какой день недели? | Выводит текущий день недели (Понедельник, Вторник и т. д.) |
| Какой сегодня день? | Выводит цифру – номер дня (Понедельник – 1, Вторник – 2 и т. д.) |
| Сколько дней до зачета? | * + - 1. Если есть сохраненная дата, то выводит количество времени (Год/месяц/дней) до этой даты.       2. Если нет сохраненной даты, то спрашивает: "Дата не найдена. Вы желаете добавить? (Y/N)". Если ответ: "y", то выводит: "Введите дату: дд.мм.гггг". Вводим дату по формату, ответ – количество времени (Год/месяц/дней) до этой даты.   Если ответ: "n", то запрос закрывается: "Хорошо. Закрываю вопрос." |
| Сколько дней до dd.mm.yyyy | Выводит количество времени (Год/месяц/дней) до этой даты. |
| Погода в городе Название города | Выводит количество градусов и информацию о погоде. |
| Число (например, 20) в строку | Переводит число из цифр в слово/слова. |
| Праздник вчера\сегодня\завтра\dd/mm/yyyy | Выводит название праздника в заданный день. |
| Совет | Вывод случайно выбранный совет. |
| Курс криптовалюты | Выводит названия криптовалют и стоимость в долларах. |

1. Описание реализации:
   1. Работа с API проводилась с помощью сериализации, использовалась библиотека Retrofit2. Для каждого API было создано 4 класса (для примера API: <https://api.adviceslip.com/>):

Класс сущности API с определением необходимых тегов

class Advice : Serializable {  
 @JvmField  
 @SerializedName("slip")  
 @Expose  
 var slip: adv? = null  
  
 inner class adv {  
 @JvmField  
 @SerializedName("advice")  
 @Expose  
 var advice: String? = null  
 }  
}

Класс с созданием соединения к сайту с помощью библиотеки Retrofit2:

object AdviceService {  
 @JvmStatic  
 val aPI: AdviceAPI  
 get() {  
 val retrofit = Retrofit.Builder()  
 .baseUrl("https://api.adviceslip.com/")  
 .addConverterFactory(GsonConverterFactory.create())  
 .build()  
 return retrofit.create(AdviceAPI::class.*java*)  
 }  
}

Класс с описанием конкретного Get запроса:

interface AdviceAPI {  
 @get:GET("/advice")  
 val advice: Call<Advice?>?  
}

Класс для непосредственного взаимодействия API и возвращения значений в основной код:

object AdviceToString {  
 @JvmStatic  
 fun getAdvice(number: String?, callback: Consumer<String?>) {  
 val api = *aPI* val call = api.advice  
 val str = *arrayOf*<String?>("-1")  
 call!!.enqueue(object : Callback<Advice?> {  
 override fun onResponse(call: Call<Advice?>, response: Response<Advice?>) {  
 val result = response.body()  
 if (result != null) {  
 if (result.slip!!.advice != null) {  
 str[0] = result.slip!!.advice  
 callback.accept(str[0])  
 } else {  
 str[0] = "result.advice=null"  
 callback.accept(str[0])  
 }  
 *//callback.accept(String.valueOf(newStr));* } else {  
 str[0] = "result = null"  
 callback.accept(str[0])  
 }  
 }  
  
 override fun onFailure(call: Call<Advice?>, t: Throwable) {  
 Log.w("ADVICE", Objects.requireNonNull(t.message))  
 str[0] = "Не могу перевести failure"  
 }  
 })  
 }  
}

* 1. Для парсинга сетевого ресурса была использована библиотека Jsoup, с помощью которой создается соединение с сайтом, и далее с помощью css-запроса возвращаются необходимые элементы.

private const val URL2 = "https://ru.investing.com/crypto/"

@get:Throws(IOException::class)  
val cryptoCurrencyExchangeRate: String  
 get() {  
 val document = Jsoup.connect(URL2).get()  
 val str = StringBuilder()  
 val elements = document.select("tr[i]")  
 for (element in elements) {  
 str.append(element.getElementsByClass("left bold elp name cryptoName first js-currency-name")  
 .select("a[href]").text())  
 .append(" ")  
 .append(element.getElementsByClass("left noWrap elp symb js-currency-symbol").text())  
 .append(" - ")  
 .append(element.getElementsByClass("price js-currency-price").select("a[href]").text())  
 .append("$\n")  
 }  
 return str.toString()  
 }

