

**Компонент ОПОП 09.03.01 Информатика и вычислительная техника,  
направленность (профиль) «Технологии разработки программного обеспечения»**  
наименование ОПОП

**Б1.В.ДВ.03.01**  
шифр дисциплины

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**Дисциплины**  
**(модуля)**      **Разработка web-приложений на языке Python**

Разработчик (и):  
Ляш О.И.  
ФИО  
  
зав.кафедрой  
должность  
  
  
канд.пед.наук,  
доцент  
ученая степень,  
звание

Утверждено на заседании кафедры  
информационных технологий  
наименование кафедры  
  
протокол № 6 от 24.02.2026  
  
Заведующий кафедрой ИТ  
  
  
подпись  
  
Ляш О.И.  
ФИО

# 1. Критерии и средства оценивания компетенций и индикаторов их достижения, формируемых дисциплиной (модулем)

| Код и наименование компетенции                                                                        | Код и наименование индикатора(ов) достижения компетенции                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Результаты обучения по дисциплине (модулю)                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Оценочные средства текущего контроля                                                                                                                | Оценочные средства промежуточной аттестации |
|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <i>Знать</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <i>Уметь</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <i>Владеть</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                     |                                             |
| ПК-1<br>Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие | ИД-1 <sub>ПК-1</sub><br>Анализирует существующую документацию (техническое задание, спецификации) и требования к программному обеспечению для выделения перечня подлежащих разработке программных компонентов.<br>ИД-2 <sub>ПК-1</sub><br>Разрабатывает, согласовывает и утверждает технические спецификации на программные компоненты, описывающие их структуру, функции, логику работы и атрибуты качества.<br>ИД-3 <sub>ПК-1</sub><br>Проектирует интерфейсы взаимодействия программных компонентов (API) и протоколы обмена данными между ними, фиксируя результаты в спецификации.<br>ИД-4 <sub>ПК-1</sub><br>Оформляет техническую документацию на компоненты и их | <ul style="list-style-type: none"> <li>– Архитектурные принципы построения web-приложений: клиент-серверное взаимодействие, протокол HTTP/HTTPS, WSGI/ASGI.</li> <li>– Возможности и особенности web-фреймворков Python: Django/Flask/FastAPI.</li> <li>– Способы организации маршрутизации, middleware, шаблонизации, работы с формами и валидации данных.</li> <li>– Методы проектирования и интеграции реляционных баз данных через ORM.</li> <li>– Подходы к аутентификации, авторизации и</li> </ul> | <ul style="list-style-type: none"> <li>– Выбирать и обосновывать подходящий фреймворк под задачи проекта.</li> <li>– Проектировать структуру web-приложения: маршруты, контроллеры, шаблоны, модели БД, формы.</li> <li>– Реализовывать REST API с автоматической документацией и обработкой ошибок.</li> <li>– Настраивать аутентификацию и разграничение доступа без использования готовых «стартовых» проектов.</li> <li>– Писать интеграционные и модульные тесты для API и бизнес-логики, отлаживать приложение.</li> </ul> | <ul style="list-style-type: none"> <li>– Навыками разработки полноценного web-приложения на Django/Flask/FastAPI.</li> <li>– Приёмами работы с формами, валидацией данных и выводом сообщений об ошибках.</li> <li>– Техниками отладки (логирование, точка останова, отладчик) и профилирования запросов.</li> <li>– Навыками контейнеризации web-приложения в Docker и развёртывания на сервере.</li> <li>– Методами защиты от распространённых уязвимостей веб-приложений. декомпозиции на модули (сцены, контроллеры, сетевые менеджеры).</li> </ul> | <ul style="list-style-type: none"> <li>– комплект заданий для выполнения лабораторных (практических) работ;</li> <li>– тестовые задания;</li> </ul> | Результаты текущего контроля                |

|                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                          |  |  |  |  |
|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|--|--|--|--|
|                                                                     | взаимодействие в соответствии с требованиями стандартов.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | обеспечению безопасности web-приложений. |  |  |  |  |
| ПК-2<br>Способен проектировать компьютерное программное обеспечение | <p>ИД-1<sub>ПК-2</sub><br/>Выбирает архитектуру программного обеспечения и декомпозирует его на модули и компоненты.</p> <p>ИД-2<sub>ПК-2</sub><br/>Применяет объектно-ориентированные или иные методологии для разработки моделей предметной области, структур данных и бизнес-логики будущего ПО.</p> <p>ИД-3<sub>ПК-2</sub><br/>Разрабатывает логическую и физическую схемы баз данных, а также осуществляет проектирование пользовательских интерфейсов с учетом требований эргономики.</p> <p>ИД-4<sub>ПК-2</sub><br/>Создает проектную документацию на программное обеспечение, позволяющую осуществить кодирование и дальнейшее развертывание продукта.</p> <p>ИД-5<sub>ПК-2</sub><br/>Оценивает соответствие разработанного проекта исходным требованиям и техническому заданию, а также анализирует возможные риски реализации.</p> |                                          |  |  |  |  |



