


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

УТВЕРЖДАЮ

Заведующий кафедрой
Программного обеспечения и администрирования
информационных систем


Артемов М.А.
02.04.2024г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

**Б1.В.ДВ.03.01 Разработка приложений с использованием
фреймворков**

1. Код и наименование направления подготовки/специальности:

09.03.03 Прикладная информатика

2. Профиль подготовки/специализация:

Прикладная информатика в юриспруденции

3. Квалификация (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины:

Кафедра программного обеспечения и администрирования информационных систем

6. Составители программы: Артёмов Михаил Анатольевич, проф., д.ф.-м.н.

7. Рекомендована: НМС факультета ПММ, протокол №5 от 22.03.2024

8. Учебный год: 2024-2025

Семестр(ы)/Триместр(ы): 7

9. Цели и задачи учебной дисциплины:

Цель - ознакомление с методологией использования фреймворков при разработке программного обеспечения; формирование целостного представления о данной области знания.

Задачи дисциплины:

- сформировать представление о теоретических и практических основах разработки сложных информационных систем;
- ознакомить с терминологией предметной области и основными определениями;
- сформировать базовое представление о шаблонах разработки веб-сервисов и приложений;
- дать представление о методах защиты от распространённых векторов информационных атак;
- получить представление об использовании шаблонов проектирования как мер усиления информационной безопасности.

10. Место учебной дисциплины в структуре ООП: Дисциплина относится к вариативной части блока Б1.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-1	Способен обрабатывать, анализировать и оформлять результаты исследований и разработок под руководством специалиста более высокой квалификации	ПК-1.1	Применяет при обработке данных стандартное и оригинальное программное обеспечение	Знать: теоретические и практические основы разработки программного обеспечения, терминологию предметной области Уметь: использовать стандартные алгоритмы и методы разработки программного обеспечения для решения задач Владеть: знаниями о методах разработки программного обеспечения в области обработки данных.
ПК-3	Способен разрабатывать и тестировать алгоритмическое и программное обеспечение для отдельных компонентов вычислительных систем	ПК-3.1 ПК-3.2	Демонстрирует знание методов и способов формализации и алгоритмизации поставленных задач, способы анализа требований к ПО, методологии проектирования ПО Создает программный код с использованием языков программирования, определения и	Знать: шаблоны проектирования распределённых систем и веб-сервисов. Уметь: использовать архитектурные подходы для построения отказоустойчивых и безопасных распределённых систем. Владеть: знаниями о методах проектирования и тестирования распределённых систем и веб-сервисов в различных предметных областях.

			манипулирования данными и оформляет его в соответствии с установленными требованиями, проверяет и отлаживает его	
--	--	--	------------------------------------------------------------------------------------------------------------------	--

12. Объем дисциплины в зачетных единицах/час: 3/108

Форма промежуточной аттестации:

Зачёт

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			7 семестр		...
Аудиторные занятия		64	64		
в том числе:	лекции	16	16		
	практические	16	16		
	лабораторные	32	32		
Самостоятельная работа		44	44		
в том числе: курсовая работа (проект)		0	0		
Форма промежуточной аттестации (зачёт с оценкой)		0	0		
Итого:		108	108		

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
2. Лекции			
1.1	Введение	История, цели применения шаблонов проектирования в веб-разработке. Обзор протокола HTTP.	-
1.2	Основы криптографии.	Основные подходы к защите информации: хэширование, симметричное и асимметричное шифрование. Реализация подходов на конкретных алгоритмах. Критерии криптографической устойчивости.	-
1.3	Разделение доступа в веб-сервисах.	Идентификация. Аутентификация. Авторизация. Подходы, основные проблемы. Делегация процессов аутентификации и делегации.	-
1.4	Шаблоны проектирования веб-приложений.	Понятие шаблона проектирования. Архитектура REST. Шаблон проектирования MVC.	-
1.5	Языки программирования с динамической типизацией.	Отличия языков со статической типизацией от языков с динамической типизацией. Разновидности языков. Синтаксис языка Ruby.	
1.6	Использование	Контейнеры, образы. Docker, Docker-Compose.	-

	микросервисной архитектуры в веб-разработке.	Отличия монолитной и микросервисной архитектуры. Облачные вычисления. Kubernetes.	
1.7	Операционная система Linux.	Скриптовые языки. Интерпретируемые и компилируемые языки. Общая архитектура Linux. Написание простейшей автоматизации с помощью Bash.	-
1.8	Контейнерная разработка.	Использование Docker-контейнеров для разработки и разворачивания веб-сервисов и приложений.	-
1.9	Разработка веб-приложения.	Процессы и методологии разработки программного обеспечения. Специфика разработки веб-приложений.	-

13.2. Темы (разделы) дисциплины и виды занятий

14. Методические указания для обучающихся по освоению дисциплины

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				Всего
		Лекции	Лабораторные	Практические	Самостоятельная работа	
1	Введение	2	0	0	2	4
2	Основы криптографии.	2	0	2	6	12
3	Разделение доступа в веб-сервисах.	2	0	2	6	10
4	Шаблоны проектирования веб-приложений.	2	0	2	6	10
5	Языки программирования с динамической типизацией.	2	2	2	2	6
6	Использование микросервисной архитектуры в веб-разработке.	2	0	2	4	8
7	Операционная система Linux.	1	8	2	4	15
8	Контейнерная разработка.	1	10	2	6	12
9	Разработка веб-приложения.	2	12	0	8	10
Итого:		16	32	16	44	108

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
	https://lib.vsu.ru/url.php?url=http://e.lanbook.com ЭБС изд-во «Лань»
1	Мартин К. Криптография. Как защитить свои данные в цифровом пространстве / К. Мартин – М.: БОМБОРА, 2022 – 371 с.

