

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ПО ДИСЦИПЛИНЕ**  
Мобильная разработка под Android

**Код модуля**  
1157339(1)

**Модуль**  
Мобильная разработка под Android

**Екатеринбург**

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Демидова А Д		Руководитель группы	Яндекс
2	Пьянзина Елена Сергеевна	кандидат физико-математических наук, без ученого звания	Доцент	Кафедра теоретической и математической физики

**Согласовано:**

Управление образовательных программ

Ю.Д. Маева

**Авторы:**

- Демидова А Д, Руководитель группы, Яндекс
- Пьянзина Елена Сергеевна, Доцент, Кафедра теоретической и математической физики

**1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Мобильная разработка под Android**

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Практические/семинарские занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Домашняя работа	1

**2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Мобильная разработка под Android**

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ОПК-2 -Способен применять программные средства системного и прикладного характера, в том числе отечественного производства, для решения задач профессиональной деятельности	З-1 - Характеризовать состав, классификацию, особенности функционирования программных средств системного и прикладного назначений П-1 - Иметь практический опыт разработки системного программного обеспечения для решения задач профессиональной деятельности П-2 - Иметь практический опыт разработки прикладного программного обеспечения для решения задач профессиональной деятельности У-1 - Выбирать с учетом задачи и рационально использовать	Домашняя работа Зачет Лекции Практические/семинарские занятия

	функциональные возможности программных средств системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	
ОПК-7 -Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	<p>З-1 - Характеризовать алгоритмические основы программирования на языках общего назначения</p> <p>З-2 - Характеризовать языки программирования общего назначения</p> <p>З-3 - Характеризовать методы, реализуемые в современных инструментальных средствах программирования</p> <p>П-1 - Иметь практический опыт разработки алгоритмов для последующего создания программ на языках общего назначения</p> <p>П-2 - Иметь практический опыт использования типовых инструментальных средств программирования для решения профессиональных задач</p> <p>У-1 - Осуществлять обоснованный выбор способов организации программ и инструментария программирования при решении профессиональных задач</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Лекции</p> <p>Практические/семинарские занятия</p>

### **3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)**

#### **3.1. Процедуры текущей и промежуточной аттестации по дисциплине**

**1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.4**

Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>Работа на занятиях</i>	8,17	100
Весовой коэффициент значимости результатов текущей аттестации по лекциям – <b>0.4</b>		
Промежуточная аттестация по лекциям – <b>зачет</b> Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – <b>0.6</b>		
<b>2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – 0.6</b>		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	8,17	100
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – <b>1</b>		
Промежуточная аттестация по практическим/семинарским занятиям – <b>нет</b> Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – <b>не предусмотрено</b>		
<b>3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – не предусмотрено</b>		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям – <b>не предусмотрено</b>		
Промежуточная аттестация по лабораторным занятиям – <b>нет</b> Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – <b>не предусмотрено</b>		
<b>4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий – не предусмотрено</b>		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям – <b>не предусмотрено</b>		
Промежуточная аттестация по онлайн-занятиям – <b>нет</b> Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – <b>не предусмотрено</b>		

### 3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– **не предусмотрено**

Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – **не предусмотрено**

#### 4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

##### Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

##### Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)			
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания	
		Традиционная характеристика уровня	Качественная характеристика уровня

1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

## 5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

### 5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

#### 5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

#### 5.1.2. Практические/семинарские занятия

Примерный перечень тем

1. Введение
2. Графический интерфейс пользователя
3. Бизнес-логика в приложении
4. Архитектура приложения, паттерны

Примерные задания

Задание 1

- Основные требования.
  - о Сверстать экран редактирования заметки. Создать отдельный класс наследник UIView, который размещается в UIScrollView внутри контроллера.
  - о Верстка должна быть адаптивной: на разных размерах экрана и в разных ориентациях должно выглядеть аккуратно.
- Усложнения 1.
  - о Контент должен скроллиться.
  - о DatePicker должен появляться/исчезать при смене состояния свитчера.
  - о Высота поля для ввода текста заметки должна динамически меняться в зависимости от содержимого (с некоторым минимальным размером).

- o При появлении клавиатуры на экране контент должен оставаться просматриваемым.
- o Выбор цвета должен быть реализован в виде цветных квадратиков с черной рамочкой. Текущий выбранный цвет должен помечаться галочкой. Галочка должна быть отрисована с помощью CoreGraphics.

- Усложнения 2.

- o На основном экране редактирования заметки в секции выбора цвета должен быть дополнительный квадратик для выбора кастомного цвета. Изначально должен выглядеть как палитра цветов.

