МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Донской казачий государственный институт пищевых технологий и бизнеса (филиал) федерального государственного бюджетного образовательного учреждения «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)»

Донской казачий государственный институт пищевых технологий и бизнеса (филиал)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00AA7853CEFA35F0837BA866654F3503A6
Владелец: Павлятенко Геннадий Владимирович
Действителен: с 08.12.2025 до 03.03.2027

УТВЕРЖДАЮ
И.о. директора ДКГИПТиБ (филиала)
Г.В. Павлятенко
«10» апреля 2026 г.

Рабочая программа дисциплины (модуля)

ФТД.02 Управление разработкой веб-сайтов

Кафедра:	Физика, математика и информационные технологии
Направление подготовки:	09.03.03 Прикладная информатика
Направленность (профиль):	Управление проектами в области информационных технологий
Квалификация выпускника:	Бакалавр
Форма обучения:	очно-заочная
Год набора:	2026
Общая трудоемкость:	108 часов/3 з.е.

Ростов-на-Дону, 2026 г.

Программу составил(и):
преподаватель Горбунов В.С.

Рабочая программа дисциплины (модуля)
"Управление разработкой веб-сайтов"
разработана на основании учебного плана, утвержденного ученым советом 26 марта 2026 г. протокол № 9 в соответствии с ФГОС ВО Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922), 06.014. Профессиональный стандарт "МЕНЕДЖЕР ПО ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ", утверждённый приказом Министерства труда и социальной защиты Российско Федерации от 30 августа 2021 г. N 588н (зарегистрирован Министерством юстиции Российской Федерации 1 октября 2021 г., регистрационный N 65223)

Руководитель ОПОП
канд.экон.наук,доцент Левчук В.В._____ ___

Рабочая программа обсуждена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от 30 июня 2026 г. № 11
И.о.зав. кафедрой Левчук Валерия Владимировна_____

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Цели:

Цель освоения дисциплины «Управление разработкой веб-сайтов» (далее – «дисциплина») состоит в формировании у студентов компетенций в соответствии с ФГОС ВО, учебным планом и основной профессиональной образовательной программой (далее ОПОП)

1.2. Задачи:

Задачами преподавания дисциплины является изучение вопросов теории и практики написания различных типов интернет - приложений, используя при этом самые популярные средства, такие как PHP, HTML, MySQL и CSS. Размещение ресурсов во всемирной сети интернет.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

Цикл (раздел) ОП: "Управление разработкой веб-сайтов"

Связь с последующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Преддипломная практика	9	ПКС-1.1, ПКС-1.2, ПКС-1.3, ПКС-2.1, ПКС-2.2, ПКС-2.3, ПКС-3.1, ПКС-3.2, ПКС -3.3, ПКС-4.1, ПКС-4.2, ПКС-4.3

Распределение часов дисциплины

Семестр (<Курс>.<Семестр на курсе>)	8 (4.2)		Итого	
Недель	13 4/6			
Вид занятий	УП	РП	УП	РП
Лекции	8	8	8	8
Лабораторные	8	8	8	8
В том числе электрон.	16	16	16	16
Итого ауд.	16	16	16	16
Контактная работа	16	16	16	16
Сам. работа	92	92	92	92
Итого	108	108	108	108

Вид промежуточной аттестации:

ЗаО 8 семестр

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций и индикаторов их

ПКС-2:Способен формировать и согласовывать принципы управления проектами в области прикладных информационных технологий

ПКС-2.1: Знает базовые принципы контрактования в информационных технологиях, методы непрерывного улучшения управления проектами в области прикладных информационных технологий

ПКС-2.2: Умеет осуществлять руководство проектами в области информационных технологий, организовывать деятельность по непрерывному улучшению управления проектами в области прикладных информационных технологий

ПКС-2.3: Владеет базовыми навыками применения методов формирования и согласования принципов управления проектами в области информационных технологий

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименования разделов, тем, их краткое содержание и результаты освоения /вид занятия/	Семестр	Часы	Интеракт.	Прак. подг.	Индикаторы достижения компетенции	Виды контроля
	Раздел 1.Управление разработкой веб-сайтов						
1.1	**Тема 1. Введение в разработку веб-сайтов. Фреймворк Django Краткое содержание: рассмотрена архитектура Django, его основные компоненты и принципы работы.**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке

	концепция Model-View-Template (MVT) и научатся создавать простые веб-приложения с использованием Django Знать: Архитектуру и основные компоненты Django. Принцип работы MVT. Базовые концепции веб-разработки Уметь: Настраивать среду разработки для Django. Создавать новый проект Django и приложения внутри него. Определять модели данных и работать с ORM Django Владеть: навыками программирования на Python. Навыками работы с интегрированной средой разработки. Навыками отладки и устранения неполадок в приложениях Django /Ср/						
1.2	**Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции Краткое содержание: изучение шаблона архитектуры MVS (Model-View-Controller) в Laravel. Рассматриваются контроллеры и маршруты, которые отвечают за обработку HTTP-запросов и определяют логику приложения, изучаются модели и миграции, которые представляют структуру и взаимодействие с базами данных Знать: Концепцию шаблона MVS и его преимущества. Назначение контроллеров и маршрутов в Laravel. Синтаксис определения маршрутов и связывания их с контроллерами Уметь: Создавать контроллеры и определять методы для обработки HTTP-запросов. Настраивать маршруты и связывать их с соответствующими методами контроллеров Владеть: Навыками разработки приложений с использованием шаблона MVS. Способностью работать с моделями и выполнять операции с базой данных /Ср/**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.3	**Тема 3. Шаблоны Django. Администраторская панель Краткое содержание: Работа с шаблонами Django: синтаксис и структура. Наследование шаблонов и использование блоков. Фильтры и теги для форматирования и манипуляции данными. Использование статических файлов (CSS, JavaScript) в шаблонах. Администраторская панель Django: настройка, регистрация моделей. Кастомизация админки: формы, списки, фильтры Знать: Синтаксис и структура шаблонов Django; Наследование шаблонов и использование блоков; Фильтры и теги для форматирования и манипуляции данными Уметь: Создавать и редактировать**	8	10	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке

	шаблоны; Создавать базовые и дочерние шаблоны с использованием наследования и блоков; Применять фильтры для форматирования данных в шаблонах Владеть: Навыком создания кастомизированных административных интерфейсов, включая формы, списки и фильтры /Ср/						
1.4	**Тема 4. Связи моделей и виды. Директивы и теги Краткое содержание: изучение связей между моделями в Django, теги и директивы, используемые для создания и управления этими связями, рассматриваются примеры использования связей, особенности их реализации и возможные проблемы, с которыми может столкнуться разработчик Знать: Виды связей моделей в Django. Директивы для создания связей. Теги для отображения связанных данных в шаблонах Уметь: Создавать связи между моделями с помощью директив. Настраивать параметры связей. Выполнять запросы к связанным моделям через менеджеры и атрибуты Владеть: Навыками проектирования и моделирования связей между сущностями в приложении. Навыками работы с ORM Django для управления связями на уровне базы данных /Ср/**	8	10	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.5	**Тема 5. Параметры в URL-запросах и имена маршрутов. Наследование шаблонов, пользовательские теги Краткое содержание: работа с параметрами URL-запросов, именами маршрутов, наследованием шаблонов и пользовательскими тегами в Django. Рассматриваются способы передачи данных через URL, использование именованных маршрутов, создание иерархии шаблонов через наследование, а также создание и применение пользовательских тегов и фильтров в шаблонах. Знать: Синтаксис передачи параметров в URL-запросах. Концепцию именованных маршрутов в Django. Принципы наследования шаблонов Уметь: Передавать данные через параметры URL-запросов. Использовать именованные маршруты для ссылок на URL-адреса. Создавать иерархию шаблонов с помощью наследования Владеть: навыками работы с параметрами URL и именованными маршрутами; навыками организации структуры шаблонов через наследование;**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке

	навыками создания пользовательских тегов и фильтров для решения специфических задач /Ср/						
1.6	**Тема 6. Обратное разрешение адресов. Статические файлы. Работа с файлами Краткое содержание: обратное разрешению адресов (reverse proxy) и работе с файлами в веб-приложениях. Рассматриваются статические файлы, такие как HTML, CSS и JavaScript, и их обслуживание веб-сервером. Также изучается работа с файлами в приложении, включая чтение, запись и загрузку файлов. Знать: Концепцию обратного прокси-сервера и его преимущества. Различие между статическими и динамическими файлами. Методы обслуживания статических файлов веб-сервером Уметь: Настраивать обратный прокси-сервер для распределения нагрузки. Обслуживать статические файлы через веб-сервер. Читать и записывать файлы в приложении. Владеть: Навыками настройки и управления обратным прокси-сервером. Способами эффективной работы с файлами в приложении. Инструментами для тестирования и отладки при работе с файлами. /Ср/**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.7	**Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar Краткое содержание: рассматриваются классы ListView, DetailView и CreateView, являющиеся частью обобщенных представлений в Django. Также будут изучены улучшения, такие как Django Debug Toolbar, которая является полезным инструментом для отладки и оптимизации веб-приложений Django. Знать: назначение и возможности классов ListView, DetailView и CreateView; как настраивать и использовать эти классы в своих представлениях Уметь: создавать представления на основе классов ListView, DetailView и CreateView; Настраивать атрибуты и методы этих классов в соответствии с требованиями приложения Владеть: Навыками работы с обобщенными представлениями в Django; Навыками настройки и использования классов ListView, DetailView и CreateView; Навыками установки и настройки Django Debug Toolbar /Ср/**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.8	**Тема 8. Кастомизация администраторской панели и миксины. Постраничная**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке

	навигация Краткое содержание: кастомизация администраторской панели в Django, включая настройку отображения моделей, полей, фильтров и действий. Также рассматриваются миксины - повторно используемые куски кода для представлений. Изучается постраничная навигация для отображения списков объектов по частям. Знать: Структуру и компоненты администраторской панели Django. Способы настройки отображения моделей, полей, фильтров и действий. Концепцию миксинов и их роль в Django Уметь: Регистрировать модели в администраторской панели. Настраивать отображение полей и списков моделей. Создавать пользовательские фильтры и действия Владеть: Навыками кастомизации интерфейса администраторской панели; Способностью создавать повторно используемый код с помощью миксинов /Ср/						
1.9	**Тема 9. Регистрация и авторизация Краткое содержание: процессы регистрации и авторизации пользователей в Django. Знать: Принципы работы системы аутентификации в Django. Структуру стандартной модели пользователя (User). Концепцию сессий и их роль в процессе авторизации. Уметь: Настраивать параметры аутентификации в settings.py. Создавать формы регистрации и входа пользователей. Владеть: Навыками интеграции системы аутентификации в веб-приложение. Навыками кастомизации моделей пользователей и регистрационных форм. /Ср/**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.10	**Тема 10. CKEditor, капча. Кэширование Краткое содержание: визуальный редактор HTML для веб-разработки. Капча (CAPTCHA). Временное хранение данных для повышения производительности Знать: Принципы работы CKEditor и его интеграцию в веб-приложения. Концепцию и назначение капчи в веб-разработке Уметь: Настраивать и использовать CKEditor в веб-приложениях. Внедрять и обрабатывать капчу для защиты форм от ботов Владеть: Навыками работы с JavaScript и библиотеками для интеграции CKEditor. Методами работы с капчей и ее валидацией /Ср/**	8	9	0	0	ПКС-2.1,ПКС-2.2,ПКС-2.3	вопросы к самоподготовке
1.11	**Тема 1. Введение в разработку веб-сайтов. Фреймворк Django Краткое содержание: рассмотрена**	8	1	0	0	ПКС-2.1	устный опрос, тестирование

	архитектура Django, его основные компоненты и принципы работы. концепция Model-View-Template (MVT) и научатся создавать простые веб-приложения с использованием Django Знать: Архитектуру и основные компоненты Django. Принцип работы MVT. Базовые концепции веб-разработки /Лек/						
1.12	**Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции Краткое содержание: изучение шаблона архитектуры MVS (Model-View-Controller) в Laravel. Рассматриваются контроллеры и маршруты, которые отвечают за обработку HTTP-запросов и определяют логику приложения, изучаются модели и миграции, которые представляют структуру и взаимодействие с базами данных Знать: Концепцию шаблона MVS и его преимущества. Назначение контроллеров и маршрутов в Laravel. Синтаксис определения маршрутов и связывания их с контроллерами /Лек/**	8	1	0	0	ПКС-2.1	устный опрос, тестирование
1.13	**Тема 3. Шаблоны Django. Администраторская панель Краткое содержание: Работа с шаблонами Django: синтаксис и структура. Наследование шаблонов и использование блоков. Фильтры и теги для форматирования и манипуляции данными. Использование статических файлов (CSS, JavaScript) в шаблонах. Администраторская панель Django: настройка, регистрация моделей. Кастомизация админки: формы, списки, фильтры Знать: Синтаксис и структура шаблонов Django; Наследование шаблонов и использование блоков; Фильтры и теги для форматирования и манипуляции данными /Лек/**	8	1	0	0	ПКС-2.1	устный опрос, тестирование
1.14	**Тема 4. Связи моделей и виды. Директивы и теги Краткое содержание: изучение связей между моделями в Django, теги и директивы, используемые для создания и управления этими связями, рассматриваются примеры использования связей, особенности их реализации и возможные проблемы, с которыми может столкнуться разработчик Знать: Виды связей моделей в Django. Директивы для создания связей. Теги для отображения связанных данных в шаблонах /Лек/**	8	1	0	0	ПКС-2.1	устный опрос, тестирование
1.15	**Тема 5. Параметры в URL-запросах и имена маршрутов. Наследование шаблонов, пользовательские теги Краткое содержание: работа с**	8	1	0	0	ПКС-2.1	устный опрос, тестирование

	параметрами URL-запросов, именами маршрутов, наследованием шаблонов и пользовательскими тегами в Django. Рассматриваются способы передачи данных через URL, использование именованных маршрутов, создание иерархии шаблонов через наследование, а также создание и применение пользовательских тегов и фильтров в шаблонах. Знать: Синтаксис передачи параметров в URL-запросах. Концепцию именованных маршрутов в Django. Принципы наследования шаблонов /Лек/						
1.16	**Тема 6. Обратное разрешение адресов. Статические файлы. Работа с файлами Краткое содержание: обратное разрешению адресов (reverse proxy) и работе с файлами в веб-приложениях. Рассматриваются статические файлы, такие как HTML, CSS и JavaScript, и их обслуживание веб-сервером. Также изучается работа с файлами в приложении, включая чтение, запись и загрузку файлов. Знать: Концепцию обратного прокси-сервера и его преимущества. Различие между статическими и динамическими файлами. Методы обслуживания статических файлов веб-сервером /Лек/**	8	1	0	0	ПКС-2.1	устный опрос, тестирование
1.17	**Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar Краткое содержание: рассматриваются классы ListView, DetailView и CreateView, являющиеся частью обобщенных представлений в Django. Также будут изучены улучшения, такие как Django Debug Toolbar, которая является полезным инструментом для отладки и оптимизации веб-приложений Django. Знать: назначение и возможности классов ListView, DetailView и CreateView; как настраивать и использовать эти классы в своих представлениях /Лек/**	8	1	0	0	ПКС-2.1	устный опрос, тестирование
1.18	**Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация Краткое содержание: кастомизация администраторской панели в Django, включая настройку отображения моделей, полей, фильтров и действий. Также рассматриваются миксины - повторно используемые куски кода для представлений. Изучается постраничная навигация для отображения списков объектов по частям. Знать: Структуру и компоненты администраторской**	8	1	0	0	ПКС-2.1	устный опрос, тестирование

	панели Django. Способы настройки отображения моделей, полей, фильтров и действий. Концепцию миксинов и их роль в Django /Лек/						
1.19	**Тема 1. Введение в разработку веб-сайтов. Фреймворк Django Краткое содержание: рассмотрена архитектура Django, его основные компоненты и принципы работы. концепция Model-View-Template (MVT) и научатся создавать простые веб-приложения с использованием Django Уметь: Настраивать среду разработки для Django. Создавать новый проект Django и приложения внутри него. Определять модели данных и работать с ORM Django Владеть: навыками программирования на Python. Навыками работы с интегрированной средой разработки. Навыками отладки и устранения неполадок в приложениях Django /Лаб/**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе
1.20	**Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции Краткое содержание: изучение шаблона архитектуры MVS (Model-View-Controller) в Laravel. Рассматриваются контроллеры и маршруты, которые отвечают за обработку HTTP-запросов и определяют логику приложения, изучаются модели и миграции, которые представляют структуру и взаимодействие с базами данных Уметь: Создавать контроллеры и определять методы для обработки HTTP-запросов. Настраивать маршруты и связывать их с соответствующими методами контроллеров Владеть: Навыками разработки приложений с использованием шаблона MVS. Способностью работать с моделями и выполнять операции с базой данных /Лаб/**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе
1.21	**Тема 3. Шаблоны Django. Администраторская панель Краткое содержание: Работа с шаблонами Django: синтаксис и структура. Наследование шаблонов и использование блоков. Фильтры и теги для форматирования и манипуляции данными. Использование статических файлов (CSS, JavaScript) в шаблонах. Администраторская панель Django: настройка, регистрация моделей. Кастомизация админки: формы, списки, фильтры Уметь: Создавать и редактировать шаблоны; Создавать базовые и дочерние шаблоны с использованием наследования и блоков; Применять фильтры для форматирования данных в шаблонах Владеть:**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе

	Навыком создания кастомизированных административных интерфейсов, включая формы, списки и фильтры /Лаб/						
1.22	**Тема 4. Связи моделей и виды. Директивы и теги Краткое содержание: изучение связей между моделями в Django, теги и директивы, используемые для создания и управления этими связями, рассматриваются примеры использования связей, особенности их реализации и возможные проблемы, с которыми может столкнуться разработчик Уметь: Создавать связи между моделями с помощью директив. Настраивать параметры связей. Выполнять запросы к связанным моделям через менеджеры и атрибуты Владеть: Навыками проектирования и моделирования связей между сущностями в приложении. Навыками работы с ORM Django для управления связями на уровне базы данных /Лаб/**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе
1.23	**Тема 5. Параметры в URL-запросах и имена маршрутов. Наследование шаблонов, пользовательские теги Краткое содержание: работа с параметрами URL-запросов, именами маршрутов, наследованием шаблонов и пользовательскими тегами в Django. Рассматриваются способы передачи данных через URL, использование именованных маршрутов, создание иерархии шаблонов через наследование, а также создание и применение пользовательских тегов и фильтров в шаблонах. Уметь: Передавать данные через параметры URL-запросов. Использовать именованные маршруты для ссылок на URL-адреса. Создавать иерархию шаблонов с помощью наследования Владеть: навыками работы с параметрами URL и именованными маршрутами; навыками организации структуры шаблонов через наследование; навыками создания пользовательских тегов и фильтров для решения специфических задач /Лаб/**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе
1.24	**Тема 6. Обратное разрешение адресов. Статические файлы. Работа с файлами Краткое содержание: обратное разрешению адресов (reverse proxy) и работе с файлами в веб-приложениях. Рассматриваются статические файлы, такие как HTML, CSS и JavaScript, и их обслуживание веб-сервером. Также изучается работа с**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе

	файлами в приложении, включая чтение, запись и загрузку файлов. Уметь: Настраивать обратный прокси-сервер для распределения нагрузки. Обслуживать статические файлы через веб-сервер. Читать и записывать файлы в приложении. Владеть: Навыками настройки и управления обратным прокси-сервером. Способами эффективной работы с файлами в приложении. Инструментами для тестирования и отладки при работе с файлами. /Лаб/						
1.25	**Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar Краткое содержание: рассматриваются классы ListView, DetailView и CreateView, являющиеся частью обобщенных представлений в Django. Также будут изучены улучшения, такие как Django Debug Toolbar, которая является полезным инструментом для отладки и оптимизации веб-приложений Django. Уметь: создавать представления на основе классов ListView, DetailView и CreateView; Настраивать атрибуты и методы этих классов в соответствии с требованиями приложения Владеть: Навыками работы с обобщенными представлениями в Django; Навыками настройки и использования классов ListView, DetailView и CreateView; Навыками установки и настройки Django Debug Toolbar /Лаб/**	8	1	0	0	ПКС-2.2	отчет по лабораторной работе
1.26	**Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация Краткое содержание: кастомизация администраторской панели в Django, включая настройку отображения моделей, полей, фильтров и действий. Также рассматриваются миксины - повторно используемые куски кода для представлений. Изучается постраничная навигация для отображения списков объектов по частям. Уметь: Регистрировать модели в администраторской панели. Настраивать отображение полей и списков моделей. Создавать пользовательские фильтры и действия Владеть: Навыками кастомизации интерфейса администраторской панели; Способностью создавать повторно используемый код с помощью миксинов /Лаб/**	8	1	0	0	ПКС-2.2,ПКС-2.3	отчет по лабораторной работе
	Раздел 2.Контроль						
2.1	**Экзамен Знать: базовые принципы контрактования в**	8	0	0	0	ПКС-2.1,ПКС-	вопросы к зачету, тестирование

	информационных технологиях, методы непрерывного улучшения управления проектами в области прикладных информационных технологий Уметь: осуществлять руководство проектами в области информационных технологий, организовывать деятельность по непрерывному улучшению управления проектами в области прикладных информационных технологий Владеть: базовыми навыками применения методов формирования и согласования принципов управления проектами в области информационных технологий /ЗаО/					2.2,ПКС-2.3	

Перечень применяемых активных и интерактивных образовательных технологий:
Информационные технологии

Личностно ориентированная технология, способ организации самостоятельной деятельности учащихся, направленный на решение задачи учебного проекта

Компьютерная технология обучения

Основана на использовании информационных технологий в учебном процессе. Реализация данной технологии осуществляется посредством компьютера и иных мультимедийных средств. Использование компьютерных технологий делает учебный процесс не только современным и познавательным, но интересным для обучающихся

Создание одностраничных Веб-сатов студенческого проекта

Обучение студентов основам Веб-технологий, работе в прикладных пакетах Adobe Photoshop и Gimp, работе в Веб-сервисе Figma и конструкторе сайтов Tilda для создания одностраничных Веб-сатов студенческого проекта

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная (внеаудиторная) работа обучающегося (далее - СРС) – это планируемая учебная, практическая, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время (свободное от аудиторных учебных занятий) по заданию и при методическом руководстве научно-педагогического работника (далее – преподаватель) и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, но без их непосредственного участия.
СРС по заданию и при методическом руководстве преподавателя и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, реализуется во время групповых консультаций и (или) индивидуальной работы обучающихся с преподавателями Университета и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях (в том числе индивидуальных консультаций), а также во время текущего контроля выполнения заданий, отнесенных к самостоятельной работе обучающихся.
Целью СРС является овладение формирование компетенций через овладение знаниями, умениями и навыками профессиональной деятельности по направлению подготовки (специальности). Самостоятельная работа является обязательной для каждого обучающегося.
Формы самостоятельной работы обучающихся определяются преподавателями кафедр Университета при разработке рабочих программ дисциплин (модулей), рабочих программ практик, программ государственной итоговой (итоговой) аттестации, методических указаний по выполнению практических, лабораторных работ, написанию курсовых работ/проектов и ВКР в соответствии с их содержанием.
В Университете оборудованы специальные помещения для самостоятельной работы обучающихся. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно -телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Перечень компетенций с указанием уровней их сформированности

ПКС-2:Способен формировать и согласовывать принципы управления проектами в области прикладных информационных технологий

Недостаточный уровень:

Знание базовые принципы контрактования в информационных технологиях, методы непрерывного улучшения управления проектами в области прикладных информационных технологий не сформировано

Умение осуществлять руководство проектами в области информационных технологий, организовывать деятельность по непрерывному улучшению управления проектами в области прикладных информационных технологий не сформировано

Владение базовыми навыками применения методов формирования и согласования принципов управления проектами в области информационных технологий не сформировано

Пороговый уровень:

Знает базовые принципы контрактования в информационных технологиях

Умеет осуществлять руководство проектами в области информационных технологий

Владеет базовыми навыками применения методов формирования принципов управления проектами

Продвинутый уровень:

Знает базовые принципы контрактования в информационных технологиях, методы непрерывного улучшения управления проектами

Умеет осуществлять руководство проектами в области информационных технологий, организовывать деятельность по непрерывному улучшению управления проектами

Владеет базовыми навыками применения методов формирования и согласования принципов управления проектами

Высокий уровень:

Знает базовые принципы контрактования в информационных технологиях, методы непрерывного улучшения управления проектами в области прикладных информационных технологий

Умеет осуществлять руководство проектами в области информационных технологий, организовывать деятельность по непрерывному улучшению управления проектами в области прикладных информационных технологий

Владеет базовыми навыками применения методов формирования и согласования принципов управления проектами в области информационных технологий

6.2. Описание критериев оценивания успеваемости

НЕДОСТАТОЧНЫЙ УРОВЕНЬ	ПОРОГОВЫЙ УРОВЕНЬ	ПРОДВИНУТЫЙ УРОВЕНЬ	ВЫСОКИЙ УРОВЕНЬ
Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие умения выполнять практические задания, предусмотренные программой дисциплины (модуля); - отсутствие готовности (способности) к дискуссии и низкая степень контактности.	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной программой дисциплины (модуля); - умение без грубых ошибок решать практические задания, которые следует выполнить.	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала; - способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; - правильные и конкретные, без грубых ошибок ответы на поставленные вопросы; - умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины (модуля); - наличие собственной обоснованной позиции по обсуждаемым вопросам; - незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность в ответах на дополнительные вопросы.	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и теории; - логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.
Оценка «незачет», «неудовлетворительно»	**Оценка «зачтено/удовлетворительно», «удовлетворительно»**	**Оценка «зачтено/хорошо», «хорошо»**	**Оценка «зачтено/отлично», «отлично»**

6.3. Оценочные средства текущего контроля

ВОПРОСЫ ДЛЯ УСТНОГО ОПРОСА

Тема 1. Введение в разработку веб-сайтов. Фреймворк Django

1. Что такое веб-фреймворк и для чего он используется?
2. Какие основные компоненты и принципы Django вы знаете?
3. Что такое архитектура MTV (Model-View-Template) и как она реализована в Django?

4. Какие преимущества использования Django по сравнению с разработкой веб-приложений с нуля?
5. Что такое проект и приложение в Django, и в чем их различие?
6. Как устанавливается Django и какие зависимости необходимы?
7. Что такое виртуальное окружение и зачем оно используется в Django?
8. Какие инструменты и библиотеки входят в экосистему Django?
9. Какие этапы необходимы для создания нового веб-приложения на Django?
10. Какие источники документации и обучающих материалов по Django вы можете порекомендовать?

Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции
1. Что такое шаблон MVS (Model-View-Serializer) и как он реализован в Django REST Framework?
2. Какова роль контроллеров (views) в Django и как они связаны с маршрутами (URLs)?
3. Как определяются маршруты в Django и какие виды маршрутов существуют?
4. Что такое модели в Django и для чего они используются?
5. Как создать новую модель и определить ее поля?
6. Что такое миграции в Django и для чего они необходимы?
7. Как применить миграции к базе данных и как отменить их?
8. Какие типы полей доступны в моделях Django и какие их основные параметры?
9. Как устанавливать отношения между моделями (один-к-одному, один-ко-многим, многие-ко-многим)?
10. Как выполнять запросы к моделям и фильтровать данные?

Тема 3. Шаблоны Django. Администраторская панель
1. Что такое шаблоны в Django и для чего они используются?
2. Какой синтаксис используется в шаблонах Django для отображения данных и логики?
3. Как использовать фильтры и теги в шаблонах для форматирования и манипуляции данными?
4. Как включить статические файлы (CSS, JavaScript) в шаблоны Django?
5. Что такое наследование шаблонов и как оно работает?
6. Для чего используется администраторская панель Django?
7. Как зарегистрировать модели в администраторской панели?
8. Какие настройки доступны для кастомизации администраторской панели?
9. Как создать пользовательские классы форм и списков для администраторской панели?
10. Как добавить пользовательские действия и виджеты в администраторской панели?

Тема 4. Связи моделей и виды. Директивы и теги
1. Какие типы связей между моделями существуют в Django и для чего они используются?
2. Как определить связь один-ко-многим и многие-ко-многим в моделях?
3. Что такое виды (views) в Django и какова их роль?
4. Как создать новый вид и отобразить данные из модели?
5. Какие способы рендеринга шаблонов существуют в видах?
6. Что такое директивы (directives) в шаблонах Django и какие основные директивы вы знаете?
7. Как использовать циклы и условные операторы в шаблонах с помощью директив?
8. Что такое теги (tags) в шаблонах Django и какие основные теги вы знаете?
9. Как создать пользовательские теги и использовать их в шаблонах?
10. Какие рекомендации по безопасности и производительности следует учитывать при работе с шаблонами?