## 2. Оценка уровня сформированности компетенций (индикаторов их достижения)

| Показатели оценивания компетенций (индикаторов их достижения) | Шкала и критерии оценки уровня сформированности компетенций (индикаторов их достижения)                                                                                                                                         |                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                               | Ниже порогового<br>(«неудовлетворительно»)                                                                                                                                                                                      | Пороговый<br>(«удовлетворительно»)                                                                                                                                                                                                                             | Продвинутый<br>(«хорошо»)                                                                                                                                                                                                                       | Высокий<br>(«отлично»)                                                                                                                                                                                                                                                                  |
| <b>Полнота знаний</b>                                         | Уровень знаний ниже минимальных требований. Имели место грубые ошибки.                                                                                                                                                          | Минимально допустимый уровень знаний. Допущены не грубые ошибки.                                                                                                                                                                                               | Уровень знаний в объёме, соответствующем программе подготовки. Допущены некоторые погрешности.                                                                                                                                                  | Уровень знаний в объёме, соответствующем программе подготовки.                                                                                                                                                                                                                          |
| <b>Наличие умений</b>                                         | При выполнении стандартных заданий не продемонстрированы основные умения. Имели место грубые ошибки.                                                                                                                            | Продemonстрированы основные умения. Выполнены типовые задания с не грубыми ошибками. Выполнены все задания, но не в полном объеме (отсутствуют пояснения, неполные выводы)                                                                                     | Продemonстрированы все основные умения. Выполнены все основные задания с некоторыми погрешностями. Выполнены все задания в полном объёме, но некоторые с недочетами.                                                                            | Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Задания выполнены в полном объеме без недочетов.                                                                                                                     |
| <b>Наличие навыков (владение опытом)</b>                      | При выполнении стандартных заданий не продемонстрированы базовые навыки. Имели место грубые ошибки.                                                                                                                             | Имеется минимальный набор навыков для выполнения стандартных заданий с некоторыми недочетами.                                                                                                                                                                  | Продemonстрированы базовые навыки при выполнении стандартных заданий с некоторыми недочетами.                                                                                                                                                   | Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Продemonстрирован творческий подход к решению нестандартных задач.                                                                                                   |
| <b>Характеристика сформированности компетенции</b>            | Компетенции фактически не сформированы. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач.<br><br>ИЛИ<br>Зачетное количество баллов не набрано согласно установленному диапазону | Сформированность компетенций соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач.<br><br>ИЛИ<br>Набрано зачетное количество баллов согласно установленному диапазону | Сформированность компетенций в целом соответствует требованиям. Имеющихся знаний, умений, навыков достаточно для решения стандартных профессиональных задач.<br><br>ИЛИ<br>Набрано зачетное количество баллов согласно установленному диапазону | Сформированность компетенций полностью соответствует требованиям. Имеющихся знаний, умений, навыков в полной мере достаточно для решения сложных, в том числе нестандартных, профессиональных задач.<br><br>ИЛИ<br>Набрано зачетное количество баллов согласно установленному диапазону |

### 3. Критерии и шкала оценивания заданий текущего контроля

За выполнение всех видов работ на курсе можно получить 100 баллов. При выполнении работ студент получает первичный балл (ПБ) в диапазоне от 0 до 1 в соответствии с приведенными ниже таблицами. Все баллы, полученные студентом за текущий контроль и промежуточную аттестацию суммируются и пропорционально преобразуются к итоговому значению не превышающему величину 100 баллов.

Для «зачёта» и «зачёта с оценкой» за текущий контроль можно получить максимально 100 баллов.

Для «экзамена» за текущий контроль можно получить максимально 80 баллов, а за промежуточную аттестацию можно получить максимально «20 баллов».