б) дополнительная литература:

№	Источник
---	----------

п/п	
	https://lib.vsu.ru/url.php?url=http://e.lanbook.com ЭБС изд-во «Лань»
1	Райс Л. Безопасность контейнеров. Фундаментальный подход к защите контейнеризированных приложений / Л. Райс – СПб.: Питер, 2021 – 224 с.
2	Янка Т. Безопасность веб-приложений. Исчерпывающий гид для начинающих разработчиков / Т. Янка – М.:БОМБОРА, 2023 – 428 с.

в) информационные электронно-образовательные ресурсы:

№ п/п Источник
 Электронный университет ВГУ
 Зиновьев С. В. Разработка приложений с использованием фреймворков
<https://edu.vsu.ru/course/view.php?id=10972>

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п Источник
 Парасрам Ш. Kali Linux. Тестирование на проникновение и безопасность / Ш. Парасрам, А. Замм – СПб: Питер, 2022. – 448 с.

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости):

18. Материально-техническое обеспечение дисциплины: аудитория, проектор

19. Оценочные средства для проведения текущей и промежуточной аттестаций

№ п/п	Наименование раздела дисциплины (модуля)	Компетенции	Индикаторы достижения компетенции	Оценочные средства
1.	Раздел 1.1 Введение Раздел 1.2 Основы криптографии. Раздел 1.3 Разделение доступа в веб-сервисах. Раздел 1.4 Шаблоны проектирования веб-приложений. Раздел 1.5 Языки программирования с динамической типизацией. Раздел 1.6 Использование микросервисной архитектуры в веб-разработке. Раздел 1.7 Операционная система Linux. Раздел 1.8 Контейнерная разработка. Раздел 1.9 Разработка веб-приложения.	ПК-1 ПК-1, ПК-3	ПК-1.1 ПК-1.1, ПК-3.1, ПК-3.2	Опрос, обсуждение теоретических и практических вопросов

№ п/п	Наименование раздела дисциплины (модуля)	Компетенции	Индикаторы достижения компетенции	Оценочные средства
	Промежуточная аттестация форма контроля – зачёт			П.20.2

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Опрос, обсуждение

20.2 Промежуточная аттестация

Вопросы:

1. HTTP запросы: виды, строение.
2. URI: что такое, где используются, разновидности, пример.
3. Архитектура REST: базовые принципы.
4. Паттерн проектирования MVC: базовые принципы.
5. Уязвимости при работе с веб-приложениями: примеры, способы «закрытия дыр».
- 6.
7. Условные операторы в Ruby: виды, использование.
8. Циклы в Ruby: виды, использование.
9. Основные типы данных в Ruby.
10. Аутентификация, идентификация, авторизация.
11. Аутентификация по паролю (password authentication): способы, проблемы.
12. Аутентификация по сертификату (certificate authentication): принцип работы, проблемы.
13. Мультифакторная аутентификация (MFA): принцип работы, способы, проблемы.
14. Аутентификация на токенах (token authentication): способы, проблемы.
15. OAuth: что такое, принцип действия, проблемы.
16. Централизованные vs. децентрализованные системы контроля версий.
17. Ветки в Git: что такое, как с ними работать.
18. Merge в Git.
19. Rebase в Git.
20. Виды аутентификации в клиент-серверной модели веб-сервисов.
21. CRUD. Его применение в Rails.
22. Отличия генератора случайных чисел и псевдослучайных чисел.
23. Хэширование — что такое, какие требования, какие где применяется.
24. MAC и HMAC — что такое, чем отличаются, как работают.
25. Симметричное шифрование — что такое, как работает, какие бывают, где применяются.
26. Ассиметричное шифрование — что такое, как работает, какие бывают, где применяются.
27. HTTPS: что такое, как работает установление соединения.
28. Автоматизация настройки рабочего окружения и контейнеризация — что такое, зачем. Принципы работы с Docker.
29. Облачные технологии и их отличия от классической (on-premise) архитектуры.
30. IAM — что такое, зачем нужно.
31. Виды Compute (вычислительных мощностей) в облаке и их отличия.
32. Виды Storage (хранилищ) в облаке и их отличия.
33. Особенности сетевого взаимодействия в облаках.
34. Журналирование, аудит и мониторинг — что такое и зачем.
35. Хранение чувствительных данных в облаке.
36. Kubernetes — что такое, основные принципы работы.
37. Архитектура Kubernetes кластера. Основные составные части.

Пример КИМ

Вопрос: Паттерн проектирования MVC: базовые принципы.