

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ  
имени И.Т. ТРУБИЛИНА»

Факультет прикладной информатики  
Компьютерных технологий и систем

УТВЕРЖДЕНО

Декан

Замотайлова Д.А.

21.05.2026

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)  
«ПРОГРАММИРОВАНИЕ И РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) подготовки: Разработка и модификация информационных систем и баз данных

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем:  
в зачетных единицах: 4 з.е.  
в академических часах: 144 ак.ч.



**Разработчики:**

Доцент, кафедра экономической кибернетики Аврахам Е.М.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденного приказом Минобрнауки от 19.09.2017 № 926, с учетом трудовых функций профессиональных стандартов: "Администратор баз данных", утвержден приказом Минтруда России от 27.04.2023 № 408н.

**Согласование и утверждение**

| № | Подразделение или коллегиальный орган | Ответственное лицо                                               | ФИО             | Виза        | Дата, протокол (при наличии) |
|---|---------------------------------------|------------------------------------------------------------------|-----------------|-------------|------------------------------|
| 1 | Компьютерных технологий и систем      | Заведующий кафедрой, руководитель подразделения, реализующего ОП | Лукьяненко Т.В. | Согласовано | 06.04.2026, № 10             |

## 1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - является формирование у студентов базовых знаний, умений и навыков решения наиболее важных и часто встречаемых на практике задач по веб-программированию на языке программирования Python, а также создание систем и приложений с использованием CMS Django.

Задачи изучения дисциплины:

- - сформировать у обучающихся способность применять знания программирования для хранения и обработки данных;;
- - сформировать у обучающихся способность применять знания программирования для решения типовых задач;;
- - сформировать у обучающихся способность применять знания программирования для решения задач по созданию веб-приложений..

## 2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

*Компетенции, индикаторы и результаты обучения*

ПК-П4 Способность выполнять работы по созданию (модификации) и сопровождению информационных систем.

ПК-П4.1 Знает инструменты и методы проектирования архитектуры ис, инструменты и методы верификации архитектуры ис, возможности ис, предметную область автоматизации, архитектуру, устройство и функционирование вычислительных систем

*Знать:*

ПК-П4.1/Зн1 Коммуникационное оборудование сетевые протоколы основы современных операционных систем основы современных СУБД устройство и функционирование современных ис

ПК-П4.1/Зн2 Архитектура мультиарендного программного обеспечения основы ИБ организации современные стандарты информационного взаимодействия систем программные средства и платформы инфраструктуры информационных технологий организаций современные подходы и стандарты автоматизации организации

*Уметь:*

ПК-П4.1/Ум1 Осуществлять коммуникации с заинтересованными сторонами в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Владеть:*

ПК-П4.1/Нв1 Навыками создания вариантов архитектурных спецификаций ис в рамках выполнения работ

ПК-П4.2 Умеет проектировать архитектуру ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, проверять (верифицировать) архитектуру ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, анализировать исходную документацию в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, разрабатывать документы в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, осуществлять коммуникации с заинтересованными сторонами в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Знать:*

ПК-П4.2/Зн1 Системы классификации и кодирования информации, в том числе присвоения кодов документам и элементам справочников

ПК-П4.2/Зн2 Отраслевая нормативно-техническая документация

*Уметь:*

ПК-П4.2/Ум1 Разрабатывать документы в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

ПК-П4.2/Ум2 Анализировать исходную документацию в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Владеть:*

ПК-П4.2/Нв1 Навыками создания вариантов архитектурных спецификаций ис в рамках управления работами по созданию (модификации) и сопровождению ис

ПК-П4.3 Владеет навыками создания вариантов архитектурных спецификаций ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, выбора и согласования с заинтересованными сторонами оптимальной архитектурной спецификации ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Знать:*

ПК-П4.3/Зн1 Источники информации, необходимой для профессиональной деятельности при выполнении работ и управлении работами по созданию (модификации) и сопровождению ис

*Уметь:*

ПК-П4.3/Ум1 Проверять (верифицировать) архитектуру ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

ПК-П4.3/Ум2 Проектировать архитектуру ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Владеть:*

ПК-П4.3/Нв1 Навыками выбора и согласование с заинтересованными сторонами оптимальной архитектурной спецификации ис

ПК-П13 Способность выполнять логическую и функциональную работу по созданию комплекса программ.

ПК-П13.1 Знает инструменты и методы проектирования структур баз данных, инструменты и методы верификации структуры базы данных, возможности ис, предметную область автоматизации, основы современных субд, теория баз данных, основы программирования, современные объектно-ориентированные языки программирования, современные структурные языки программирования, языки современных бизнес-приложений

*Знать:*

ПК-П13.1/Зн1 Источники информации, необходимой для профессиональной деятельности при выполнении работ и управлении работами по созданию (модификации) и сопровождению ис

*Уметь:*

ПК-П13.1/Ум1 Работать с субд

*Владеть:*

ПК-П13.1/Нв1 Навыками разработки структуры баз данных ис в соответствии с архитектурной спецификацией

ПК-П13.2 Умеет работать с субд в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, анализировать и структурировать входные данные в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Знать:*

ПК-П13.2/Зн1 Теория баз данных

*Уметь:*

ПК-П13.2/Ум1 Анализировать и структурировать входные данные

*Владеть:*

ПК-П13.2/Нв1 Навыками верификации структуры баз данных ис относительно архитектуры ис и требований заказчика

ПК-П13.3 Владеет навыками разработки структуры баз данных ис в соответствии с архитектурной спецификацией в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, верификация структуры баз данных ис относительно архитектуры ис и требований заказчика к ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис, устранения обнаруженных несоответствий в структуре баз данных ис в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ис