#### 3.1 Критерии и шкала оценивания лабораторных работ

Перечень лабораторных, описание порядка выполнения и защиты работы, требования к результатам работы, структуре и содержанию отчета и т.п. представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

| Оценка                     | Баллы    | Критерии оценивания                                                                                                                                                                                                              |
|----------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Отлично</b>             | 0.9-1    | Задание выполнено полностью и правильно. Отчет по лабораторной/практической работе подготовлен качественно в соответствии с требованиями. Полнота ответов на вопросы преподавателя при защите работы.                            |
| <b>Хорошо</b>              | 0.89     | Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений. Все требования, предъявляемые к работе, выполнены. |
| <b>Удовлетворительно</b>   | 0.6-0.79 | Задания выполнены частично с ошибками. Демонстрирует средний уровень выполнения задания на лабораторную/практическую работу. Большинство требований, предъявляемых к заданию, выполнены.                                         |
| <b>Неудовлетворительно</b> | 0-0.59   | Задание выполнено со значительным количеством ошибок на низком уровне. Многие требования, предъявляемые к заданию, не выполнены.<br>ИЛИ<br>Задание не выполнено.                                                                 |

#### 3.2. Критерии и шкала оценивания тестирования

Перечень тестовых вопросов и заданий, описание процедуры тестирования представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

В ФОС включен типовый вариант тестового задания:

При разработке технической спецификации на REST API для Python-веб-приложения обязательным элементом описания каждого эндпоинта является:

- А) Только HTTP-метод
- В) HTTP-метод, URL, формат запроса/ответа, коды состояния
- С) Имя обработчика в коде
- Д) Версия Python и фреймворка

| Оценка             | Баллы | Критерии оценки             |
|--------------------|-------|-----------------------------|
| <b>Тест зачтён</b> | 0.6-1 | 60-100 % правильных ответов |

### 3.3. Критерии и шкала оценивания расчетно-графической работы

Перечень заданий для расчетно-графической работы, рекомендации по выполнению представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

В ФОС включен типовой вариант расчетно-графической работы:

#### Варианты предметных областей:

1. Система управления задачами (Task Manager) – создание, назначение, изменение статуса задач, комментарии, уведомления по email.
2. Блог-платформа – публикация статей, комментарии, теги, лайки, роли (админ, автор, читатель), подписка на обновления.
3. Интернет-магазин (минимальный функционал) – товары, корзина, заказы, пользователи, история покупок, генерация чека в PDF.
4. Система бронирования (отели, конференц-залы) – поиск доступных слотов, бронирование, отмена, личный кабинет.
5. API для мобильного приложения погоды с пользовательскими заметками – получение данных из внешнего API (mock или открытое API), сохранение заметок пользователя, геолокация.

#### Состав и содержание расчетно-графической работы

Работа включает пояснительную записку и графическую часть (схемы, диаграммы, макеты сцен).

1. Анализ требований и выделение компонентов
  - Описание функциональных требований к системе (по варианту).
  - Описание нефункциональных требований (производительность, безопасность, масштабируемость, удобство использования).
  - Выделение перечня программных компонентов веб-приложения: маршруты (контроллеры), сервисы бизнес-логики, ORM-модели, middleware, утилиты, внешние интеграции.
2. Разработка технических спецификаций на компоненты и их взаимодействие
  - Для каждого выделенного компонента описать: структуру, функции, логику работы, атрибуты качества.
  - Спроектировать интерфейсы взаимодействия компонентов:
    - Внутренние интерфейсы (между модулями Python: сигнатуры функций, классы, передача данных).
    - Внешние интерфейсы (REST API для фронтенда или мобильного клиента).
  - Описать протокол обмена данными: HTTP/HTTPS, форматы (JSON, возможно XML/FormData), коды состояния.
  - Зафиксировать спецификацию API в формате OpenAPI (Swagger) – предоставить фрагмент (YAML/JSON) в виде текста в приложении.
3. Проектирование архитектуры
  - Выбрать архитектуру веб-приложения (монолитная, модульный монолит, микросервисы). Обосновать выбор с учётом предметной области.
  - Выбрать фреймворк (Django, Flask или FastAPI) и обосновать его уместность.
  - Представить диаграмму компонентов (UML Component Diagram) и диаграмму развёртывания (Deployment Diagram) для выбранной архитектуры.
  - Выполнить декомпозицию системы на модули (приложения в Django, blueprints во Flask, routers в FastAPI). Описать взаимодействие модулей.

#### 4. Проектирование базы данных

- Разработать логическую схему БД (ER-диаграмма) с указанием сущностей, атрибутов, связей (1:1, 1:N, M:N).
- Разработать физическую схему БД для выбранной СУБД (PostgreSQL, MySQL или SQLite): типы данных, первичные/внешние ключи, индексы.
- Для реляционной БД привести SQL-скрипты создания таблиц (или код моделей ORM).
- При необходимости (если предметная область требует) – предложить NoSQL-решение и обосновать.

