

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
Математического обеспечения ЭВМ



Абрамов Г.В.

27.05.2023г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.03 Создание мобильных приложений iOS**

1. Код и наименование направления подготовки/специальности:
02.04.02 Фундаментальная информатика и информационные технологии
2. Профиль подготовки/специализация: Технологии разработки мобильных приложений
3. Квалификация выпускника: магистр
4. Форма обучения: очная
5. Кафедра, отвечающая за реализацию дисциплины: МО ЭВМ
6. Составители программы: Болотова Светлана Юрьевна,
кандидат физико-математических наук, доцент
7. Рекомендована: НМС факультета ПММ, протокол №7 от 26.05.2023.
8. Учебный год: _ 2024/2025 Семестр(ы): 3

9. Цели и задачи учебной дисциплины

Целями освоения дисциплины являются: углубленное изучение принципов программной инженерии в области разработки программного обеспечения для мобильных устройств, основных сценариев архитектуры мобильных устройств и получение навыков разработки профессионально-ориентированных программных средств и приложений на основе мобильных технологий с внедрением новых алгоритмических и технологических решений.

Основные задачи преподавания дисциплины, следующие: углубленное изучение мобильной операционной системы iOS;

изучение методов и приемов алгоритмизации поставленных задач с учетом возможностей современных мобильных технологий;

изучение основ проектирования и получение практических навыков проектирования архитектуры приложения и оценивания программных продуктов;

получение практического опыта в выборе технологий и средств разработки ПО, а также - разработки мобильного приложения с учетом целей и ограничений.

10. Место учебной дисциплины в структуре ООП: Дисциплина относится к части, формируемой участниками образовательных отношений, блока Б1. Изучение курса должно базироваться на знании обучающимися материала курса «Программирование на платформе iOS». Дисциплина является базовой для изучения курсов «Безопасность мобильных устройств».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-4 ПК-5	Способен разрабатывать профессионально - ориентированные программные средства и приложения на основе мобильных технологий Способен выбирать технологии и средства разработки мобильных приложений, определять ключевые сценарии для архитектуры мобильных приложений, разрабатывать новые алгоритмические, методические и технологические решения в сфере разработки мобильных приложений	ПК-4.1 ПК-5.1 ПК-5.2 ПК-5.3	Использует методы и приемы алгоритмизации поставленных задач с учетом возможностей современных мобильных технологий Владеет основами проектирования, знает элементы архитектурных решений информационных систем, технологии и средства разработки программного обеспечения. Проектирует архитектуру, оценивание ПО, применяет в практической деятельности профессиональные стандарты в области информационных технологий. Имеет	Знать: элементы архитектурных решений информационных систем, технологии и средства разработки программного обеспечения Уметь: проектировать архитектуру, оценивать ПО, применять в практической деятельности профессиональные стандарты в области информационных технологий. Владеть: выбором технологий и средств разработки ПО, определением целей, предпочтений и ограничений, а также - методами и приемами алгоритмизации поставленных задач с учетом возможностей современных мобильных технологий.

			практический опыт в выборе технологий и средств разработки ПО, определяет цели, предположения и ограничения.	
--	--	--	--	--

12. Объем дисциплины в зачетных единицах/час. — 5/180.

Форма промежуточной аттестации: экзамен

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость		
		Всего	По семестрам	
			3 семестр	-
Контактная работа				
в том числе:	лекции		16	
	практические			
	лабораторные		32	
	курсовая работа			
Самостоятельная работа			96	
Промежуточная аттестация		Экзамен	36	
Итого:			180	