*Знать:*

ПК-П13.3/Зн1 Основы современных субд

*Уметь:*

ПК-П13.3/Ум1 Работать с субд в рамках управления работами по созданию (модификации) и сопровождению ис

*Владеть:*

ПК-П13.3/Нв1 Навыками устранения обнаруженных несоответствий в структуре баз данных ис в рамках выполнения работ

### 3. Место дисциплины в структуре ОП

Дисциплина (модуль) «Программирование и разработка Web-приложений» относится к формируемой участниками образовательных отношений части образовательной программы и изучается в семестре(ах): 6.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

### 4. Объем дисциплины (модуля) и виды учебной работы

| Период обучения | Общая трудоемкость (часы) | Общая трудоемкость (ЗЕТ) | Контактная работа (часы, всего) | Внеаудиторная контактная работа (часы) | Лабораторные занятия (часы) | Лекционные занятия (часы) | Практические занятия (часы) | Самостоятельная работа (часы) | Промежуточная аттестация (часы) |
|-----------------|---------------------------|--------------------------|---------------------------------|----------------------------------------|-----------------------------|---------------------------|-----------------------------|-------------------------------|---------------------------------|
|                 |                           |                          |                                 |                                        |                             |                           |                             |                               |                                 |

|                |     |   |    |   |    |    |    |    |                                    |
|----------------|-----|---|----|---|----|----|----|----|------------------------------------|
| Шестой семестр | 144 | 4 | 65 | 5 | 22 | 16 | 22 | 52 | Курсовая работа<br>Экзамен<br>(27) |
| Всего          | 144 | 4 | 65 | 5 | 22 | 16 | 22 | 52 | 27                                 |

## 5. Содержание дисциплины (модуля)

### 5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

| Наименование раздела, темы                                                                  | Всего      | Внеаудиторная контактная работа | Лабораторные занятия | Лекционные занятия | Практические занятия | Самостоятельная работа | Планируемые результаты обучения, соотношенные с результатами освоения программы |
|---------------------------------------------------------------------------------------------|------------|---------------------------------|----------------------|--------------------|----------------------|------------------------|---------------------------------------------------------------------------------|
| <b>Раздел 1. Функциональное и объектно ориентированное программирование на языке PYTHON</b> | <b>74</b>  |                                 | <b>10</b>            | <b>8</b>           | <b>16</b>            | <b>40</b>              | ПК-П4.1<br>ПК-П4.2<br>ПК-П4.3<br>ПК-П13.1                                       |
| Тема 1.1. Структуры данных в Python.                                                        | 18         |                                 | 2                    | 2                  | 4                    | 10                     | ПК-П13.2<br>ПК-П13.3                                                            |
| Тема 1.2. Функциональное программирование.                                                  | 18         |                                 | 2                    | 2                  | 4                    | 10                     |                                                                                 |
| Тема 1.3. Объектно-ориентированное программирование в Python.                               | 19         |                                 | 3                    | 2                  | 4                    | 10                     |                                                                                 |
| Тема 1.4. Использование Python для работы с базой данных.                                   | 19         |                                 | 3                    | 2                  | 4                    | 10                     |                                                                                 |
| <b>Раздел 2. Разработка Web-приложений</b>                                                  | <b>38</b>  |                                 | <b>12</b>            | <b>8</b>           | <b>6</b>             | <b>12</b>              | ПК-П4.1<br>ПК-П4.2<br>ПК-П4.3<br>ПК-П13.1                                       |
| Тема 2.1. Основы взаимодействия с Интернет.                                                 | 12         |                                 | 4                    | 2                  | 2                    | 4                      | ПК-П13.2<br>ПК-П13.3                                                            |
| Тема 2.2. Фреймворк Django и Python.                                                        | 14         |                                 | 4                    | 4                  | 2                    | 4                      |                                                                                 |
| Тема 2.3. Разработка приложений с использованием Django.                                    | 12         |                                 | 4                    | 2                  | 2                    | 4                      |                                                                                 |
| <b>Раздел 3. Промежуточная аттестация (экзамен, курсовая работа)</b>                        | <b>5</b>   | <b>5</b>                        |                      |                    |                      |                        | ПК-П4.1<br>ПК-П4.2<br>ПК-П4.3                                                   |
| Тема 3.1. Курсовая работа                                                                   | 2          | 2                               |                      |                    |                      |                        | ПК-П13.1<br>ПК-П13.2                                                            |
| Тема 3.2. Экзамен                                                                           | 3          | 3                               |                      |                    |                      |                        | ПК-П13.3                                                                        |
| <b>Итого</b>                                                                                | <b>117</b> | <b>5</b>                        | <b>22</b>            | <b>16</b>          | <b>22</b>            | <b>52</b>              |                                                                                 |

### 5.2. Содержание разделов, тем дисциплин

## **Раздел 1. Функциональное и объектно ориентированное программирование на языке PYTHON**

**(Лабораторные занятия - 10ч.; Лекционные занятия - 8ч.; Практические занятия - 16ч.; Самостоятельная работа - 40ч.)**

### *Тема 1.1. Структуры данных в Python.*

*(Лабораторные занятия - 2ч.; Лекционные занятия - 2ч.; Практические занятия - 4ч.; Самостоятельная работа - 10ч.)*

1. Программное обеспечение и запуск.
2. Работа в консоли GUI
3. Простые структуры данных и операции над ними
4. Строки, кортежи, списки, словари и операции над ними.
5. Управляющие конструкции.

### *Тема 1.2. Функциональное программирование.*

*(Лабораторные занятия - 2ч.; Лекционные занятия - 2ч.; Практические занятия - 4ч.; Самостоятельная работа - 10ч.)*

1. Функции и лямбда функции в Python.
2. Шифрование строк.
3. Кодировка строк.
4. Модули os и sys.