#### 5. Проектирование пользовательских интерфейсов (API и веб-интерфейса).

- Описать API как основной пользовательский интерфейс для внешних клиентов: методы, маршруты, параметры, примеры запросов и ответов.
- Привести сценарии ключевых операций (например, создание заказа, регистрация пользователя) в виде последовательности запросов.
- Спроектировать веб-интерфейс (набор шаблонов или страниц) с учётом требований эргономики: схема навигации, макет основных страниц (можно в виде wireframe). Для API-ориентированных систем – описание предполагаемого клиентского интерфейса (мобильного или SPA).

#### 6. Проектная документация и оценка рисков.

- Создать фрагмент проектной документации, достаточный для кодирования и развёртывания:
  - Описание конфигурационных переменных окружения.
  - Инструкция по установке зависимостей, выполнению миграций, запуску в режиме разработки.
  - Инструкция по запуску в production (WSGI/ASGI сервер, proxy).
- Оценить соответствие спроектированного решения исходным требованиям (составить таблицу трассировки требований к компонентам).
- Провести анализ возможных рисков реализации приложения:
  - Технические риски (производительность ORM, потенциальные узкие места в БД, сложность асинхронной логики).
  - Риски безопасности (SQL-инъекции, XSS, CSRF, подбор паролей, утечка данных).
  - Риски эксплуатации (сложность развёртывания, масштабирования, резервного копирования).
- Предложить меры по минимизации рисков (например, индексы БД, использование prepared statements, rate limiting, регулярные backup).

#### Графическая часть (обязательные схемы)

1. Диаграмма компонентов (UML Component Diagram) – компоненты приложения и их зависимости.
2. Диаграмма развёртывания (UML Deployment Diagram) – узлы (сервер БД, сервер приложений, клиент) и связи.
3. ER-диаграмма базы данных (логическая схема) – сущности, атрибуты, связи.
4. Диаграмма последовательности (UML Sequence Diagram) для одного ключевого сценария (например, «оформление заказа» или «публикация статьи»).

| Оценка         | Баллы | Критерии оценивания                                                                                                        |
|----------------|-------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Отлично</b> | 0.9-1 | Работа выполнена полностью, без ошибок (возможна одна неточность, описка, не являющаяся следствием непонимания материала). |



|                            |          |                                                                                                                                                                                      |
|----------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Хорошо</i>              | 0.8-0.89 | Работа выполнена полностью, но обоснования шагов решения недостаточны, допущена одна негрубая ошибка или два-три недочета, не влияющих на правильную последовательность рассуждений. |
| <i>Удовлетворительно</i>   | 0.6-0.79 | В работе допущено более одной грубой ошибки или более двух-трех недочетов, но обучающийся владеет обязательными умениями по проверяемой теме.                                        |
| <i>Неудовлетворительно</i> | 0-0.59   | В работе есть грубые ошибки и недочеты<br>ИЛИ<br>Контрольная работа не выполнена.                                                                                                    |

### 3.3. Критерии и шкала оценивания реферата

Тематика рефератов по дисциплине (модулю), требования к структуре, содержанию и оформлению изложены в методических материалах по освоению дисциплины (модуля), представлены в электронном курсе в ЭИОС МАУ.

В ФОС включены примерные темы рефератов:

1. История развития технологий виртуальной реальности в веб-среде.
2. Сравнительный анализ веб-фреймворков Python: Django, Flask, FastAPI (архитектура, производительность, удобство разработки).
3. Принципы работы WSGI и ASGI — эволюция серверных интерфейсов для Python.
4. Шаблонизатор Jinja2: возможности, производительность, интеграция с Flask и Django.
5. Объектно-реляционное отображение (ORM) на примере SQLAlchemy и Django ORM: сходства и различия.
6. Миграции баз данных в веб-проектах на Python: инструменты Alembic, Django migrations.
7. Аутентификация и авторизация в Django: встроенная система, расширение модели пользователя, социальная аутентификация.
8. Разработка REST API с помощью Django REST Framework (DRF): сериализаторы, viewsets, permissions.
9. Создание высокопроизводительных API с использованием FastAPI: асинхронность, автодокументация, валидация Pydantic.
10. Сравнение JWT и сессионной аутентификации в веб-приложениях на Python: безопасность и масштабируемость.
11. Обеспечение безопасности веб-приложений на Python: защита от CSRF, XSS, SQL-инъекций и кликджекинга.
12. Кэширование в веб-приложениях на Python: стратегии, использование Redis и memcached.
13. Тестирование веб-приложений на Python: unit-тесты (pytest), интеграционные тесты API, тесты с Selenium.
14. Проектирование и реализация middleware в Django и FastAPI: логгирование, аутентификация, сжатие ответов.
15. Паттерны проектирования в веб-разработке на Python: MVC, репозиторий, сервисный слой, зависимостей инъекция.
16. Асинхронные веб-приложения на Python: сравнение FastAPI, Starlette, aiohttp, Django Channels.
17. Real-time функционал на Python: веб-сокеты с Django Channels и FastAPI WebSocket.
18. Обработка фоновых задач в веб-приложениях: Celery, Dramatiq, фоновые задачи FastAPI.
19. Контейнеризация Python-веб-приложений: написание Dockerfile, мультистейдж-сборка, docker-compose для связки с БД и кэшем.
20. CI/CD-пайплайны для Python-веб-проектов: GitHub Actions, GitLab CI, Jenkins (примеры настройки, тестирования и деплоя).

21. Развертывание Python-веб-приложений на облачных платформах (Heroku, Render, Yandex Cloud, AWS Elastic Beanstalk).
22. Использование production-серверов: Gunicorn vs Uvicorn vs Hypercorn, настройка Nginx как reverse-proxy.
23. Оптимизация запросов к базе данных в Django и SQLAlchemy: проблема N+1, индексы, агрегации.
24. Профилирование и оптимизация производительности веб-приложений на Python: django-debug-toolbar, py-spy, cProfile.
25. Мониторинг и логирование в продакшене: Sentry, ELK-стек (Elasticsearch, Logstash, Kibana), structured logging.
26. Обработка ошибок и graceful degradation в веб-приложениях на Python: глобальные хендлеры, fallback-функции.
27. Управление конфигурацией и секретами: переменные окружения, .env, Vault для Python-приложений.
28. Графовые API: реализация GraphQL на Python с использованием Graphene (Django/Flask).
29. Интеграция Python-веб-приложений с внешними API: примеры с requests, httpx, асинхронные вызовы.
30. Реализация пагинации, фильтрации и поиска в REST API на Django REST Framework и FastAPI.
31. Методологии разработки веб-приложений на Python: Agile, TDD, BDD (примеры для Flask/Django).
32. Сравнение Python (Django/Flask) и Node.js (Express/NestJS) для веб-разработки: производительность, модель параллелизма, экосистема.
33. PHP (Laravel/Symfony) против Python (Django/Flask): синтаксис, скорость выполнения, популярность в корпоративной среде.
34. Java (Spring Boot) и Python (FastAPI/Django): сравнение типизации, производительности, сложности настройки и времени разработки.
35. Go (Gin/Echo) против Python (FastAPI): асинхронность, горутины vs event loop, бенчмарки для высоконагруженных API.
36. Ruby on Rails и Django: философия «соглашение над конфигурацией», ORM, скорость прототипирования, сообщество.
37. C# (ASP.NET Core) и Python (Django): сравнение производительности, кроссплатформенности, поддержки асинхронности.
38. Сравнение шаблонизаторов: Jinja2 (Python) против Blade (PHP) против EJS (Node.js) — синтаксис, безопасность, производительность.
39. Организация фоновых задач: Celery (Python) против Bull (Node.js) против Sidekiq (Ruby) — архитектура и сценарии использования.
40. Сравнение ORM: SQLAlchemy/Django ORM (Python) vs Hibernate (Java) vs Entity Framework (C#) — подходы к работе с БД.
41. Веб-сокеты и real-time: Django Channels (Python) vs Socket.IO (Node.js) vs Action Cable (Rails) — удобство и масштабируемость.
42. Асинхронные веб-фреймворки: FastAPI (Python) против aiohttp против Sanic — сравнение производительности и удобства разработки.
43. Микросервисная архитектура: Python (Nameko, FastAPI) против Go (gRPC) против Java (Spring Cloud) — зрелость решений и overhead.
44. Сравнение экосистем пакетов: PyPI (Python) против npm (Node.js) против Maven (Java) — качество, безопасность, управление зависимостями.
45. Инструменты тестирования веб-приложений: pytest (Python) против JUnit (Java) против Mocha (Node.js) — фичи, плагины, скорость.
46. Развертывание и DevOps: Python (Gunicorn, uWSGI) против Node.js (PM2) против Go (статическая сборка) — удобство контейнеризации и оркестрации.