Тема 5. Параметры в URL-запросах и имена маршрутов. Наследование шаблонов, пользовательские теги
1. Как передавать параметры в URL-запросах в Django?
2. Какие виды параметров в URL поддерживаются (позиционные, именованные, регулярные выражения)?
3. Для чего используются имена маршрутов и как их задавать?
4. Как обратиться к маршруту по его имени в шаблонах и видах?
5. Что такое наследование шаблонов и какие преимущества оно дает?
6. Как определить базовый шаблон и блоки для наследования?
7. Как расширить базовый шаблон в дочернем шаблоне?
8. Что такое пользовательские теги в Django и для чего они используются?
9. Как создать простой пользовательский тег и зарегистрировать его?
10. Какие примеры использования пользовательских тегов вы можете привести?

Тема 6. Обратное разрешение адресов. Статические файлы. Работа с файлами
1. Что такое обратное разрешение адресов (URL reverse) в Django и для чего оно используется?
2. Как получить URL-адрес маршрута по его имени в шаблонах и видах?
3. Что такое статические файлы (CSS, JavaScript, изображения) и как их включать в Django-проект?
4. Как настроить обработку статических файлов для разных сред (разработка, продакшн)?
5. Как загружать и обрабатывать файлы в Django (например, изображения)?
6. Какие поля и параметры необходимо использовать в моделях для работы с файлами?
7. Как сохранять и удалять загруженные файлы в Django?
8. Как отображать загруженные файлы в шаблонах?
9. Какие рекомендации следует учитывать при работе с файлами в веб-приложениях?
10. Как обрабатывать ошибки и ограничения при загрузке файлов?

Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar

1. Что такое обобщенные представления (generic views) в Django и для чего они используются?
2. Как работает класс ListView и какие параметры он принимает?
3. Как использовать класс DetailView для отображения детальной информации о записи?
4. Как создать форму для создания новой записи с помощью класса CreateView?
5. Какие другие обобщенные представления доступны в Django и для каких задач они предназначены?
6. Что такое Django Debug Toolbar и для чего она используется?
7. Как установить и настроить Django Debug Toolbar в проекте?
8. Какие панели и инструменты доступны в Django Debug Toolbar?
9. Как использовать Django Debug Toolbar для отладки и оптимизации?
10. Какие альтернативы Django Debug Toolbar существуют?

Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация
1. Как кастомизировать внешний вид администраторской панели Django (логотип, заголовки, цвета и т.д.)?
2. Как создать пользовательские действия в администраторской панели?
3. Как настроить фильтры и поиск для моделей в администраторской панели?
4. Что такое миксины в Django и для чего они используются?
5. Как создать простой миксин и использовать его в представлении?
6. Какие встроенные миксины доступны в Django и для каких задач они предназначены?
7. Как реализовать постраничную навигацию (пагинацию) в Django?
8. Какие параметры и настройки доступны для пагинации?
9. Как отображать ссылки на следующую и предыдущую страницы в шаблонах?
10. Как кастомизировать внешний вид и поведение постраничной навигации?

ВОПРОСЫ ДЛЯ САМОПОДГОТОВКИ
Тема 1. Введение в разработку веб-сайтов. Фреймворк Django
1. Что такое веб-фреймворк и какие преимущества он дает по сравнению с разработкой с нуля?
2. Опишите архитектуру Model-View-Template (MVT) Django и роль каждого компонента.
3. Как создать новый проект и приложение в Django?
4. Что такое ORM и как он используется в Django?
5. Какие базовые концепции веб-разработки необходимо знать для работы с Django?
6. Как настроить среду разработки для работы с Django?
7. Что такое виртуальное окружение и зачем оно нужно?
8. Какие основные компоненты входят в экосистему Django?
9. Как отлаживать и устранять неполадки в приложениях Django?
10. Какие источники документации и обучающих материалов по Django вы можете порекомендовать?

Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции
1. В чем заключается концепция шаблона MVS (Model-View-Serializer)?
2. Какова роль контроллеров (views) в Django?
3. Как определяются маршруты (URLs) в Django?
4. Что такое модели в Django и для чего они используются?
5. Как создать новую модель и определить ее поля?
6. Что такое миграции и для чего они нужны?
7. Как применить миграции к базе данных?
8. Какие типы полей доступны в моделях Django?
9. Как устанавливать отношения между моделями (один-к-одному, один-ко-многим, многие-ко-многим)?
10. Как выполнять запросы к моделям и фильтровать данные?

Тема 3. Шаблоны Django. Администраторская панель
1. Какой синтаксис используется в шаблонах Django для отображения данных?
2. Что такое наследование шаблонов и как его реализовать?
3. Какие фильтры и теги доступны в шаблонах Django для форматирования данных?
4. Как включить статические файлы (CSS, JavaScript) в шаблоны?
5. Для чего используется администраторская панель Django?
6. Как зарегистрировать модели в администраторской панели?
7. Какие настройки доступны для кастомизации администраторской панели?
8. Как создать пользовательские формы и списки для администраторской панели?
9. Как добавить пользовательские действия и фильтры в администраторской панели?
10. Какие преимущества дает использование администраторской панели в процессе разработки?

Тема 4. Связи моделей и виды. Директивы и теги
1. Какие типы связей между моделями существуют в Django?
2. Как определить связь один-ко-многим в моделях?
3. Как определить связь многие-ко-многим в моделях?
4. Что такое виды (views) в Django и какова их роль?
5. Как создать новый вид и отобразить данные из модели?
6. Что такое директивы в шаблонах Django и какие основные директивы вы знаете?
7. Как использовать циклы и условные операторы в шаблонах с помощью директив?
8. Что такое теги в шаблонах Django и какие основные теги вы знаете?

9. Как создать пользовательский тег и использовать его в шаблонах?
10. Какие рекомендации по безопасности и производительности следует учитывать при работе с шаблонами?

Тема 5. Параметры в URL-запросах и имена маршрутов. Наследование шаблонов, пользовательские теги
1. Как передавать параметры в URL-запросах в Django?
2. Какие виды параметров в URL поддерживаются (позиционные, именованные, регулярные выражения)?
3. Для чего используются имена маршрутов и как их задавать?
4. Как обратиться к маршруту по его имени в шаблонах и видах?
5. Что такое наследование шаблонов и какие преимущества оно дает?
6. Как определить базовый шаблон и блоки для наследования?
7. Как расширить базовый шаблон в дочернем шаблоне?
8. Что такое пользовательские теги в Django и для чего они используются?
9. Как создать простой пользовательский тег и зарегистрировать его?
10. Какие примеры использования пользовательских тегов вы можете привести?

Тема 6. Обратное разрешение адресов. Статические файлы. Работа с файлами
1. Что такое обратное разрешение адресов (URL reverse) в Django и для чего оно используется?
2. Как получить URL-адрес маршрута по его имени в шаблонах и видах?
3. Что такое статические файлы (CSS, JavaScript, изображения) и как их включать в Django-проект?
4. Как настроить обработку статических файлов для разных сред (разработка, продакшн)?
5. Как загружать и обрабатывать файлы в Django (например, изображения)?
6. Какие поля и параметры необходимо использовать в моделях для работы с файлами?
7. Как сохранять и удалять загруженные файлы в Django?
8. Как отображать загруженные файлы в шаблонах?
9. Какие рекомендации следует учитывать при работе с файлами в веб-приложениях?
10. Как обрабатывать ошибки и ограничения при загрузке файлов?

Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar
1. Что такое обобщенные представления (generic views) в Django и для чего они используются?
2. Как работает класс ListView и какие параметры он принимает?
3. Как использовать класс DetailView для отображения детальной информации о записи?
4. Как создать форму для создания новой записи с помощью класса CreateView?
5. Какие другие обобщенные представления доступны в Django и для каких задач они предназначены?
6. Что такое Django Debug Toolbar и для чего она используется?
7. Как установить и настроить Django Debug Toolbar в проекте?
8. Какие панели и инструменты доступны в Django Debug Toolbar?
9. Как использовать Django Debug Toolbar для отладки и оптимизации приложения?
10. Какие альтернативы Django Debug Toolbar существуют и в чем их отличия?

Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация
1. Как кастомизировать внешний вид администраторской панели Django (логотип, заголовки, цвета и т.д.)?
2. Как создать пользовательские действия в администраторской панели?
3. Как настроить фильтры и поиск для моделей в администраторской панели?
4. Что такое миксины в Django и для чего они используются?
5. Как создать простой миксин и использовать его в представлении?
6. Какие встроенные миксины доступны в Django и для каких задач они предназначены?
7. Как реализовать постраничную навигацию (пагинацию) в Django?
8. Какие параметры и настройки доступны для пагинации?
9. Как отображать ссылки на следующую и предыдущую страницы в шаблонах?
10. Как кастомизировать внешний вид и поведение постраничной навигации?

Тема 9. Регистрация и авторизация
1. Какие принципы работы системы аутентификации в Django вы знаете?
2. Как устроена структура стандартной модели пользователя (User) в Django?
3. Что такое сессии и какую роль они играют в процессе авторизации?
4. Как настроить параметры аутентификации в settings.py?
5. Как создать форму регистрации нового пользователя?
6. Как создать форму входа (логина) для авторизации пользователя?
7. Как проверять авторизацию пользователя в представлениях?
8. Какие методы и декораторы используются для ограничения доступа в Django?
9. Как кастомизировать модель пользователя в Django?
10. Как интегрировать систему аутентификации в свое веб-приложение?