### *Тема 1.3. Объектно-ориентированное программирование в Python.*

*(Лабораторные занятия - 3ч.; Лекционные занятия - 2ч.; Практические занятия - 4ч.; Самостоятельная работа - 10ч.)*

1. Описание и использование классов.
2. Объекты и наследование.
3. Обработка исключений.
4. Декораторы и генераторы в объектно-ориентированном программировании в Python/

### *Тема 1.4. Использование Python для работы с базой данных.*

*(Лабораторные занятия - 3ч.; Лекционные занятия - 2ч.; Практические занятия - 4ч.; Самостоятельная работа - 10ч.)*

1. Работа с реляционными базами данных (SQLite, MySQL, PostgreSQL)
2. Работа с не реляционными базами данных (MongoDB).

## **Раздел 2. Разработка Web-приложений**

**(Лабораторные занятия - 12ч.; Лекционные занятия - 8ч.; Практические занятия - 6ч.; Самостоятельная работа - 12ч.)**

### *Тема 2.1. Основы взаимодействия с Интернет.*

*(Лабораторные занятия - 4ч.; Лекционные занятия - 2ч.; Практические занятия - 2ч.; Самостоятельная работа - 4ч.)*

1. Использование регулярных выражений.
2. Скрапинг с использованием BeautifulSoup.
3. Многопоточные и многопроцессорные возможности python

### *Тема 2.2. Фреймворк Django и Python.*

*(Лабораторные занятия - 4ч.; Лекционные занятия - 4ч.; Практические занятия - 2ч.; Самостоятельная работа - 4ч.)*

1. Знакомство с Django.
2. Библиотеки Django.
3. Использование библиотек Django для создания блога.

*Тема 2.3. Разработка приложений с использованием Django.*

*(Лабораторные занятия - 4ч.; Лекционные занятия - 2ч.; Практические занятия - 2ч.; Самостоятельная работа - 4ч.)*

1. Администраторская часть Django при разработке приложений.
2. Данные, видимые пользователям.
3. Авторизация и использование форм.

**Раздел 3. Промежуточная аттестация (экзамен, курсовая работа)**  
***(Внеаудиторная контактная работа - 5ч.)***

*Тема 3.1. Курсовая работа*

*(Внеаудиторная контактная работа - 2ч.)*

Темы курсовых работ.

1. Разработка Web-приложения благотворительного фонда
2. Разработка Web-приложения для компьютерного магазина
3. Разработка учебного Web-приложения
4. Разработка Web-приложения «Зоопарк»
5. Разработка Web-приложения любителей собак
6. Разработка Web-приложения футбольного клуба
7. Разработка Web-приложения для клуба любителей кошек
8. Разработка Web-приложения для кулинаров
9. Разработка Web-приложения для садовода
10. Разработка Web-приложения для центра дистанционного обучения
11. Разработка Web-приложения строительной фирмы
12. Разработка Web-приложения общественно-политической организации
13. Разработка Web-приложения электронной библиотеки
14. Разработка Web-приложения деятелей науки
15. Разработка Web-приложения картинной галереи
16. Разработка Web-приложения для кондитерской фабрики
17. Разработка Web-приложения магазина бытовой техники
18. Разработка личного Web-приложения
19. Разработка Web-приложения для салона красоты
20. Разработка Web-приложения для салона автомобилей
21. Разработка Web-приложения «Герои нашего времени»
22. Разработка Web-приложения «Жизнь замечательных людей»
23. Разработка Web-приложения для строительной организации
24. Разработка Web-приложения для туристической фирмы
25. Разработка Web-приложения для мебельного магазина

*Тема 3.2. Экзамен*

*(Внеаудиторная контактная работа - 3ч.)*

## Вопросы к экзамену

1. Введение в серверное Web-программирование
2. Язык программирования Python
3. Библиотека Django
4. Создание проекта и приложения Django
5. Модели Django
6. Контроллеры Django. Регулярные выражения
7. Простые шаблоны Django
8. Постраничный вывод данных. Пагинатор Django
9. Вывод на основе классов. Классы-контроллеры Django
10. Простые формы Django
11. Выгрузка файлов на Web-сайт
12. Разграничение доступа
13. Комментарии Django
14. Статичные страницы Django
15. Создание и вывод миниатюр. Библиотека easy-thumbnails
16. Привязка тегов к данным. Библиотека django-taggit
17. Создание web-сайта. Планирование и предварительные действия
18. Главная страница
19. Гостевая книга
20. Список новостей. Хранилище изображений
21. Список категорий товаров
22. Список товаров
23. Блог
24. Генерирование каналов новостей RSS и Atom
25. Рассылка электронной почты
26. Журналирование
27. Настройка встроенного административного сайта Django
28. Публикация Web-сайта
29. Миграции в Python
30. Особенности системы администрирования фрейворка Django

## 6. Оценочные материалы текущего контроля

### **Раздел 1. Функциональное и объектно ориентированное программирование на языке PYTHON**

*Форма контроля/оценочное средство: Задача*

*Вопросы/Задания:*

1. Объясните, как работает протокол HTTP. Опишите процесс отправки и получения HTTP-запроса от клиента к серверу, включая ключевые этапы и компоненты (URL, метод, заголовки, тело).

Объясните, как работает протокол HTTP. Опишите процесс отправки и получения HTTP-запроса от клиента к серверу, включая ключевые этапы и компоненты (URL, метод, заголовки, тело).

2. Сравните методы HTTP: GET, POST, PUT, DELETE. В каких случаях каждый из них используется? Приведите примеры из реальных веб-приложений.