- o По долгому нажатию на доп. квадратик должен открываться экран с компонентом ColorPicker.

- o ColorPicker обязательно должен быть выполнен в виде отчуждаемого компонента (то есть класс ColorPickerView).

- o Выбор цвета осуществляется путем перемещения пальца по палитре.

- o Элемент указывающий на текущий цвет в палитре в центре должен быть залит в текущий цвет.

- o Элемент отображающий текущий цвет (в левом верхнем углу) должен иметь скругление углов (радиус выбираете сами).

- o Выбранный цвет должен сохраняться при повторном заходе на экран ColorPicker.

- Скриншоты с иллюстрацией желаемого результата:

- o Основные требования: <https://yadi.sk/i/7wNgzGSx3HR4D2>

- o С усложнением 1: <https://yadi.sk/i/CKNjpMo23HR4D8>

- o С усложнением 2: <https://yadi.sk/i/G2KD19G83HR4DY>

- o ColorPicker: <https://yadi.sk/i/w5FR44FO3HR4Dg>

## Задание 2

- Требуется реализовать загрузку и сохранение заметок на сервер.

- o Реализовать все необходимые запросы к бэкенду. Вызывать их через созданные в предыдущей лекции операции. Обработать ошибки. Авторизация через Bearer.

- o Описание бэкенда: <https://yadi.sk/i/yKp7dUYX3JGcMv> .

- o Bearer токены в личные сообщения в Телеграмме.

- Усложнение 1:

- o Реализовать авторизацию через Яндекс.Паспорт и OAuth.

- o Описание API и OAuth Паспорта — <https://tech.yandex.ru/oauth/> .

- o OAuth токен должен быть получен средствами паспорта.

- Усложнение 2:

- o В случае ошибок бэкенда изменяющие операции повторить через определенный промежуток времени до тех пор, пока не выполнится. Обеспечить правильную последовательность повтора.

LMS-платформа – не предусмотрена

## 5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля



Разноуровневое (дифференцированное) обучение.

## Базовый

### 5.2.1. Домашняя работа

Примерный перечень тем

1. Введение. Графический интерфейс пользователя. Бизнес-логика в приложении.

Примерные задания

Раздел I:

Задание 1

- Общие требования.
    - o Реализовать структуру Note (требования ниже).
    - o Реализовать расширение Note для разбора json (требования ниже).
    - o Реализовать протокол Notebook и его реализацию FileNotebook (требования ниже).
    - o Реализовать сохранение и загрузку FileNotebook в файл и из файла.
  - Note.
    - o Иммуutable структура.
    - o Содержит уникальный идентификатор `uid`, если не задан пользователем - генерируется (`UUID().uuidString`).
    - o Содержит обязательные строковые поля — `title` и `content`.
    - o Содержит цвет заметки, пользователь структуры может его не задать, тогда белый по умолчанию (Класс `UIColor` из `UIKit`).
    - o Содержит обязательное поле `importance`, должно быть `enum`, может иметь три варианта — «неважная», «обычная» и «важная».
  - Note, json parsing.
    - o Расширение для структуры Note.
    - o Содержит функцию (`static func parse(json: [String: Any]) -> Note?`) для разбора json.
    - o Содержит вычисляемое свойство (`var json: [String: Any]`) для формирования json'a.
    - o Цвет сохранять в json только, если он не белый.
    - o Не сохранять в json `importance`, если она «обычная».
    - o Не сохранять в json сложные объекты (`UIColor`, `enum`).
  - FileNotebook.
    - o Содержит закрытую для внешнего изменения, но открытую для получения коллекцию Note.
      - o Содержит функции добавления новой заметки.
      - o Содержит функцию удаления заметки (на основе `uid`).
      - o Содержит функцию сохранения всей записной книжки в файл.
      - o Содержит функцию загрузки записной книжки из файла.
- Раздел II:
- Задание 2
- Общие требования.
    - o Создать и настроить проект Notes.
    - o Подключить к нему `CocoaPods` и настроить зависимость.

- o Восстановить работоспособность кода из предыдущего задания, добавить логирование действий.
- o С усложнением 1: провести анализ своего кода на производительность и утечки памяти.
- o С усложнением 2: симулировать проблемы утечек и производительности, продиагностировать и показать решение.
  - Настроить проект.
  - o Необходимо создать проект Notes (+workspace через CocoaPods).
  - o Необходимо настроить сборку Notes, добавить иконки и стартовый экран.
  - o Необходимо настроить дополнительную конфигурацию сборки TestFlight (к Debug и Release).
  - o В схеме настроить логирование выделения памяти и зомби.
    - CocoaPods.
  - o Подключить CocoaPods.
  - o Настроить Podfile.
  - o Добавить зависимость CocoaLumberjack (для Swift).
  - o Попробовать логировать свои действия.
  - Дополнительные требования к сдаче.
  - o Проект на git\*.
  - o Демонстрация работы с Instruments - \*.trace файл и набор скриншотов.