Тема 10. CKEditor, капча. Кэширование
1. Что такое CKEditor и для чего он используется в веб-разработке?
2. Как интегрировать CKEditor в веб-приложение Django?
3. Какие основные настройки и опции доступны в CKEditor?
4. Что такое капча (CAPTCHA) и для чего она используется?
5. Как внедрить и обрабатывать капчу в формах Django?

6. Какие типы капчи существуют и в чем их отличия?
7. Что такое кэширование и для чего оно нужно в веб-приложениях?
8. Какие виды кэширования доступны в Django?
9. Как настроить и использовать кэширование в Django?
10. Какие стратегии и подходы к кэшированию вы можете рекомендовать?

ТЕСТИРОВАНИЕ ТЕКУЩЕГО КОНТРОЛЯ

Тема 1. Введение в разработку веб-сайтов. Фреймворк Django

1. Что такое веб-фреймворк? a) Библиотека для создания веб-сайтов b) Набор инструментов и компонентов для быстрой разработки веб-приложений c) Язык программирования для веб-разработки d) Система управления базами данных
2. Что такое архитектура MVT в Django? a) Model-View-Template - разделение на модели данных, представления и шаблоны b) Model-View-Table - работа с базами данных через ORM c) Model-View-Test - подход для тестирования приложений d) Model-View-Translation - система для локализации
3. Что такое ORM в Django? a) Объектно-реляционное отображение для работы с базами данных b) Обратное разрешение маршрутов c) Механизм очередей для асинхронных задач d) Система регистрации и аутентификации пользователей
4. Что нужно сделать для создания нового проекта в Django? a) django-admin startproject имя_проекта b) python manage.py makemigrations c) python manage.py startapp имя_приложения d) pip install django
5. Какой инструмент в Django используется для отладки приложений? a) Django Debug Toolbar b) Django Admin c) Django ORM d) Django Forms
6. Какой язык программирования используется в Django? a) Java b) C++ c) Python d) Ruby
7. Что такое виртуальное окружение в Python? a) Интегрированная среда разработки b) Механизм изолированной установки пакетов для проекта c) Система контроля версий d) Облачная среда для запуска приложений
8. Какой компонент отвечает за логику приложения в шаблоне MVT? a) Model b) View c) Template d) Form
9. Какой инструмент используется для создания новых приложений в Django проекте? a) django-admin startapp имя_приложения b) python manage.py makeapp имя_приложения c) pip install имя_приложения d) django-admin createapp имя_приложения
10. Что такое Django REST framework? a) Фреймворк для создания REST API b) Система кэширования данных c) Механизм обработки форм d) Административная панель

Тема 2. Шаблон MVS, контроллеры и маршруты. Модели и миграции

1. Что означает MVS в контексте Django REST Framework? a) Model-View-Serializer b) Model-View-Service c) Model-View-Session d) Model-View-Security
2. Что делают контроллеры (views) в Django? a) Отображают данные в шаблонах b) Определяют структуру данных c) Обрабатывают HTTP-запросы и выполняют логику приложения d) Управляют авторизацией пользователей
3. Как определяются маршруты (URLs) в Django? a) В файле models.py b) В файле views.py c) В файле urls.py d) В файле settings.py
4. Что такое модели в Django? a) Шаблоны для отображения данных b) Классы, описывающие структуру данных в базе данных c) HTTP-обработчики для маршрутов d) Формы для ввода данных
5. Как создать новую модель в Django? a) Создать класс, наследующийся от models.Model b) Создать функцию в файле views.py c) Создать шаблон в templates/ d) Создать форму в forms.py
6. Что такое миграции в Django? a) Система контроля версий b) Скрипты для применения изменений в моделях к базе данных c) Механизм кэширования данных d) Процесс перевода приложения на другие языки
7. Как применить миграции к базе данных? a) python manage.py makemigrations b) python manage.py migrate c) python manage.py applymigrations d) python manage.py updatemigrations
8. Какой тип поля в моделях Django используется для внешнего ключа? a) ForeignKey b) PrimaryKey c) CharField d) IntegerField
9. Как определить связь многие-ко-многим в моделях Django? a) OneToOneField b) ForeignKey c) ManyToManyField d) RelationField
10. Что такое менеджеры моделей в Django? a) Инструменты для управления базами данных b) Классы для выполнения операций с данными модели c) Компоненты для отображения данных в шаблонах d) Механизм аутентификации пользователей

Тема 3. Шаблоны Django. Администраторская панель

1. Какой синтаксис используется для вывода переменной в шаблонах Django? a) {% variable %} b) {{ variable }} c) (variable) d) [variable]
2. Как создать цикл в шаблоне Django? a) {% for item in list %} ... {% endfor %} b) {{ for item in list }} ... {{ endfor }} c) (for item in list) ... (endfor) d) [for item in list] ... [endfor]
3. Что такое наследование шаблонов в Django? a) Система импорта шаблонов из других проектов b) Возможность создавать базовый шаблон и расширять его в дочерних шаблонах c) Механизм кэширования шаблонов d) Способ локализации текста в шаблонах
4. Как включить CSS-файл в шаблон Django? a) {% include "styles.css" %} b) {{ load_css "styles.css" }} c) {% load static %} {% static "css/styles.css" %} d) {{ css_file "styles.css" }}
5. Для чего используется администраторская панель Django? a) Для создания и управления контентом веб-сайта b) Для отображения данных в шаблонах c) Для редактирования кода приложения d) Для управления базами данных
6. Как зарегистрировать модель в администраторской панели Django? a) Добавить ее в файл models.py b) Добавить ее в файл views.py c) Добавить ее в файл admin.py и зарегистрировать с помощью admin.site.register(Model) d) Добавить ее в файл urls.py
7. Какой атрибут используется для настройки полей, отображаемых в списке модели в админке? a) list_view b) display_list c) list_display d) show_fields
8. Как создать пользовательскую форму для модели в администраторской панели? a) Определить класс формы в

2. Какой веб-сервер рекомендуется использовать для обслуживания статических файлов в Django? a) Gunicorn b) Apache c) Nginx d) Все ответы верны
3. Как настроить обслуживание статических файлов в Django? a) Через настройки STATIC_URL и STATIC_ROOT b) Через настройки MEDIA_URL и MEDIA_ROOT c) Через настройку DEBUG=True d) Через настройку ALLOWED_HOSTS
4. Какая функция используется для открытия файла на чтение в Django? a) open() b) read() c) write() d) file()
5. Какая функция используется для записи данных в файл в Django? a) open() b) read() c) write() d) file()
6. Как загрузить файл, присланный пользователем, в Django? a) Через поле FileField в модели b) Через настройку MEDIA_ROOT c) Через функцию open() d) Через декоратор @file_upload
7. Какой модуль в Django используется для работы с файловой системой? a) os b) sys c) file d) filesystem
8. Что такое хендлер в Django для обработки загруженных файлов? a) Функция для валидации загруженного файла b) Поле в модели для хранения пути к файлу c) Настройка для указания директории для загрузки файлов d) Декоратор для обработки загруженных файлов
9. Какой способ рекомендуется для удаления файлов в Django? a) Удалить файл через функцию os.remove() b) Удалить файл через метод delete() модели c) Удалить файл через администраторскую панель d) Все ответы верны
10. Как обеспечить безопасность при работе с файлами в Django? a) Использовать валидацию загруженных файлов b) Ограничить типы загружаемых файлов c) Проверять права доступа к файлам d) Все ответы верны

Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar
1. Что такое обобщенные представления (generic views) в Django? a) Встроенные классы для создания типовых представлений b) Функции для работы с базой данных c) Механизм для создания пользовательских тегов d) Система кэширования
2. Какой класс используется для создания представления со списком объектов? a) ListView b) DetailView c) CreateView d) UpdateView
3. Какой атрибут используется в классе ListView для указания модели? a) model b) queryset c) template_name d) context_object_name
4. Что такое Django Debug Toolbar? a) Панель инструментов для отладки и оптимизации приложения b) Механизм для создания пользовательских тегов c) Система кэширования d) Обратный прокси-сервер
5. Как включить Django Debug Toolbar в проекте? a) Добавить ее в INSTALLED_APPS и внутреннем среднем файле urls.py b) Создать файл debug_toolbar.py и добавить его в корневую директорию проекта c) Использовать команду manage.py debug_toolbar d) Установить отдельный веб-сервер для Debug Toolbar
6. Какие данные отображает Django Debug Toolbar? a) Информацию о запросах к базе данных b) Заголовки HTTP-запросов и ответов c) Время выполнения представлений d) Все ответы верны
7. Какой класс используется для создания представления с деталями объекта? a) ListView b) DetailView c) CreateView d) UpdateView
8. Какой атрибут используется в классе DetailView для указания модели? a) model b) queryset c) template_name d) context_object_name
9. Что такое контекстный процессор в Django? a) Функция, добавляющая данные в контекст шаблона b) Механизм кэширования шаблонов c) Средство для создания пользовательских тегов d) Компонент для обработки форм
10. Как создать пользовательский контекстный процессор в Django? a) Создать функцию в файле views.py b) Создать функцию в файле models.py c) Создать функцию в директории context_processors d) Создать класс в файле forms.py

Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация
1. Как зарегистрировать модель в администраторской панели Django? a) Создать файл admin.py в приложении и добавить модель в admin.site.register() б) Добавить модель в файл models.py в) Создать файл views.py и добавить модель в представлении г) Создать файл urls.py и добавить маршрут для модели
2. Как изменить порядок отображения полей модели в администраторской панели? a) Использовать атрибут fields в классе ModelAdmin б) Использовать атрибут ordering в классе ModelAdmin в) Использовать атрибут list_display в классе ModelAdmin г) Использовать атрибут list_filter в классе ModelAdmin
3. Какой атрибут используется для добавления поиска по полям модели в администраторской панели? a) search_fields б) list_filter в) ordering г) fields
4. Что такое миксин в Django? a) Класс, предоставляющий повторно используемый код для представлений б) Функция для обработки форм в) Тег для отображения данных в шаблоне г) Модуль для работы с базой данных
5. Как создать пользовательский миксин в Django? a) Создать класс, наследующийся от django.views.generic.base.View б) Создать функцию в файле views.py в) Создать класс, наследующийся от django.forms.Form г) Создать тег в директории templatetags
6. Что такое постраничная навигация в Django? a) Механизм для отображения списка объектов по частям б) Система кэширования в) Средство для создания пользовательских тегов г) Компонент для обработки форм
7. Какой класс используется для постраничной навигации в Django? a) Paginator б) ListView в) DetailView г) CreateView
8. Как добавить пользовательское действие в администраторской панели Django? a) Определить функцию в классе ModelAdmin и добавить ее в атрибут actions б) Создать файл actions.py и добавить функцию в) Использовать декоратор @action для представления г) Добавить действие в файл urls.py
9. Какой атрибут используется для добавления фильтров в боковой панели администраторской панели? a) search_fields б) list_filter в) ordering г) fields
10. Как изменить заголовок модели в администраторской панели? a) Использовать атрибут verbose_name в классе Model б) Использовать атрибут verbose_name_plural в классе Model в) Использовать атрибут list_display в классе ModelAdmin г) Использовать атрибут search_fields в классе ModelAdmin

ЗАДАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ
Тема 1. Введение в разработку веб-сайтов. Фреймворк Django
1. Создать в проекте еще одно приложение с произвольным названием, зарегистрировать его в settings.py в константе INSTALLED_APPS

2.Собрать статику
3.Добавить маршрут для страницы с формой добавления записи в модель Human
4.Создать в приложении файл forms.py и в нем описать форму добавления записи
5.Создать контроллер для данной формы и страницы
6.Создать шаблон для формы добавления новости

Тема 7. Классы ListView, DetailView, CreateView. Улучшения и Django Debug Toolbar
1.Улучшить шаблоны для страниц взаимодействия с моделями Human и Profession как посчитаете необходимым(в частно с применением классов bootstrap)
2.При помощи Django Debug Toolbar просмотрите, как работают ваши дополнительные страницы(оценить кол-во запросов, какие шаблоны используются и т. д.)
3.Уменьшите кол-во запросов к базе данных(модифицируйте метод get_queryset())

Тема 8. Кастомизация администраторской панели и миксины. Постраничная навигация
1.При помощи файла admin.css сделайте больше изменений оформления страницы администратора. Для просмотра элементов, с которыми необходимо взаимодействовать для их изменений, откройте страницу админки, нажмите правой

6.4. Оценочные средства промежуточной аттестации
ВОПРОСЫ К ЗАЧЕТУ
Индикатор достижения компетенции Знать:
Задания на установление последовательности
Задание 1. Установите правильную последовательность обработки запроса в архитектуре MVT фреймворка Django: А) Рендеринг шаблона с данными Б) Обработка URL и вызов представления В) Получение запроса от пользователя Г) Взаимодействие с моделью для получения данных Д) Возврат HTTP-ответа клиенту
Задание 2. Установите правильную последовательность шагов при создании новой модели в Django: А) Регистрация модели в административной панели Б) Создание класса модели в файле models.py В) Создание миграции с помощью makemigrations Г) Применение миграции командой migrate Д) Определение полей и отношений модели
Задание 3. Установите правильную последовательность этапов аутентификации пользователя в Django: А) Проверка учетных данных Б) Создание сессии для пользователя В) Получение формы с учетными данными Г) Перенаправление на защищенную страницу Д) Валидация полученных данных
Задание 4. Установите правильную последовательность настройки статических файлов в Django: А) Создание директории для статических файлов Б) Настройка STATIC_URL в settings.py В) Использование тега {% static %} в шаблонах Г) Добавление 'django.contrib.staticfiles' в INSTALLED_APPS Д) Определение STATICFILES_DIRS в settings.py
Задание 5. Установите правильную последовательность настройки обратного прокси-сервера для Django-приложения: А) Конфигурация веб-сервера (Nginx/Apache) Б) Настройка WSGI-сервера (Gunicorn/uWSGI) В) Тестирование конфигурации Г) Настройка Django для работы за прокси Д) Включение проксирования запросов к Django-приложению

Задания на установление соответствия
Задание 1. Установите соответствие между компонентами Django и их определениями:
1. Model
2. View
3. Template
4. URL dispatcher
5. Middleware
А) Компонент, отвечающий за обработку HTTP-запросов и возврат HTTP-ответов Б) Система перехвата запросов и ответов для выполнения глобальных операций В) Компонент для описания структуры данных и бизнес-логики приложения Г) HTML-файл с встроенным языком шаблонов для динамического формирования страниц Д) Система сопоставления URL-адресов с функциями представлений
Задание 2. Установите соответствие между классами Django и их назначением:
1. ListView
2. DetailView
3. CreateView
4. UpdateView
5. DeleteView
А) Класс для отображения подробной информации об объекте Б) Класс для удаления существующего объекта В) Класс для отображения списка объектов Г) Класс для создания новых объектов Д) Класс для изменения существующего объекта
Задание 3. Установите соответствие между типами связей моделей в Django и их характеристиками:
1. OneToOneField
2. ForeignKey
3. ManyToManyField
4. GenericForeignKey
5. Self-referential relationship
А) Связь, позволяющая модели ссылаться на экземпляры самой себя Б) Связь, позволяющая объекту одной модели быть связанным с несколькими объектами другой модели В) Связь, создающая отношение "один к одному" между моделями Г) Связь, позволяющая ссылаться на объекты различных моделей Д) Связь, создающая отношение "многие к одному" между моделями
Задание 4. Установите соответствие между компонентами Django Debug Toolbar и их функциями:
1. SQL panel
2. Timer panel
3. Request panel
4. Template panel
5. Cache panel

Задание 2
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Что такое ORM в Django?
A) Операционная система для разработки B) Объектно-реляционное отображение для работы с базой данных C) Метод оптимизации серверных ресурсов D) Формат хранения статических файлов
Задание 3
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Как в Laravel создается новый контроллер с помощью Artisan?
A) php artisan create:controller UserController B) php artisan make:controller UserController C) php artisan new:controller UserController D) php artisan generate:controller UserController
Задание 4
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Какой файл в Django отвечает за определение маршрутов приложения?
A) models.py B) views.py C) urls.py D) settings.py
Задание 5
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
С помощью какого тега в Django создаются блоки в шаблонах?
A) {% section %} B) {% block %} C) {% include %} D) {% extends %}
Задание 6
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Какая модель связи в Django используется для связи "многие ко многим"?
A) ForeignKey B) OneToOneField C) ManyToManyField D) RelatedField
Задание 7
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Как правильно передать данные через параметры URL в Django?
A) path('articles/<slug:slug>/', views.article_detail) B) path('articles/{slug}/', views.article_detail) C) path('articles/:slug/', views.article_detail) D) path('articles/get?slug=', views.article_detail)
Задание 8
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Какая функция используется для создания именованных маршрутов в Django?
A) named_path() B) path() с параметром name C) url_name() D) route()
Задание 9
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Какое представление в Django используется для отображения списка объектов?
A) DetailView B) FormView C) ListView D) TemplateView
Задание 10
Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор.
Как в Django настроить статические файлы?
A) Определить переменную STATIC_URL в settings.py B) Использовать команду python manage.py collectstatic C) Создать директорию static в корне проекта D) Все вышеперечисленное

Индикатор достижения компетенции Владеть:
1. Прочитайте текст и запишите развернутый ответ. Вы разрабатываете веб-приложение на Django для блога, где пользователи могут публиковать статьи. Необходимо создать модель для статей и связать ее с моделью пользователей. Опишите, какие виды связей вы будете использовать и как их реализовать.
2. Прочитайте текст и запишите развернутый ответ. В вашем Django-проекте требуется создать страницу с списком всех категорий продуктов. Каждая категория должна быть представлена ссылкой, которая ведет на отдельную страницу со списком продуктов этой категории. Опишите, какие шаги вы предпримете для реализации этой функциональности.
3. Прочитайте текст и запишите развернутый ответ. Вы работаете над формой регистрации пользователей в Django-приложении. Необходимо добавить капчу для защиты от ботов. Опишите, как вы будете внедрять и обрабатывать капчу в этой форме.
4. Прочитайте текст и запишите развернутый ответ. В вашем веб-приложении на Django есть страница со списком всех зарегистрированных пользователей. Количество пользователей постоянно растет, и вам необходимо реализовать постраничную навигацию для более удобного отображения списка. Опишите, как вы будете это делать.
5. Прочитайте текст и запишите развернутый ответ. Вы создаете веб-сайт для блога на Django, где пользователи могут оставлять комментарии к статьям. Требуется реализовать систему авторизации, чтобы только зарегистрированные пользователи могли оставлять комментарии. Опишите, какие шаги вы предпримете для реализации этой функциональности.
6. Прочитайте текст и запишите развернутый ответ. В вашем Django-приложении есть форма для создания новых объектов. Вы хотите добавить визуальный редактор HTML (CKEditor) для поля описания. Опишите, как вы будете интегрировать CKEditor в эту форму.
7. Прочитайте текст и запишите развернутый ответ. Вы разрабатываете веб-приложение на Django для интернет-магазина. Необходимо реализовать функциональность добавления товаров в корзину покупок. Опишите, какие модели и связи между ними вам понадобятся для этого, а также как вы будете работать с сессиями для хранения данных корзины.
8. Прочитайте текст и запишите развернутый ответ. В вашем Django-проекте есть страница с детальной информацией о продукте. Вы хотите добавить возможность загрузки файлов, например, руководств пользователя или технических спецификаций. Опишите, как вы будете реализовывать эту функциональность, включая обработку загруженных файлов и их хранение.
9. Прочитайте текст и запишите развернутый ответ. Вы работаете над админ-панелью Django для управления объектами вашего приложения. Необходимо настроить отображение списка объектов с возможностью фильтрации и сортировки по определенным полям. Опишите, как вы будете это делать.
10. Прочитайте текст и запишите развернутый ответ. В вашем Django-приложении есть форма для создания нового объекта. Вы хотите добавить пользовательский фильтр в шаблон, который будет преобразовывать текст в заглавные буквы