Сравните методы HTTP: GET, POST, PUT, DELETE. В каких случаях каждый из них используется? Приведите примеры из реальных веб-приложений.

3. Что такое REST и какие принципы он включает? Почему REST стал стандартом для API? Приведите пример RESTful API с описанием ресурсов и маршрутов.

Что такое REST и какие принципы он включает? Почему REST стал стандартом для API? Приведите пример RESTful API с описанием ресурсов и маршрутов.

4. В чем разница между REST и RPC? Какие преимущества и недостатки у каждого подхода при разработке веб-сервисов?

В чем разница между REST и RPC? Какие преимущества и недостатки у каждого подхода при разработке веб-сервисов?

5. Опишите, как работает WebSocket. В каких случаях он предпочтительнее HTTP? Приведите примеры приложений, где используется WebSocket.

Опишите, как работает WebSocket. В каких случаях он предпочтительнее HTTP? Приведите примеры приложений, где используется WebSocket.

6. Сравните форматы данных JSON, XML и Protobuf. В каких сценариях лучше использовать каждый из них? Обоснуйте с точки зрения производительности, читаемости и совместимости.

Сравните форматы данных JSON, XML и Protobuf. В каких сценариях лучше использовать каждый из них? Обоснуйте с точки зрения производительности, читаемости и совместимости.

7. Что такое сериализация и десериализация данных? Как они реализуются в Python при работе с JSON? Приведите примеры использования `json.dumps()` и `json.loads()`.

Что такое сериализация и десериализация данных? Как они реализуются в Python при работе с JSON? Приведите примеры использования `json.dumps()` и `json.loads()`.

8. Объясните, как работает кэширование в HTTP. Какие заголовки отвечают за кэширование (например, Cache-Control, ETag)? Как они влияют на производительность?

Объясните, как работает кэширование в HTTP. Какие заголовки отвечают за кэширование (например, Cache-Control, ETag)? Как они влияют на производительность?

9. Что такое CORS и почему он важен для безопасности веб-приложений? Как настроить CORS на сервере (например, в Django или FastAPI)?

Что такое CORS и почему он важен для безопасности веб-приложений? Как настроить CORS на сервере (например, в Django или FastAPI)?

10. Опишите, как работает процесс авторизации и аутентификации на уровне HTTP (например, через заголовки Authorization, токены JWT). Какие уязвимости могут возникнуть при неправильной реализации?

Опишите, как работает процесс авторизации и аутентификации на уровне HTTP (например, через заголовки Authorization, токены JWT). Какие уязвимости могут возникнуть при неправильной реализации?

## **Раздел 2. Разработка Web-приложений**

*Форма контроля/оценочное средство: Задача*

*Вопросы/Задания:*

1. В чем заключаются основные принципы архитектуры Django? Опишите MVC/MVT-паттерн и как он реализован в Django.

В чем заключаются основные принципы архитектуры Django? Опишите MVC/MVT-паттерн и как он реализован в Django.

2. Что такое ORM в Django? Как она помогает в разработке? Приведите примеры сложных запросов с использованием QuerySet, включая фильтрацию, агрегацию и аннотации. Что такое ORM в Django? Как она помогает в разработке? Приведите примеры сложных запросов с использованием QuerySet, включая фильтрацию, агрегацию и аннотации.

3. Объясните, как работает система шаблонов в Django. В чем разница между шаблонами и статическими файлами? Как обеспечить безопасность при рендеринге шаблонов?

Объясните, как работает система шаблонов в Django. В чем разница между шаблонами и статическими файлами? Как обеспечить безопасность при рендеринге шаблонов?

4. Что такое Django REST Framework (DRF)? Какие компоненты он предоставляет (сериализаторы, вьюсеты, маршрутизация)? Приведите пример создания API для модели.

Что такое Django REST Framework (DRF)? Какие компоненты он предоставляет (сериализаторы, вьюсеты, маршрутизация)? Приведите пример создания API для модели.

5. Сравните Django REST Framework и Django Ninja. В каких случаях лучше использовать Ninja? Какие преимущества он дает в производительности и типизации?

Сравните Django REST Framework и Django Ninja. В каких случаях лучше использовать Ninja?

Какие преимущества он дает в производительности и типизации?

6. Опишите архитектуру FastAPI. Почему он считается одним из самых быстрых фреймворков для Python? Какие особенности делают его подходящим для микросервисов? Опишите архитектуру FastAPI. Почему он считается одним из самых быстрых фреймворков для Python? Какие особенности делают его подходящим для микросервисов?

7. Что такое Pydantic и как он используется в FastAPI? Объясните концепцию валидации данных на основе моделей Pydantic. Приведите пример.

Что такое Pydantic и как он используется в FastAPI? Объясните концепцию валидации данных на основе моделей Pydantic. Приведите пример.

8. Как работает асинхронность в FastAPI? Напишите пример асинхронного эндпоинта, который делает HTTP-запрос к внешнему API с помощью httpx.

Как работает асинхронность в FastAPI? Напишите пример асинхронного эндпоинта, который делает HTTP-запрос к внешнему API с помощью httpx.

9. В чем заключается преимущество автоматической генерации OpenAPI и Swagger в FastAPI? Как пользователь может использовать эти документы?

В чем заключается преимущество автоматической генерации OpenAPI и Swagger в FastAPI? Как пользователь может использовать эти документы?

10. Опишите архитектуру Flask. Почему он считается микрофреймворком? Какие расширения (например, Flask-SQLAlchemy, Flask-RESTful) расширяют его функциональность?

Опишите архитектуру Flask. Почему он считается микрофреймворком? Какие расширения (например, Flask-SQLAlchemy, Flask-RESTful) расширяют его функциональность?