### Задание 3

- Основные требования.
  - o В корне должен быть UINavigationController.
  - o Создать контроллер, отображающий таблицу со списком заметок.
  - o По нажатию на ячейку таблицы нужно переходить на экран редактирования заметки.
  - o По нажатию на «+» в navigation bar нужно переходить на экран редактирования заметки.
    - o Удаление заметки из списка осуществляется свайпом влево и нажатием на «Delete».
    - o Усложнение 1: сделать одно любое дополнительное требование.
    - o Усложнение 2: сделать оба дополнительных требования.
  - Дополнительное требование 1.
    - o Сделать высоту ячеек в таблице зависимой от контента.
    - o Сделать кастомную презентацию color picker.
  - Дополнительное требование 2.
    - o Вместо таблицы использовать UICollectionView.
- Примеры:
  - o Базовая таблица: <https://yadi.sk/i/1gPC60gs3HeoDA>
  - o Custom presentation: <https://yadi.sk/i/fkCUU6up3HeopV>
  - o collectionView: <https://yadi.sk/i/4YTZ1CnS3HepEn>

### Раздел III:

#### Задание 4

- Общие требования.
  - о Реализовать основные операции для приложения, можно использовать стандарт-ные, например, BlockOperation.
  - о Необходимо реализовать ваши операции как асинхронные операции (нельзя использовать стандартные).
  - о Реализовать диспетчер для управления порядком выполнения операций (разные очереди).
- Базовая задача.
  - о Операции для работы с локальным кэшем (использовать FileNotebook из первых задания): загрузить заметку(и) в кэш, загрузить заметку(и) из кэша, удалить все заметки в кэше.
  - о Операции для работы с сетью (пока без реализации на тестовых данных): загрузить с запросом к API, сохранить заметку с запросом к API, удалить заметку с за-просом к API.
  - о Операцию получения списка заметок (с работой с API и с кэшем), операции изменения коллекции заметок (удалить, изменить, переместить).
  - о Операции для обновления интерфейса (UI операции).
  - о Можно использовать стандартные операции.
- Усложнение 1.
  - о Необходимо реализовать ваши операции как асинхронные.
  - о Необходимо добиться их корректного выполнения.
- Усложнение 2.
  - о Необходимо реализовать объект — диспетчер (Dispatcher), который будет управ-лять несколькими очередями и планированием выполнения операций.
  - о Требуется поддерживать несколько разных очередей для разных ресурсов (кэш, сеть, фоновое выполнение, задачи для интерфейса).

## Задание 5

- Необходимо реализовать локальный кэш заметок на базе CoreData.
  - о Приложение (через операции) должно уметь использовать локальный кэш через Core Data (сохранение заметки и списка, получение конкретной заметки, изменение конкретной заметки).
  - о Можно использовать любой способ построения стэка CoreData (но не PersistentContainer).
- Усложнение 1.
  - о Реализовать работу со стэком CoreData только на background context'ах.
- Усложнение 2.
  - о Реализовать несколько версий модели (не забывают про тег в гите на модель предыдущей версии и не забывают указать его при сдаче).
  - о Реализовать миграцию между ними, Lightweight миграции приниматься не будут.

## Раздел IV:

### Задание 6

- Проработать архитектуру и нарисовать схему приложения.
- Привести архитектуру приложения в соответствии со схемой.

## Задание 7

- Реализовать Unit-тесты.

LMS-платформа – не предусмотрена

### 5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

#### 5.3.1. Зачет

Список примерных вопросов

1. Язык Swift
2. Создание и уничтожение объектов (инициализация и деинициализация)
3. Слой представления — класс UIView и производные, представления из стандартной библиотеки UIKit
4. Слой контроллеров — класс UIViewController и производные, стандартные контроллеры
5. InterfaceBuilder – утилита для создания пользовательских интерфейсов
6. Базовые примитивы GCD
7. Очередь операций, еще раз про последовательные и параллельные очереди, но уже в контексте очередей операций
8. Операция: состояния, зависимости и немножко про выполнение на разных очередях
9. Классы NSURLSession\*Tasks
10. Организация запроса данных в CoreData

LMS-платформа – не предусмотрена

### 5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	учебно-исследовательская, научно-исследовательская	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности	ОПК-7	У-1	Зачет