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. В каком файле определяются URL-маршруты в Django?
• a) models.py
• b) urls.py
• c) views.py
• d) templates.py

Вопрос 5

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое миграции в Django?
• a) Перемещение данных между базами данных
• b) Изменения в структуре базы данных
• c) Обновление версии Django
• d) Процесс развертывания приложения

Вопрос 6

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Какой шаблон используется в Laravel?
• a) MVT
• b) MVC
• c) MVS
• d) MVP

Вопрос 7

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое Django Admin?
• a) Панель администрирования для управления моделями данных
• b) Интерфейс для взаимодействия с базой данных
• c) Инструмент для создания форм
• d) Библиотека для тестирования

Вопрос 8

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Какая директива в Django ORM используется для создания одно-к-одной связи между моделями?
• a) ForeignKey
• b) ManyToManyField
• c) OneToOneField
• d) RelatedField

Вопрос 9

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое миксины в Django?
• a) Классы, предоставляющие повторно используемый код
• b) Функции для обработки HTTP-запросов
• c) Шаблоны для отображения данных
• d) Модули для работы с базами данных

Вопрос 10

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое ListView в Django?
• a) Класс для отображения списка объектов
• b) Класс для отображения деталей одного объекта
• c) Класс для создания новых объектов
• d) Класс для редактирования существующих объектов

Вопрос 11

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Какая директива в Django ORM используется для создания многие-ко-многим связи между моделями?
• a) ForeignKey
• b) ManyToManyField
• c) OneToOneField
• d) RelatedField

Вопрос 12

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое тег в шаблонах Django?
• a) Конструкция для вывода данных
• b) Конструкция для выполнения логики
• c) Конструкция для создания циклов и условий
• d) Всё вышеперечисленное

Вопрос 13

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Какой тег в Django используется для создания циклов?
• a) {% loop %}
• b) {% for %}
• c) {% iterate %}
• d) {% cycle %}

Вопрос 14

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое наследование шаблонов в Django?
• a) Механизм для создания шаблонов на основе других шаблонов
• b) Механизм для вложенности шаблонов друг в друга
• c) Механизм для переопределения блоков в базовом шаблоне
• d) Всё вышеперечисленное

Вопрос 15

Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор. Что такое капча (CAPTCHA) в веб-

• c) Система для авторизации пользователей
• d) Система для кэширования данных

6.5. Примерная тематика курсовых работ (проектов)
Учебным планом не предусмотрено

6.6. Методические указания для обучающихся по освоению дисциплины (модуля)
Методические рекомендации по подготовке к лекционным занятиям

Просмотрите конспект сразу после занятий. Пометьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.
Работа с рекомендованной литературой:
При работе с основной и дополнительной литературой целесообразно придерживаться такой последовательности. Сначала прочитать весь заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом. Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать. План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала. Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов: - план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения, - текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника, - свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом, - тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу. В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации по подготовке к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение практического занятия предполагает, например:
- индивидуальные выступления студентов с сообщениями по какому-либо вопросу изучаемой темы;
- фронтальное обсуждение рассматриваемой проблемы, обобщения и выводы;
- решение задач и упражнений по образцу;
- решение вариантных задач и упражнений;
- решение ситуационных производственных (профессиональных) задач;
- проектирование и моделирование разных видов и компонентов профессиональной деятельности;
- выполнение контрольных работ;
- работу с тестами.
При подготовке к практическим занятиям студентам рекомендуется: внимательно ознакомиться с тематикой практического занятия; прочесть конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; проверить свои знания, отвечая на вопросы для самопроверки; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в тетради. Все письменные задания выполнять в рабочей тетради. Практические занятия развивают у студентов навыки самостоятельной работы по решению конкретных задач.

Методические рекомендации по подготовке к лабораторным работам

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в изучаемой дисциплине (модуле). Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение лабораторной работы предполагает:
- изучение теоретического материала по теме лабораторной работы (по вопросам изучаемой темы);
- выполнение необходимых расчетов и экспериментов;
- оформление отчета с заполнением необходимых таблиц, построением графиков, подготовкой выводов по проделанным экспериментам и теоретическим расчетам;
- по каждой лабораторной работе проводится контроль: проверяется содержание отчета, проверяется усвоение

теоретического материала.
Контроль усвоения теоретического материала является индивидуальным.

Методические указания по выполнению отчёта к лабораторным работам

Основным требованием по выполнению лабораторных и практических работ является полное исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения и профессиональной подготовки студентов.
Методические указания обеспечивают комплексный подход в учебной работе студентов, единство и преемственность требований к оформлению результатов работы на разных этапах обучения. С единых позиций приведены основные требования к структуре, оформлению и содержанию отчета по лабораторным и практическим работам.
Структура отчёта:
- цель работы;
- краткие теоретические сведения;
- ход выполнения работы;
- выводы.
Дополнительные элементы:
- приложения;
- библиографический список.
Требования к содержанию отчёта:
1. Титульный лист
В верхнем поле листа указывают полное наименование учебного заведения.
В среднем поле указывается вид работы, в данном случае лабораторная или практическая работа с указанием курса, по которому она выполнена, и ниже ее название. Название работы приводится без слова "тема" и в кавычки не заключается.
Далее ближе к правому краю титульного листа указывают фамилию, инициалы и группу учащегося, выполнившего работу, а также фамилию, инициалы преподавателя, принявшего работу.
В нижнем поле листа указывается место выполнения работы и год ее написания (без слова год).
2. Цель работы должна отражать тему работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до 0,5 страницы.
3. Краткие теоретические сведения. В этом разделе излагается краткое теоретическое описание изучаемой в работе темы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для дальнейшей обработки полученных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.
4. Ход выполнения работы. В данном разделе подробно излагается методика выполнения работы, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.
5. Выводы по работе - кратко излагаются результаты работы, полученные в результате выполнения работы, а также краткий анализ полученных результатов.
Отчет по лабораторной работе оформляется на листе формата А4. Допускается оформление отчета по лабораторной работе в электронном виде средствами Microsoft Office. Текст работы должен быть напечатан через полтора интервала шрифтом Times New Roman, кегль – 12. Поля должны оставаться по всем четырем сторонам печатного листа: левое – не менее 30 мм, правое – не менее 10, нижнее – не менее 20 и верхнее – не 15 мм.
Для защиты лабораторной работы студент должен подготовить отчет, провести самостоятельную работу, иметь отметку о проверенном отчете.
Результаты определяются по пятибалльной системе оценок.

Методические рекомендации по выполнению реферата

Реферат – письменная работа объемом 8–10 страниц. Это краткое и точное изложение сущности какого-либо вопроса, темы. Тему реферата студент выбирает из предложенных преподавателем или может предложить свой вариант. В реферате нужны развернутые аргументы, рассуждения, сравнения. Содержание темы излагается объективно от имени автора. Функции реферата: информативная, поисковая, справочная, сигнальная, коммуникативная. Степень выполнения этих функций зависит от содержательных и формальных качеств реферата и для каких целей их использует. Требования к языку реферата: должен отличаться точностью, краткостью, ясностью и простотой.
Структура реферата:
1. Титульный лист
2. Оглавление (на отдельной странице). Указываются названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. Введение. Аргументируется актуальность исследования, т.е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками, перечисляются положения, которые должны быть обоснованы. Обязательно формулируются цель и задачи реферата.
4. Основная часть. Подчиняется собственному плану, что отражается в разделении текста на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала. В случае если используется чья-либо неординарная мысль, идея, то обязательно нужно сделать ссылку на того автора, у кого взят данный материал.
5. Заключение. Последняя часть научного текста. В краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования.

6. Приложение. Может включать графики, таблицы, расчеты.
7. Библиография (список литературы). Указывается реально использованная для написания реферата литература. Названия книг располагаются по алфавиту с указанием их выходных данных. Общие требования к построению, содержанию и оформлению».
При проверке реферата оцениваются:
- знание фактического материала, усвоение общих представлений, понятий, идей;
- характеристика реализации цели и задач исследования;
- степень обоснованности аргументов и обобщений;
- качество и ценность полученных результатов;
- использование литературных источников;
- культура письменного изложения материала;
- культура оформления материалов работы.
Правила написания научных текстов (реферат):
Здесь приводятся рекомендации по консультированию студентов относительно данного вида самостоятельной работы. Во время консультаций руководителю следует предложить к обсуждению следующие вопросы.
• Какова истинная цель Вашего научного текста – это поможет Вам разумно распределить свои силы и время.
• Важно разобраться, кто будет «читателем» Вашей работы.
• Начинать писать серьезную работу следует не раньше, чем возникнет ощущение, что по работе с источниками появились идеи, которыми можно поделиться.
• Должна быть идея, а для этого нужно научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного).
• Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно, а также стремясь структурировать свой текст.
• Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Методические рекомендации по выполнению контрольных работ

Контрольная работа выполняется по вариантам. На бланке указывается факультет, курс, группа, ФИО студента. Вопросы строятся на основе тестовых и ситуативных заданий. В тестовых заданиях, выбирается правильный(ые) ответ(ы). При решении ситуативных заданий выбирается правильная последовательность действий в рассматриваемой ситуации. Проверка контрольной работы позволяет выявить и исправить допущенные студентами ошибки, указать, какие вопросы дисциплины (модуля) ими недостаточно усвоены и требуют доработки. Студент должен внимательно ознакомиться с письменными замечаниями преподавателя и приступить к их исправлению, для чего еще раз повторить соответствующий материал.