11. Сравните Flask и FastAPI. В каких проектах предпочтительнее использовать каждый из них? Приведите аргументы с точки зрения производительности, типизации и асинхронности.

Сравните Flask и FastAPI. В каких проектах предпочтительнее использовать каждый из них? Приведите аргументы с точки зрения производительности, типизации и асинхронности.

12. Что такое Starlette? Как он связан с FastAPI? В каких случаях можно использовать Starlette напрямую?

Что такое Starlette? Как он связан с FastAPI? В каких случаях можно использовать Starlette напрямую?

13. Расскажите о BlackSheep как альтернативе FastAPI. Чем он отличается? Какие преимущества и ограничения он имеет?

Расскажите о BlackSheep как альтернативе FastAPI. Чем он отличается? Какие преимущества и ограничения он имеет?

14. Как реализовать middleware в Django, FastAPI и Flask? Приведите примеры использования (логирование, аутентификация, обработка ошибок).

Как реализовать middleware в Django, FastAPI и Flask? Приведите примеры использования (логирование, аутентификация, обработка ошибок).

15. Опишите процесс обработки запроса в каждом из трех основных фреймворков (Django, FastAPI, Flask). Как формируется ответ?

Опишите процесс обработки запроса в каждом из трех основных фреймворков (Django, FastAPI, Flask). Как формируется ответ?

### **Раздел 3. Промежуточная аттестация (экзамен, курсовая работа)**

*Форма контроля/оценочное средство:*

*Вопросы/Задания:*

.

## **7. Оценочные материалы промежуточной аттестации**

Вопросы/Задания:

1. Блог-платформа на Django с REST API (DRF) и пользовательской аутентификацией
  - Django ORM, шаблонизация, DRF, JWT, PostgreSQL
2. Сервис новостей с парсингом RSS и асинхронной загрузкой данных (FastAPI + BeautifulSoup + aiohttp)
  - Асинхронность, FastAPI, Pydantic, кеширование в Redis
3. Микросервис управления задачами (To-Do) с использованием Flask и MongoDB
  - Flask-RESTful, NoSQL, CRUD, REST API
4. Система чатов в реальном времени с WebSockets (Django Channels)
  - WebSocket, Django, аутентификация, группы чатов
5. API для интернет-магазина на FastAPI с поддержкой OpenAPI и Swagger
  - FastAPI, Pydantic, валидация, документация, PostgreSQL
6. Сервис коротких ссылок (аналог Bitly) на Flask с кешированием в Redis
  - Flask, Redis, редиректы, аналитика посещений
7. Портфолио-сайт с генерацией контента через ИИ (ChatGPT API)
  - Flask/Django, AI-интеграция, HTML/CSS, JSON API
8. Система мониторинга веб-сайтов с уведомлениями (Telegram-бот)
  - Асинхронные проверки, FastAPI, Telegram Bot API, уведомления
9. CRM-система на Django с пользовательскими ролями и шаблонами
  - Django ORM, пользовательские модели, админка, шаблоны
10. Фотогалерея с загрузкой изображений и тегами (Django + PostgreSQL)
  - Файловая система, формы, тегирование, ORM
11. Поисковый микросервис с индексацией статей (Elasticsearch + FastAPI)
  - Поиск, индексация, асинхронная обработка, REST
12. Сервис опросов (анкет) с GraphQL API (Strawberry + Django)
  - GraphQL, Strawberry, Django, пользовательские ответы
13. Календарь событий с синхронизацией и напоминаниями (Flask + Celery)
  - Планировщик задач, Flask, уведомления, PostgreSQL
14. Система комментариев для сайта с модерацией и AI-фильтрацией (Llama 3)
  - FastAPI, ИИ-анализ текста, модерация, REST
15. Веб-приложение для просмотра погоды с картой (React + FastAPI)
  - Интеграция с внешним API, JSON, CORS, фронтенд-бэкенд

16. Сервис аналитики посещений сайта (аналог Google Analytics)
  - Flask, хранение данных в PostgreSQL, визуализация
17. Чат-бот для Telegram с генерацией текста (ChatGPT) и командным интерфейсом
  - Python, Telegram API, AI, асинхронность
18. Онлайн-консультант с поддержкой WebSockets и AI-подсказками
  - WebSocket, FastAPI, ИИ, интерфейс поддержки
19. Контейнеризованный блог на Django с Docker и разворачиванием на AWS
  - Docker, Django, PostgreSQL, AWS EC2, Nginx
20. Микросервисная архитектура: пользователи (FastAPI) + заказы (Flask) + шина сообщений (RabbitMQ)
  - Микросервисы, REST, асинхронная коммуникация, Docker
21. Serverless-функция на AWS Lambda для обработки форм (Python + API Gateway)
  - Serverless, AWS Lambda, Python, JSON, валидация
22. Сервис генерации PDF-отчётов из данных (Django + WeasyPrint)
  - Генерация PDF, Django, шаблоны, экспорт данных
23. Веб-приложение с WebAssembly (Wasm) для обработки изображений на фронтенде
  - Rust + Wasm, Python-бэкенд (FastAPI), обмен данными
24. Система голосования с защитой от дублирования (Redis + Flask)
  - Redis (сессии), IP-ограничения, простой интерфейс
25. API с поддержкой нескольких форматов: JSON, XML, Protobuf (FastAPI)
  - FastAPI, сериализация, content negotiation, Pydantic
26. Сервис распознавания текста на изображениях (OCR) через ИИ (Tesseract + FastAPI)
  - Обработка файлов, OCR, REST, асинхронность
27. Платформа для онлайн-курсов с прогрессом обучения (Django + DRF)
  - Пользовательские роли, прогресс, медиа, REST API
28. Реализация GraphQL API для управления библиотекой (Strawberry + TortoiseORM)
  - GraphQL, асинхронный ORM, фильтрация, сортировка
29. Деплой FastAPI-приложения в Kubernetes (minikube + Docker)
  - Kubernetes, Helm, Docker, масштабируемость, CI/CD (опционально)
30. Гибридное веб-приложение: Django (админка) + FastAPI (API) + React (фронт)