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Многopotочность	Рассматривается использование многопоточности только в одной важной нише, связанной с HE блокировкой UI.	Создание мобильных приложений (iOS)
1.2	Жесты. JSON	Рассматривается пара тем (UserDefaults API и JSON encoding/decoding), связанных с постоянным хранением (persistence), а также - концептуальные основы обработки «жестов» мультитач (multi-touch gesture).	Создание мобильных приложений (iOS)
1.3	Data Flow.	Обсуждаются «Обертки Свойств», Property Wrappers, (типа @Published, @EnvironmentObject, @Binding) обсуждаются наряду с «издателями» Publishers. Затем EmojiArt использует их для автосохранения и для поддержки выбора между множеством палитр с эмоджи .	Создание мобильных приложений (iOS)
1.4	Modal Presentation и Navigation	Расширение возможностей UI с помощью методов модального представления и навигации.	Создание мобильных приложений (iOS)

		Получение текста от пользователя через TextField. Понимание ТИПа KeyPath. Хранение множества EmojiArt документов.	
1.5	Core Data	UserDefaults. Простая. Ограниченная (только Property Lists). Очень небольшая по объему. Codable / JSON. Понятный способ преобразования практически любой структуры данных в формат JSON обмена данными и наоборот. UIDocument. Core Data Мощная. Объектно-ориентированная. Элегантная интеграция со SwiftUI. Cloud Kit. Запоминание данных в базе данных, расположенной в «облаке» (то есть в интернете), которая работает полностью асинхронно. Следовательно, данные появляются на любых устройствах пользователя. Это очень простая в использовании база данных, у нее есть основные операции базы данных, но не такие полноценные как у Core Data, однако она отлично работает с Core Data (так что ваши данные в Core Data могут также появиться на всех устройствах) FileManager/URL/Data. Запоминание данных в Unix файловой системе в iOS.	Создание мобильных приложений (iOS)
1.6	Интеграция UIKit	Интеграция пре-SwiftUI iOS кода в SwiftUI.	Создание мобильных приложений (iOS)
2. Лабораторные работы			
2.1	Многопоточность	Идет разработка совершенно нового демонстрационного примера EmojiArt, начиная с обзора MVVM, а затем с использованием таких API, как ScrollView, UIImage и Drag & Drop.	Создание мобильных приложений (iOS)
2.2	Постоянное хранение и жесты	Усовершенствуется демонстрационный пример EmojiArt в плане постоянного хранения (persistence) внесенных изменений и поддержки мультитач «жестов» pinching (жест сведения и разведения пальцев), используемого для масштабирования, и rap (движение пальца по экрану) для перемещения документа по экрану.	Создание мобильных приложений (iOS)
2.3	Enroute Picker	Разработать UI, с помощью которого можно будет осуществлять фильтрацию рейсов по аэропорту прибытия destination, по аэропорту отправления origin, по авиакомпании airline и по тому, находятся рейсы уже в воздухе или еще ожидают вылета на земле inTheAir.	Создание мобильных приложений (iOS)
2.4	Core Data	Пример добавляет базу данных Core Data для Enroute.	Создание мобильных приложений (iOS)

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Многопоточность	2		6	12	20
2	Жесты. JSON	2			12	14
3	Data Flow.	2			12	14
4	Modal Presentation и Navigation	4			12	16
5	Core Data	4		8	12	24
6	Интеграция UIKit	2		6	12	20
7	Постоянное хранение и жесты			6	12	18
8	Enroute Picker			6	12	18
	Итого:	16		32	96	144

14. Методические указания для обучающихся по освоению дисциплины

Указание наиболее сложных разделов, работа с конспектами лекций, презентационным материалом. При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей по работе на LMS-платформе, своевременно подключаться к online-занятиям, соблюдать рекомендации по организации самостоятельной работы.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Ваттер, А. П. IPAD, IPHONE, MACBOOK и сервисы APPLE. Все о совместном использовании : руководство / А. П. Ваттер. — Санкт-Петербург : Наука и Техника, 2016. — 256 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/74666 .
2	Вейн, Ч. Swift подробно / Ч. Вейн ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 422 с. — ISBN 978-5-97060-780-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131722 .