| Оценка                     | баллы    | Критерии оценки                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Отлично</b>             | 0.9-1    | Выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована ее актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объем, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы. |
| <b>Хорошо</b>              | 0.8-0.89 | Основные требования к реферату и его защите - выполнены, но при этом допущены недочеты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объем реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы.                                                              |
| <b>Удовлетворительно</b>   | 0.6-0.79 | Имеются существенные отступления от требований к реферированию. В частности, тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод.                                                                                                                                                 |
| <b>Неудовлетворительно</b> | 0-0.59   | Тема реферата не раскрыта, обнаруживается существенное непонимание проблемы.                                                                                                                                                                                                                                                                                                             |

### 3.4. Критерии и шкала оценивания эссе

Тематика эссе по дисциплине (модулю), требования к структуре, содержанию и оформлению представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

В ФОС включены примерные темы эссе:

1. История развития технологий виртуальной реальности.
2. Практическое применение технологий виртуальной реальности.
3. Перспективы развития технологий виртуальной реальности.

| Оценка                     | баллы    | Критерии оценки                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Отлично</b>             | 0.9-1    | Представлена собственная точка зрения (позиция, отношение) при раскрытии проблемы. Проблема раскрыта на теоретическом уровне, в связях и с обоснованиями, с корректным использованием обществоведческих терминов и понятий в контексте ответа. Предоставлена аргументация своего мнения с опорой на факты общественной жизни или личный социальный опыт.        |
| <b>Хорошо</b>              | 0.8-0.89 | Представлена собственная точка зрения (позиция, отношение) при раскрытии проблемы. Проблема раскрыта с корректным использованием терминов и понятий в контексте ответа (теоретические связи и обоснования не присутствуют или явно не прослеживаются). Представлена аргументация своего мнения с опорой на факты общественной жизни или личный социальный опыт. |
| <b>Удовлетворительно</b>   | 0.6-0.79 | Представлена собственная точка зрения (позиция, отношение) при раскрытии проблемы; проблема раскрыта при формальном использовании обществоведческих терминов. Представлена аргументация своего мнения с опорой на факты общественной жизни или личный социальный опыт без теоретического обоснования.                                                           |
| <b>Неудовлетворительно</b> | 0-0.59   | Представлена собственная точка зрения (позиция, отношение) при раскрытии проблемы, но проблема раскрыта не полностью. Аргументация своего мнения слабо связана с раскрытием проблемы.                                                                                                                                                                           |

### 3.5. Критерии и шкала оценивания мультимедийной презентации

Требования к структуре, содержанию и оформлению представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

| Оценка                     | баллы    | Критерии оценки                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Отлично</b>             | 0.9-1    | Презентация соответствует теме самостоятельной работы. Оформлен титульный слайд с заголовком. Сформулированная тема ясно изложена и структурирована, использованы графические изображения (фотографии, картинки и т.п.), соответствующие теме, выдержан стиль, цветовая гамма, использована анимация, звук. Логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению.<br>Работа оформлена и предоставлена в установленный срок. |
| <b>Хорошо</b>              | 0.8-0.89 | Презентация соответствует теме самостоятельной работы. Имеются неточности в изложении материала. Отсутствует логическая последовательность в суждениях. Не выдержан объем презентации, имеются упущения в оформлении. На дополнительные вопросы при защите даны неполные ответы. Работа оформлена и предоставлена в установленный срок.                                                                                                                                                                                 |
| <b>Удовлетворительно</b>   | 0.6-0.79 | Презентация соответствует теме самостоятельной работы. Сформулированная тема изложена и структурирована не в полном объеме. Не использованы графические изображения (фотографии, картинки и т.п.), соответствующие теме. Присутствуют существенные отступления от требований к составлению презентации. Допущены фактические ошибки в содержании или при ответе на дополнительные вопросы.                                                                                                                              |
| <b>Неудовлетворительно</b> | 0-0.59   | Работа не выполнена или не соответствует теме самостоятельной работы.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

### 3.6. Критерии и шкала оценивания посещаемости занятий

Посещение занятий обучающимися определяется в процентном соотношении

| Баллы    | Критерии оценки         |
|----------|-------------------------|
| 0.75-1   | посещаемость 75 - 100 % |
| 0.5-0.74 | посещаемость 50 - 74 %  |
| 0-0.49   | посещаемость менее 50 % |