*Шестой семестр, Экзамен*

*Контролируемые ИДК: ПК-П4.1 ПК-П13.1 ПК-П4.2 ПК-П13.2 ПК-П4.3 ПК-П13.3*

Вопросы/Задания:

1. Сравните реляционные (PostgreSQL, MySQL) и нереляционные (MongoDB) базы данных. В каких случаях лучше использовать каждую из них?
2. Напишите сложный SQL-запрос с JOIN, подзапросом, группировкой и агрегацией (например, подсчет количества заказов по пользователям за последний месяц).
3. Что такое транзакции в базах данных? Как они реализуются в PostgreSQL? Приведите пример использования транзакций в Django ORM.
4. Объясните, как работает индексация в базах данных. Какие типы индексов существуют? Как индексы влияют на производительность SELECT и UPDATE?
5. Что такое NoSQL и в каких сценариях он предпочтительнее SQL? Приведите пример использования MongoDB для хранения гибких документов.
6. Опишите, как Redis используется для кэширования. Приведите пример кэширования результата API-запроса с TTL.
7. Сравните ORM: Django ORM, SQLAlchemy и TortoiseORM. В каких случаях использовать каждый из них? Как TortoiseORM поддерживает асинхронность?
8. Что такое миграции базы данных? Как они работают в Django и Alembic (для SQLAlchemy)? Зачем они нужны?
9. Как реализовать пагинацию в API с использованием ORM? Приведите пример на FastAPI с SQLAlchemy.
10. Объясните, что такое N+1 проблема в ORM и как её избежать. Приведите пример с Django ORM и select\_related/prefetch\_related.
11. Что такое контейнеризация? Объясните, как работает Docker. Напишите пример Dockerfile для приложения на FastAPI.
12. Как развернуть веб-приложение на Django с помощью Docker? Опишите структуру docker-compose.yml с веб-сервером и PostgreSQL.
13. Что такое Kubernetes и зачем он нужен? Опишите основные компоненты (Pod, Deployment, Service) и их назначение.
14. Сравните развертывание приложения на виртуальной машине и в Kubernetes. Какие преимущества дает оркестрация?
15. Опишите, как работает AWS Lambda. В каких случаях использовать serverless-подход? Приведите пример API на FastAPI, задеплоенного через AWS Lambda.
16. Что такое Vercel и как он используется для деплоя веб-приложений? Можно ли разворачивать Python-приложения на Vercel? Какие ограничения есть?
17. Как настроить CI/CD для веб-приложения (например, через GitHub Actions)? Опишите этапы: тестирование, сборка Docker-образа, деплой.

18. Объясните, как работает обратный прокси (например, Nginx) в связке с Gunicorn и Django. Зачем он нужен?

19. Что такое облачная архитектура? Какие сервисы AWS/GCP/Azure могут использоваться для развертывания веб-приложения (например, EC2, S3, RDS)?

20. Как обеспечить безопасность при деплое веб-приложения (SSL, секреты, брандмауэры)? Как управлять переменными окружения в production?

21. Как интегрировать ИИ (например, ChatGPT или Llama в веб-приложение? Приведите пример чат-бота на FastAPI, использующего LLM через API.

22. Что такое GraphQL и чем он отличается от REST? Какие проблемы REST решает GraphQL? Приведите пример схемы и запроса.

23. Как использовать Strawberry для создания GraphQL API на Python? Сравните с REST-подходом.

24. Что такое WebAssembly (Wasm) и как он может использоваться в веб-приложениях? Приведите пример ускорения вычислений на фронтенде с помощью Wasm.

25. Как современные веб-приложения используют AI для генерации контента (например, текстов, изображений)? Опишите архитектуру такого сервиса (фронтенд, бэкенд, модель, API).

## **8. Материально-техническое и учебно-методическое обеспечение дисциплины**

### **8.1. Перечень основной и дополнительной учебной литературы**

#### *Основная литература*

1. Столбовский, Д. Н. Разработка Web-приложений ASP.NET с использованием Visual Studio .NET: учебное пособие / Д. Н. Столбовский, . - Разработка Web-приложений ASP.NET с использованием Visual Studio .NET - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. - 375 с. - 978-5-4497-0370-5. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/89469.html> (дата обращения: 08.10.2025). - Режим доступа: по подписке

#### *Дополнительная литература*

1. КРЕЙМЕР А. С. Web-программирование: метод. рекомендации / КРЕЙМЕР А. С., Крепышев Д. А.. - Краснодар: КубГАУ, 2020. - 36 с. - Текст: электронный. // : [сайт]. - URL: <https://edu.kubsau.ru/mod/resource/view.php?id=9244> (дата обращения: 24.07.2026). - Режим доступа: по подписке

### **8.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся**

#### *Профессиональные базы данных*

1. <http://www.iprbookshop.ru/> - IPRbook

*Ресурсы «Интернет»*

1. <https://edu.kubsau.ru/> - Образовательный портал КубГАУ
2. <https://elibrary.ru> - Научная электронная библиотека «eLIBRARY.RU»

### **8.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине**

*Перечень программного обеспечения*

*(обновление производится по мере появления новых версий программы)*

Не используется.

*Перечень информационно-справочных систем*

*(обновление выполняется еженедельно)*

Не используется.