б) дополнительная литература:

№ п/п	Источник
1	Разработка и продажа программ для iPhone и iPad / Дмитрий Елисеев. – БХВ- Петербург, 2012. ISBN 978-5-9775-0687-8.
2	SWIFT: БАЗОВОЕ РУКОВОДСТВО [Электронный ресурс] / М. Цукалос // Linux Format (Линукс Формат) .— 2016 .— №4 .— С. 82-85 .— Режим доступа: https://rucont.ru/efd/566911

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1	https://edu.vsu.ru/course/view.php?id=5156
2	https://cs193p.sites.stanford.edu Курс лекций Ipad and Iphone Application Development

* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы, онлайн-курсы, ЭУМК

16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных), курсовых работ и др.)

№ п/п	Источник
1	Соколова В. В. Разработка мобильных приложений : учебное пособие. — Томск: Изд-во ТПУ, 2014. — 175 с.: ил.
2	Нахавандипур В. iOS. Приемы программирования. — Санкт- Петербург: Питер, 2014. — 832 с.

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины используются модульно-рейтинговая и личностно-ориентированные технологии обучения (ориентированные на индивидуальность студента, компьютерные и коммуникационные технологии). В рамках дисциплины предусмотрены следующие виды лекций: информационная, лекция-визуализация, лекция с применением обратной связи.

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий, для организации самостоятельной работы обучающихся используется онлайн-курс, размещенный на платформе Электронного университета ВГУ (LMS moodle), а также другие Интернет-ресурсы, приведенные в п.15в.

18. Материально-техническое обеспечение дисциплины:

Аудитория должна быть оборудована учебной мебелью, компьютером, мультимедийным оборудованием (проектор, экран, средства звуковоспроизведения), допускается переносное оборудование.

Для самостоятельной работы необходимы компьютерные классы, помещения, оснащенные компьютерами с доступом к сети Интернет.

Программное обеспечение: Xcode

Материально-техническое обеспечение:

Моноблок Apple iMac MD093RU/A (14 шт.): процессор Intel Core i5 (2.70 GHz), оперативная память 8 Гб, HDD 1 Тб, видеокарта GeForce GT640M 512Мб, диагональ экрана 21,5"

Компьютер APPLE Mac Pro MD772RU/A Xeon W3565 в составе:

системный блок APPLE: процессор Intel Xeon W3565, оперативная память 8Гб, HDD 2Тб, видеокарта AMD Radeon HD 5770

Коммутатор HP ProCurve Switch 1400-24G

Мультимедиа-проектор BENQ MH535

Доска магнитно-маркерная на стенде (100x150см), 2-сторонняя, BRAUBERG PREMIUM

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Многопоточность	ПК-4	ПК-4.1 ПК-5.1	Собеседование
2.	Жесты. JSON	ПК-5	ПК-5.1	Собеседование
3.	Data Flow.	ПК-5	ПК-5.2	Лабораторная работа
4.	Modal Presentation и Navigation	ПК-5	ПК-5.2 ПК-5.3	Собеседование
5.	Core Data	ПК-4 ПК-5	ПК-4.1 ПК-5.2 ПК-5.3	Лабораторная работа
6.	Интеграция UIKit	ПК-4 ПК-5	ПК-4.1 ПК-5.2 ПК-5.3	Собеседование
7.	Постоянное хранение и жесты	ПК-5	ПК-5.1 ПК-5.2 ПК-5.3	Лабораторная работа
8.	Enroute Picker	ПК-5	ПК-5.2 ПК-5.3	Лабораторная работа

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
	Промежуточная аттестация форма контроля - экзамен			Перечень вопросов Практическое задание

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторная работа, тесты с вариантами ответов (ПК-4.1, ПК-5.1, ПК-5.2, ПК-5.3)

Пункт 1

Примеры лабораторных работ

Задание 1. Загрузите версию EmojiArt с Лекции. Не ломайте ничего, что там работает, как часть вашего решения этого задания.