Методические рекомендации по подготовке к коллоквиуму

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:
- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.
Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы. Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.
Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

Методические рекомендации по устному опросу/самоподготовке

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных сигналов,
воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств. В случае необходимости следует рекомендовать еще раз внимательно разобраться в материале. Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение решать задачи или пройти тестирование по пройденному материалу. Однако преподавателю следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Методические рекомендации по подготовке к семинарским занятиям

Одним из видов внеаудиторной самостоятельной работы является подготовка к семинарским занятиям. Семинар – форма учебно-практических занятий, при которой студенты обсуждают сообщения, доклады и рефераты, выполненные ими по результатам учебных или научных исследований под руководством преподавателя. Преподаватель в этом случае является координатором обсуждений темы семинара, подготовка к которому является обязательной. Поэтому тема семинара и основные источники обсуждения предъявляются до обсуждения для детального ознакомления, изучения. Цели обсуждений направлены на формирование навыков профессиональной полемики и закрепление обсуждаемого материала. Семинар – это такая форма организации обучения, при которой на этапе подготовки доминирует самостоятельная работа учащихся с учебной литературой и другими дидактическими средствами над серией вопросов, проблем и задач, а в процессе семинара идут активное обсуждение, дискуссии и выступления учащихся, где они под руководством преподавателя делают обобщающие выводы и заключения. Семинар предназначен для углубленного изучения дисциплины, овладения методологией научного познания, то главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли.

Методические рекомендации по подготовке к эссе

Одним из видов самостоятельной работы студентов является написание творческой работы по заданной либо согласованной с преподавателем теме. Творческая работа (эссе) представляет собой оригинальное произведение объемом 500-700 слов, посвященное какой-либо значимой классической либо современной проблеме в определенной теоретической и практической области. Творческая работа не является рефератом и не должна носить описательный характер, большое место в ней должно быть уделено аргументированному представлению своей точки зрения студентами, критической оценке рассматриваемого материала и проблематики, что должно способствовать раскрытию творческих и аналитических способностей. Цели написания эссе – научиться логически верно и аргументировано строить устную и письменную речь; работать над углублением и систематизацией своих философских знаний; овладеть способностью использовать основы знаний для формирования мировоззренческой позиции. Приступая к написанию эссе, изложите в одном предложении, что именно вы будете утверждать и доказывать (свой тезис).

Методические рекомендации по подготовке к докладу

Для подготовки доклада необходимо выбрать актуальную тему. Желательно, чтобы тема была интересна докладчику и вызывала желание качественно подготовить материалы. Подготовка доклада предполагает: определение цели доклада; подбор необходимого материала, определяющего содержание доклада; составление плана доклада, распределение собранного материала в необходимой логической последовательности.
Композиция доклада имеет вступление, основную часть и заключение.
Вступление должно содержать: название доклада; сообщение основной идеи; современную оценку предмета изложения; краткое перечисление рассматриваемых вопросов; интересную для слушателей форму изложения. Основная часть, в которой необходимо раскрыть суть темы, обычно строится по принципу отчёта. Задача основной части: представить достаточно данных для того, чтобы слушатели заинтересовались темой.
Заключение – чёткое обобщение и краткие выводы по излагаемой теме.

Методические рекомендации по подготовке к собеседованию

Собеседование – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.
Цель собеседования: проверка усвоения знаний; умений применять знания; сформированности профессионально значимых личностных качеств.
Подготовка к собеседованию предполагает повторение пройденного материала и приобретение навыка свободного владения терминологией и фактическими данными по определенному разделу дисциплины (модуля).

Методические рекомендации по подготовке к тестированию/ итоговому тестированию

Тестирование/итоговое тестирование – это система стандартизированных заданий, определенного содержания и упорядоченных в рамках определенной стратегии предъявления, позволяющая качественно оценить структуру и эффективно измерить уровень знаний, умений и навыков по дисциплине (модулю). Тестирование позволяет оценить у студентов уровень знаний фактического материала, умения логически мыслить, способности к рефлексии и творческому подходу к решению поставленной задачи на каждом этапе их обучения.

При самостоятельной подготовке к тестированию студенту необходимо:
- четко выяснить все условия тестирования заранее (сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов);
- проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- проработать информационный материал по разделу дисциплине;
- конспективно выписать из соответствующих разделов учебной дисциплины основные понятия, ключевые идеи, теоретические концепции и подходы;
- повторить теоретический материал лекций, самостоятельно изученной учебной и научной литературы, семинарских занятий.
При выполнении тестовых заданий рекомендуется:
- внимательно полностью прочитать вопрос и предлагаемые варианты ответов, выбрать правильные из них (их может быть несколько);
- при затруднении при ответе на сложный для студента вопрос целесообразно переходить к другим вопросам теста. По завершению ответов на все последующие вопросы теста необходимо вернуться к трудному вопросу и дать на него продуманный ответ;
- по окончании тестирования следует проверить правильность данных ответов на вопросы теста, чтобы избежать механических ошибок.

Методические рекомендации по подготовке к экзамену

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной дисциплине (модулю). Экзаменационная сессия – это серия экзаменов, установленных учебным планом. Между экзаменами интервал 2-4 дня, в течение студент систематизирует уже имеющиеся знания. На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки
вопросы. Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для
систематизации знаний.

Методические рекомендации по подготовке к зачету/зачету с оценкой

В ходе подготовки к зачету/зачету с оценкой студент, в первую очередь, должен систематизировать знания, полученные в ходе изучения дисциплины. К зачету/зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине (модулю). В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:
- программой дисциплины (модуля);
- перечнем знаний, умений и навыков, которыми студент должен владеть;
- тематическими планами лекций, семинарских занятий;
- учебниками, учебными пособиями по дисциплине, а также электронными ресурсами;
- перечнем вопросов к зачету/зачету с оценкой.
После этого у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине (модулю). Систематическое выполнение учебной работы на лекциях и лабораторных занятиях позволит успешно освоить дисциплину (модуль) и создать хорошую базу для сдачи зачета/зачета с оценкой.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература	
7.1.1. Основная литература	
Л.1.1	Янцев В. В. Web-программирование на Python [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 180 с. – Режим доступа: https://e.lanbook.com/book/310289
Л.1.2	Твердохлебова М.Д. Интернет-маркетинг [Электронный ресурс]:Учебник. - Москва: КноРус, 2023. - 190 с. – Режим доступа: https://book.ru/book/947409
7.1.2. Дополнительная литература	
Л.2.1	Диков А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 [Электронный ресурс]:учебное пособие. - Санкт-Петербург: Лань, 2019. - 188 с. – Режим доступа: https://e.lanbook.com/book/122174
Л.2.2	Терлецкий А. С. Создание практико-ориентированного web-пособия по использованию технологий формирования адаптивного web-дизайна [Электронный ресурс]:студенческая научная работа. - Б.м.: б.и., б.г.. - 96 с. – Режим доступа: https://biblioclub.ru/index.php?page=book&id=597326
Л.2.3	Беликова С. А., Беликов А. Н. Основы HTML и CSS: проектирование и дизайн веб-сайтов: учебное пособие по курсу «Web-разработка» [Электронный ресурс]:учебное пособие. - Ростов-на-Дону, Таганрог: Южный федеральный университет, 2020. - 176 с. – Режим доступа: https://biblioclub.ru/index.php?page=book&id=598663
7.2. Лицензионное и свободно распространяемое программное обеспечение в том числе отечественного производства	
7.2.1	Microsoft Windows 10

7.2.2	Kaspersky Endpoint Security
7.2.3	Microsoft Office 2013 Standard
7.2.4	Microsoft®WINHOME 10 Russian Academic OLP lLicense NoLevel Legalization GetGenuine
7.2.5	Notepad ++
7.2.6	Microsoft Visual Studio Professional 2017
7.2.7	Microsoft Visual Code
7.2.8	Python
7.3. Перечень профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет	
7.3.1	Электронно-библиотечная система "Лань". Режим доступа: https://e.lanbook.com/
7.3.2	Электронно-библиотечная система "Университетская библиотека онлайн". Режим доступа: https://biblioclub.ru/
7.3.3	Электронно-библиотечная система "BOOK.ru". Режим доступа: https://book.ru/
7.3.4	ПЛАТФОРМА ОНЛАЙН-ОБРАЗОВАНИЯ «РАЗУМ». Режим доступа: https://razoom.mgutm.ru/
7.3.5	Сайт национального открытого университета "ИНТУИТ". Режим доступа: https://intuit.ru/
7.3.6	Российский портал открытого образования. Режим доступа: https://openedu.ru/

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-28 - Компьютерный класс : 15 персональных компьютеров, объединённых в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийная установка «Toshiba» с экраном 3,5 м. Лицензионные программы:. ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01,. WinDjView, КОМПАС-3D LT, 1С Предприятие 8.1, Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7 ─ Эксперт Системс» (сетевая версия). CorelDRAW Graphics Suite X7. GRAPHISOFT ArchiCAD 15, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, QBacic, Pascal, Delphi, Kaspersky Endpoint Security. IQlib. Все компьютеры подключены к сети Internet

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Организация образовательного процесса для лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей. Предусмотрена возможность обучения по индивидуальному графику, при составлении которого возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2027 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2027 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2028 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2028 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2029 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2029 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2030 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2030 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____