### **8.4. Специальные помещения, лаборатории и лабораторное оборудование**

## **9. Методические указания по освоению дисциплины (модуля)**

Учебная работа по направлению подготовки осуществляется в форме контактной работы с преподавателем, самостоятельной работы обучающегося, текущей и промежуточной аттестаций, иных формах, предлагаемых университетом. Учебный материал дисциплины структурирован и его изучение производится в тематической последовательности. Содержание методических указаний должно соответствовать требованиям Федерального государственного образовательного стандарта и учебных программ по дисциплине. Самостоятельная работа студентов может быть выполнена с помощью материалов, размещенных на портале поддержки Moodle.

### ***Методические указания по формам работы***

#### *Лекционные занятия*

Передача значительного объема систематизированной информации в устной форме достаточно большой аудитории. Дает возможность экономно и систематично излагать учебный материал. Обучающиеся изучают лекционный материал, размещенный на портале поддержки обучения Moodle.

#### *Лабораторные занятия*

Практическое освоение студентами научно-теоретических положений изучаемого предмета, овладение ими техникой экспериментирования в соответствующей отрасли науки. Лабораторные занятия проводятся с использованием методических указаний, размещенных на образовательном портале университета.

#### *Практические занятия*

Форма организации обучения, проводимая под руководством преподавателя и служащая для детализации, анализа, расширения, углубления, закрепления, применения (или выполнения) разнообразных практических работ, упражнений) и контроля усвоения полученной на лекциях учебной информации. Практические занятия проводятся с использованием учебно-методических изданий, размещенных на образовательном портале университета.

### ***Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами***

Для инвалидов и лиц с ОВЗ может изменяться объём дисциплины (модуля) в часах, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося (при этом не увеличивается количество зачётных единиц, выделенных на освоение дисциплины).

Фонды оценочных средств адаптируются к ограничениям здоровья и восприятия информации обучающимися.

Основные формы представления оценочных средств – в печатной форме или в форме электронного документа.

Формы контроля и оценки результатов обучения инвалидов и лиц с ОВЗ с нарушением зрения:

- устная проверка: дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.;

- с использованием компьютера и специального ПО: работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, дистанционные формы, если позволяет острота зрения - графические работы и др.;

- при возможности письменная проверка с использованием рельефно-точечной системы Брайля, увеличенного шрифта, использование специальных технических средств (тифлотехнических средств): контрольные, графические работы, тестирование, домашние задания, эссе, отчеты и др.

Формы контроля и оценки результатов обучения инвалидов и лиц с ОВЗ с нарушением слуха:

- письменная проверка: контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.;

- с использованием компьютера: работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, графические работы, дистанционные формы и др.;

- при возможности устная проверка с использованием специальных технических средств (аудиосредств, средств коммуникации, звукоусиливающей аппаратуры и др.): дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.

Формы контроля и оценки результатов обучения инвалидов и лиц с ОВЗ с нарушением опорно-двигательного аппарата:

- письменная проверка с использованием специальных технических средств (альтернативных средств ввода, управления компьютером и др.): контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.;

- устная проверка, с использованием специальных технических средств (средств коммуникаций): дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.;

- с использованием компьютера и специального ПО (альтернативных средств ввода и управления компьютером и др.): работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, графические работы, дистанционные формы предпочтительнее обучающимся, ограниченным в передвижении и др.

Адаптация процедуры проведения промежуточной аттестации для инвалидов и лиц с ОВЗ.

В ходе проведения промежуточной аттестации предусмотрено:

- предъявление обучающимся печатных и (или) электронных материалов в формах, адаптированных к ограничениям их здоровья;

- возможность пользоваться индивидуальными устройствами и средствами, позволяющими адаптировать материалы, осуществлять приём и передачу информации с учетом их индивидуальных особенностей;

- увеличение продолжительности проведения аттестации;

- возможность присутствия ассистента и оказания им необходимой помощи (занять рабочее место, передвигаться, прочесть и оформить задание, общаться с преподавателем).

Формы промежуточной аттестации для инвалидов и лиц с ОВЗ должны учитывать индивидуальные и психофизические особенности обучающегося/обучающихся по АОПОП ВО (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.).

Специальные условия, обеспечиваемые в процессе преподавания дисциплины студентам с нарушениями зрения:

- предоставление образовательного контента в текстовом электронном формате, позволяющем переводить плоскочечатную информацию в аудиальную или тактильную форму;
- возможность использовать индивидуальные устройства и средства, позволяющие адаптировать материалы, осуществлять приём и передачу информации с учетом индивидуальных особенностей и состояния здоровья студента;
- предоставление возможности предкурсового ознакомления с содержанием учебной дисциплины и материалом по курсу за счёт размещения информации на корпоративном образовательном портале;
- использование чёткого и увеличенного по размеру шрифта и графических объектов в мультимедийных презентациях;
- использование инструментов «лупа», «проектор» при работе с интерактивной доской;
- озвучивание визуальной информации, представленной обучающимся в ходе занятий;
- обеспечение раздаточным материалом, дублирующим информацию, выводимую на экран;
- наличие подписей и описания у всех используемых в процессе обучения рисунков и иных графических объектов, что даёт возможность перевести письменный текст в аудиальный;
- обеспечение особого речевого режима преподавания: лекции читаются громко, разборчиво, отчётливо, с паузами между смысловыми блоками информации, обеспечивается интонирование, повторение, акцентирование, профилактика рассеивания внимания;
- минимизация внешнего шума и обеспечение спокойной аудиальной обстановки;
- возможность вести запись учебной информации студентами в удобной для них форме (аудиально, аудиовизуально, на ноутбуке, в виде пометок в заранее подготовленном тексте);
- увеличение доли методов социальной стимуляции (обращение внимания, апелляция к ограничениям по времени, контактные виды работ, групповые задания и др.) на практических и лабораторных занятиях;
- минимизирование заданий, требующих активного использования зрительной памяти и зрительного внимания;
- применение поэтапной системы контроля, более частый контроль выполнения заданий для самостоятельной работы.