Поддержите выбор одного или нескольких эмоджи, которые были “перетянуты” на ваш EmojiArt документ (то есть вы выбираете эмоджи в документе, а не на палитре в верхней части экрана). Вы можете показать, какие эмоджи были выбраны, любым способом. Этот выбор не требует постоянного хранения (другими словами, повторный запуск вашего приложения не сохранит этот выбор).

Выполнение жеста Tap на невыбранном эмоджи должно выбирать этот эмоджи.

Выполнение жеста Tap на выбранном эмоджи должно делать этот эмоджи не выбранным.

Однократный жест Tap на фоновом изображении вашего EmojiArt (то есть один раз кликнуть за исключением эмоджи) должен сделать не выбранными все эмоджи.

“Перетаскивание” выбранных эмоджи должно перемещать всю выборку эмоджи вслед за пальцами пользователя.

Если пользователь выполняет жест Drag, когда ничего не выбрано, то перемещайте (pan) полный документ.

Если пользователь выполняет жест Pinch где-то на EmojiArt документе и есть выбор эмоджи, то все выбранные эмоджи должны масштабироваться в соответствии с жестом Pinch.

Если нет выбора эмоджи во время выполнения жеста Pinch, то выполняется масштабирование полного документа.

Сделайте возможным удаление эмоджи из EmojiArt документа. В этом обязательном пункте Задания намеренно не указывается, какое UI действие должно вызывать удаление эмоджи. Будьте креативны и попытайтесь найти способ удаления эмоджи, который был бы комфортабельным и интуитивным.

Задание 2. Уберите опцию “случайного выбора числа карт” для темы (theme) из вашей игры Memorize. Теперь каждая тема (theme) будет иметь своё собственное, заранее определенное число карт. Другими словами. Сколько карт в игре - это часть темы (theme) для этой игры и не может быть больше “случайным”. Каждый раз, когда стартует новая игра, печатайте print JSON представление темы (theme), используемой в данный момент на консоли. Все элементы темы (theme) (её имя, набор эмоджи для выбора, сколько пар карт в игре и цвет темы) должны быть включены.

Задание 3. Теперь ваше приложение Memorize должно при запуске отображать UI «выбора темы».

Примеры см. в прикрепленных изображениях, но вы можете проявить творческий подход к тому, как показывать каждую тему.

Используйте List для показа тем (themes).

Каждая строка в List показывает имя темы (theme), цвет темы, сколько карт в теме и некоторую выборку эмоджи.

Касание темы (theme) в List переводит (так как List находится внутри NavigationView) вас на игру с этой темой.

Во время игры название темы должно где-то отображаться на экране, и вы также должны продолжать поддерживать существующие функции, такие как счет, новая игра и т. д. (Но вы можете изменить UI, если хотите).

Это нормально, если при переходе от игры обратно к выбору игры в List, а затем обратно к текущей игре в полном разгаре все-таки игра перезапускается, хотя разумные реализации, вероятно, этого не сделают (кроме случаев, когда тема (theme) изменена (см. Ниже), поскольку это в этом случае вы все равно захотите перезапустить игру).

Предоставьте некоторый UI для добавления новой темы (theme) в список тем List.

View для выбора темы (theme) должен поддерживать режим редактирования Edit Mode, при котором вы можете уничтожать темы (themes) и при котором у вас есть доступ к некоторому UI (то есть кнопке Button

или изображении Image в каждой строке), который выводит вас модально (через sheet или popover) на UI Редактора Темы и дает возможность редактировать эту тему (theme).

Редактора Темы должен быть формой Form.

В Редакторе Темы пользователю разрешается редактировать имя темы (theme), добавлять эмоджи к теме, удалять эмоджи из темы и сколько карт в этой теме. (Есть Дополнительный Пункт, который связан с возможностью редактирования цвета темы).

Темы (themes) должны постоянно сохраняться (перезапуск вашего приложения не должно вызывать потерю всех отредактированных тем).

Ваш UI должен хорошо выглядеть как на iPhone, так и на iPad.