Диаграмма всех классов

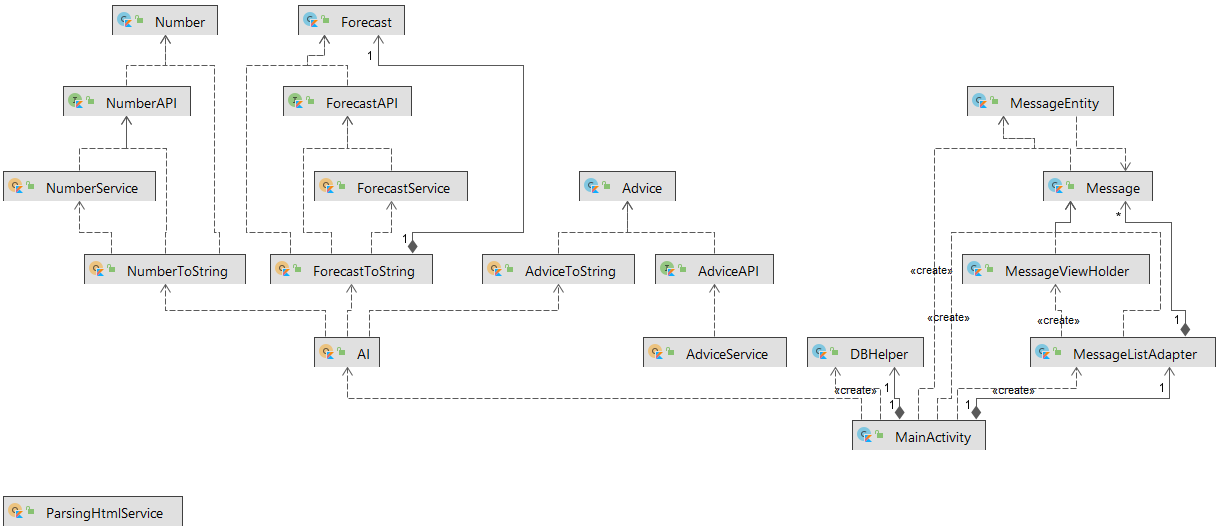


Рисунок 1.

Диаграмма классов API

* 1. описание класса ведения диалога с пользователем (достаточно сигнатур функций класса); описание взаимодействия классов, участвующих в осуществлении диалога (вместе с диаграммой).
  2. методы расширения (если есть);
  3. Все способы сохранения диалога, в зависимости от ситуации.
  4. описание наиболее сложных моментов перехода с языка программирования java на язык программирования kotlin.

1. Тестирование приложения. Как проводилось тестирование – на эмуляторе или на устройстве. Перечислить технические характеристики и версии операционных систем, на которых была проверена работоспособность приложения.

# Заключение

Учебная практика способствовала закреплению и углублению теоретической подготовки, приобретению практических навыков и компетенций в сфере профессиональной деятельности путём реализации всех этапов разработки мобильного приложения.

В ходе практики было реализовано приложение «VoiceAssistent» для операционной системы Android, Версии. Использовался язык программирования kotlin, среда разработки Android Studio. Листинг разработанного приложения приведен в Приложении 1.

Было выбрано API web-ресурса ….

При переносе приложения с языка java на язык kotlin наиболее сложным показалось... потому, что ….

# Список литературы

Приведены примеры оформления источников.

Использовать ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления:

<http://portalus.ru/modules/russianlaw/rus_readme.php?subaction=showfull&id=1434470987&archive=&start_from=&ucat=&>

1. Филлипс Б. Android. Программирование для профессионалов. – Питре, 2019. – 688 с.
2. Android Studio [Электронный ресурс]: URL: <https://developer.android.com/studio/index.html> (дата обращения 01.02.2018)

# Приложение 1. Листинг программы