Специальные условия, обеспечиваемые в процессе преподавания дисциплины студентам с нарушениями опорно-двигательного аппарата (маломобильные студенты, студенты, имеющие трудности передвижения и патологию верхних конечностей):

- возможность использовать специальное программное обеспечение и специальное оборудование и позволяющее компенсировать двигательное нарушение (коляски, ходунки, трости и др.);
  - предоставление возможности предкурсового ознакомления с содержанием учебной дисциплины и материалом по курсу за счёт размещения информации на корпоративном образовательном портале;
  - применение дополнительных средств активизации процессов запоминания и повторения;
  - опора на определенные и точные понятия;
  - использование для иллюстрации конкретных примеров;
  - применение вопросов для мониторинга понимания;
  - разделение изучаемого материала на небольшие логические блоки;
  - увеличение доли конкретного материала и соблюдение принципа от простого к сложному при объяснении материала;
  - наличие чёткой системы и алгоритма организации самостоятельных работ и проверки заданий с обязательной корректировкой и комментариями;
  - увеличение доли методов социальной стимуляции (обращение внимания, апелляция к ограничениям по времени, контактные виды работ, групповые задания др.);
  - обеспечение беспрепятственного доступа в помещения, а также пребывания в них;
  - наличие возможности использовать индивидуальные устройства и средства, позволяющие обеспечить реализацию эргономических принципов и комфортное пребывание на месте в течение всего периода учёбы (подставки, специальные подушки и др.).
- Специальные условия, обеспечиваемые в процессе преподавания дисциплины студентам с нарушениями слуха (глухие, слабослышащие, позднооглохшие):
- предоставление образовательного контента в текстовом электронном формате, позволяющем

переводить аудиальную форму лекции в плоскочечатную информацию;

- наличие возможности использовать индивидуальные звукоусиливающие устройства и сурдотехнические средства, позволяющие осуществлять приём и передачу информации; осуществлять взаимообратный перевод текстовых и аудиофайлов (блокнот для речевого ввода), а также запись и воспроизведение зрительной информации;
- наличие системы заданий, обеспечивающих систематизацию вербального материала, его схематизацию, перевод в таблицы, схемы, опорные тексты, глоссарий;
- наличие наглядного сопровождения изучаемого материала (структурно-логические схемы, таблицы, графики, концентрирующие и обобщающие информацию, опорные конспекты, раздаточный материал);
- наличие чёткой системы и алгоритма организации самостоятельных работ и проверки заданий с обязательной корректировкой и комментариями;
- обеспечение практики опережающего чтения, когда студенты заранее знакомятся с материалом и выделяют незнакомые и непонятные слова и фрагменты;
- особый речевой режим работы (отказ от длинных фраз и сложных предложений, хорошая артикуляция; четкость изложения, отсутствие лишних слов; повторение фраз без изменения слов и порядка их следования; обеспечение зрительного контакта во время говорения и чуть более медленного темпа речи, использование естественных жестов и мимики);
- чёткое соблюдение алгоритма занятия и заданий для самостоятельной работы (называние темы, постановка цели, сообщение и запись плана, выделение основных понятий и методов их изучения, указание видов деятельности студентов и способов проверки усвоения материала, словарная работа);
- соблюдение требований к предъявляемым учебным текстам (разбивка текста на части; выделение опорных смысловых пунктов; использование наглядных средств);
- минимизация внешних шумов;
- предоставление возможности соотносить вербальный и графический материал; комплексное использование письменных и устных средств коммуникации при работе в группе;
- сочетание на занятиях всех видов речевой деятельности (говорения, слушания, чтения, письма, зрительного восприятия с лица говорящего).

Специальные условия, обеспечиваемые в процессе преподавания дисциплины студентам с прочими видами нарушений (ДЦП с нарушениями речи, заболевания эндокринной, центральной нервной и сердечно-сосудистой систем, онкологические заболевания):

- наличие возможности использовать индивидуальные устройства и средства, позволяющие осуществлять приём и передачу информации;
- наличие системы заданий, обеспечивающих систематизацию вербального материала, его схематизацию, перевод в таблицы, схемы, опорные тексты, глоссарий;
- наличие наглядного сопровождения изучаемого материала;
- наличие чёткой системы и алгоритма организации самостоятельных работ и проверки заданий с обязательной корректировкой и комментариями;
- обеспечение практики опережающего чтения, когда студенты заранее знакомятся с материалом и выделяют незнакомые и непонятные слова и фрагменты;
- предоставление возможности соотносить вербальный и графический материал; комплексное использование письменных и устных средств коммуникации при работе в группе;
- сочетание на занятиях всех видов речевой деятельности (говорения, слушания, чтения, письма, зрительного восприятия с лица говорящего);
- предоставление образовательного контента в текстовом электронном формате;
- предоставление возможности предкурсового ознакомления с содержанием учебной дисциплины и материалом по курсу за счёт размещения информации на корпоративном образовательном портале;
- возможность вести запись учебной информации студентами в удобной для них форме (аудиально, аудиовизуально, в виде пометок в заранее подготовленном тексте);
- применение поэтапной системы контроля, более частый контроль выполнения заданий для самостоятельной работы;
- стимулирование выработки у студентов навыков самоорганизации и самоконтроля;
- наличие пауз для отдыха и смены видов деятельности по ходу занятия.

## **10. Методические рекомендации по освоению дисциплины (модуля)**