Заставьте ваше приложение работать на любом физическом устройстве iOS.

Описание технологии проведения

Каждая лабораторная работа выполняется на основе задания и соответствующей лекции. После выполнения задания на лабораторную работу каждый студент должен выполнить те же действия, но уже по своей теме, которая относится к домашнему заданию по дисциплине. Таким образом, после каждой лабораторной работы формируются необходимые части/знания для выполнения домашнего задания.

Требования к выполнению заданий (или шкалы и критерии оценивания)

Каждая лабораторная работа оценивается по принципу «зачет/незачет»

«Зачет» ставится, если сделано верно не менее 80% задания

«Незачет» ставится, если сделано верно менее 80% задания

Пункт 2 Пример тестов:

1. Для чего используются контейнеры HStacks и VStacks в SwiftUI?

Правильный ответ: контейнеры позиционируют View внутри себя, распределяя пространство между своими сабвью. HStack располагает View горизонтально, VStack - вертикально.

2. Что происходит в данном коде? Код представлен на языке Swift.

```
let s: String? = ...
```

```
let ss = s ?? "hello"
```

Правильный ответ: Оператор ?? используется для создания выражения, которое становится значением по умолчанию, если Optional не установлен. Если значение переменной s будет установлено, то в переменную ss запишется значение ассоциированных данных переменной s. Если значение переменной s не будет установлено (будет .none / nil), то в переменную ss запишется значение по умолчанию "hello".

3. Что используется для обозначения состояния View в SwiftUI?

Правильный ответ: @State

4. Что позволяет изменить внешний вид View в SwiftUI?

Правильный ответ: функции-модификаторы

5. Как в SwiftUI называется протокол, содержащий функции, которые можно использовать для создание некоторой вещи JSON-представляемой?

Правильный ответ: Codable

6. Что такое декларативный синтаксис в SwiftUI?

Правильный ответ: Декларативный синтаксис - это современная языковая парадигма, которая помогает разработчикам писать код процедурно. Используя декларативный синтаксис, разработчик описывает код, который он хочет написать, не беспокоясь о том, как он будет показан / реализован. SwiftUI использует декларативный синтаксис, поэтому вы можете просто указать, что должен делать ваш пользовательский интерфейс. Например: вы можете написать, что вам нужно изображение в нижней части экрана.

Описание технологии проведения

Текущая аттестация проводится на занятии одновременно во всей учебной группе в виде теста в электронной образовательной среде «Электронный университет ВГУ», адрес курса — <https://edu.vsu.ru/course/> Тест составляется из материалов ФОСа, формируется системой автоматически путём добавления случайных вопросов, количество которых соответствует образцу билета. Большая часть вопросов проверяется автоматически, проверки преподавателем с ручным оцениванием требуют только отдельные вопросы, представленные в форме эссе. Ограничение по времени на каждую попытку — 20 минут, количество попыток — 1, выставление окончательной оценки — по высшему баллу.»

Тест оценивается по принципу «зачет/незачет»

«Зачет» ставится, если сделано верно не менее 80% задания

«Незачет» ставится, если сделано верно менее 80% задания

Задания раздела 20.1 пункт 2 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных знаний по результатам освоения данной дисциплины

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Собеседование по экзаменационным билетам (по билетам к зачету)

Пример экзаменационного билета

1. Обертки Свойств
2. Интеграция в Swift UI

Описание технологии проведения

Экзамен проходит в письменной форме

Требования к выполнению заданий, шкалы и критерии оценивания

Критерии оценки:

оценка «отлично» выставляется обучающемуся, если объем правильного ответ не менее чем 75%;

оценка «хорошо» выставляется обучающемуся, если объем правильного ответ не менее чем 50%;

оценка «удовлетворительно» выставляется обучающемуся, если объем правильного ответ не менее чем 30%;

оценка «неудовлетворительно» выставляется обучающемуся, если объем правильного ответ менее чем 30%.