### 3.7. Критерии и шкала оценивания своевременной сдачи контрольных точек

Своевременность сдачи контрольных точек обучающимися определяется в процентном соотношении

| Баллы    | Критерии оценки                  |
|----------|----------------------------------|
| 0.75-1   | Своевременность сдачи 75 - 100 % |
| 0.6-0.74 | Своевременность сдачи 50 - 74 %  |
| 0-0.5    | Своевременность сдачи 50 %       |

## 4. Критерии и шкала оценивания результатов обучения по дисциплине (модулю) при проведении промежуточной аттестации (зачёт с оценкой)

### Критерии и шкала оценивания результатов освоения дисциплины (модуля) с зачетом с оценкой

Если обучающийся набрал зачетное количество баллов согласно установленному диапазону по дисциплине (модулю), то он считается аттестованным с оценкой согласно шкале баллов для определения итоговой оценки:

| Оценка                     | Баллы    | Критерии оценивания                                                     |
|----------------------------|----------|-------------------------------------------------------------------------|
| <i>Отлично</i>             | 91 - 100 | Набрано зачетное количество баллов согласно установленному диапазону    |
| <i>Хорошо</i>              | 81 - 90  | Набрано зачетное количество баллов согласно установленному диапазону    |
| <i>Удовлетворительно</i>   | 60 - 80  | Набрано зачетное количество баллов согласно установленному диапазону    |
| <i>Неудовлетворительно</i> | менее 60 | Зачетное количество согласно установленному диапазону баллов не набрано |

## 5. Задания диагностической работы для оценки результатов обучения по дисциплине (модулю) в рамках внутренней и внешней независимой оценки качества образования

ФОС содержит задания для оценивания знаний, умений и навыков, демонстрирующих уровень сформированности компетенций и индикаторов их достижения в процессе освоения дисциплины (модуля).

Комплект заданий разработан таким образом, чтобы осуществить процедуру оценки каждой компетенции, формируемых дисциплиной (модулем), у обучающегося в письменной форме.

Содержание комплекта заданий включает: *тестовые задания*.

### Комплект заданий диагностической работы

|                                                                                                                     |                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ПК-1</b><br><b>Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие</b> |                                                                                                                                                                                                                                                                                                   |
| 1.                                                                                                                  | При разработке технической спецификации на REST API для Python-веб-приложения обязательным элементом описания каждого эндпоинта является:<br><br>A) Только HTTP-метод<br>B) HTTP-метод, URL, формат запроса/ответа, коды состояния<br>C) Имя обработчика в коде<br>D) Версия Python и фреймворка  |
| 2.                                                                                                                  | В технической спецификации на компонент «Сервис аутентификации» на Django/Flask необходимо описать:<br><br>A) Только алгоритм хеширования паролей<br>B) Входные и выходные данные, логику работы, зависимости, обработку ошибок<br>C) Цвет кнопки входа в интерфейс<br>D) Тип сервера базы данных |
| 3.                                                                                                                  | Какой формат описания API чаще всего используется для автоматической генерации документации в Django REST Framework и FastAPI?<br><br>A) WSDL<br>B) RAML<br>C) OpenAPI (Swagger)<br>D) GraphQL Schema                                                                                             |
| 4.                                                                                                                  | При проектировании взаимодействия двух микросервисов на Python (один на FastAPI, другой на Django) наиболее предпочтительным протоколом для синхронного обмена с низкой задержкой является:<br><br>A) AMQP (RabbitMQ)<br>B) HTTP/JSON с keep-alive<br>C) FTP<br>D) UDP                            |

|     |                                                                                                                                                                                                                                                                                                                                                  |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5.  | <p>В спецификации на компонент «Логгер» для Python-веб-приложения атрибут качества «уровень детализации» должен позволять настраивать:</p> <p>A) Только включение/выключение записи<br/> B) Минимальный уровень серьёзности (DEBUG, INFO, WARNING, ERROR) и форматы вывода<br/> C) Имя хоста и PID процесса<br/> D) Только размер файла лога</p> |
| 6.  | <p>Какое описание интерфейса между компонентами (модулями Python) является фиксацией в технической документации?</p> <p>A) UML-диаграмма классов без методов<br/> B) Функциональная спецификация с сигнатурами функций, типами аргументов и возвращаемых значений, ожидаемым поведением<br/> C) Текст лицензии<br/> D) Схема базы данных</p>     |
| 7.  | <p>При разработке спецификации на WebSocket-взаимодействие (Django Channels или FastAPI) необходимо определить:</p> <p>A) Только IP-адрес сервера<br/> B) События (on_connect, on_message, on_disconnect), формат сообщений (JSON/бинарные), обработку разрывов<br/> C) Используемый HTTP-порт по умолчанию<br/> D) Наличие SSL-сертификата</p>  |
| 8.  | <p>Какой стандарт оформления технической документации на программные компоненты рекомендуется в отечественной разработке для Python-проектов?</p> <p>A) IEEE 829<br/> B) ГОСТ 19 (ЕСПД)<br/> C) ISO 25010<br/> D) PEP 8</p>                                                                                                                      |
| 9.  | <p>При согласовании спецификации на компонент «Кэш на Redis» для Python-веб-приложения необходимо указать:</p> <p>A) Стратегии инвалидации и время жизни ключей (TTL)<br/> B) Только IP-адрес сервера Redis<br/> C) Название Python-библиотеки (redis-py)<br/> D) Внутреннее устройство Redis</p>                                                |
| 10. | <p>В спецификации на взаимодействие компонентов API REST атрибут «идемпотентность» имеет наибольшее значение для метода:</p> <p>A) POST<br/> B) GET<br/> C) PUT / DELETE<br/> D) OPTIONS</p>                                                                                                                                                     |

#### **ПК-2**

#### ***Способен проектировать компьютерное программное обеспечение***

|     |                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11. | <p>Какой архитектурный паттерн лежит в основе большинства веб-фреймворков Python (Django, Flask)?</p> <p>A) Actor model<br/> B) MVC (Model-View-Controller) или его вариации (MTV в Django)<br/> C) Microkernel<br/> D) Blackboard</p>                                                                                                                                                          |
| 12. | <p>При проектировании базы данных для интернет-магазина на Django наиболее правильным способом реализации связи «пользователь – заказы» (один ко многим) является:</p> <p>A) Внешний ключ user_id в таблице заказов<br/> B) Поле orders в модели пользователя как JSON<br/> C) Отдельная связующая таблица с двумя внешними ключами<br/> D) Хранение заказов в NoSQL-документе пользователя</p> |
| 13. | <p>Какая диаграмма UML лучше всего подходит для документирования развёртывания Python-приложения (сервер приложений, БД, кэш, балансировщик)?</p>                                                                                                                                                                                                                                               |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <p>A) Диаграмма классов</p> <p>B) Диаграмма последовательности</p> <p>C) Диаграмма развёртывания (deployment diagram)</p> <p>D) Диаграмма состояний</p>                                                                                                                                                                                                                                                                                              |
| 14. | <p>В рамках объектно-ориентированного проектирования Python-модуля, какой принцип SOLID означает, что модуль должен зависеть от абстракций, а не от конкретных реализаций?</p> <p>A) Single Responsibility</p> <p>B) Open/Closed</p> <p>C) Liskov Substitution</p> <p>D) Dependency Inversion</p>                                                                                                                                                    |
| 15. | <p>При проектировании пользовательского интерфейса API для мобильного клиента на Django REST Framework рекомендуется использовать:</p> <p>A) Только GET-запросы для всех операций</p> <p>B) Ресурсно-ориентированный дизайн (REST) с правильными HTTP-глаголами и кодами ответов</p> <p>C) Только POST с указанием действия в теле запроса</p> <p>D) SOAP с XML</p>                                                                                  |
| 16. | <p>Оценка соответствия спроектированного Python-решения исходным требованиям (ИД-5ПК-2) предполагает:</p> <p>A) Только подсчёт строк кода</p> <p>B) Составление матрицы трассировки требований к функциям и компонентам</p> <p>C) Измерение времени отклика на главной странице</p> <p>D) Проверку наличия документации</p>                                                                                                                          |
| 17. | <p>Анализ риска «утечка памяти» в долго работающем Python-веб-приложении выполняется с помощью:</p> <p>A) Логирования всех запросов</p> <p>B) Профилирования (tracemalloc, memory_profiler) и мониторинга потребления памяти</p> <p>C) Увеличения объёма RAM</p> <p>D) Периодического перезапуска Gunicorn</p>                                                                                                                                       |
| 18. | <p>Какой метод уменьшает риск возникновения проблемы N+1 запросов при использовании Django ORM?</p> <p>A) Использование select_related и prefetch_related</p> <p>B) Применение необработанных SQL-запросов</p> <p>C) Увеличение количества соединений с БД</p> <p>D) Отключение кэша запросов</p>                                                                                                                                                    |
| 19. | <p>При проектировании системы с фоновыми задачами (отправка email, генерация отчётов) в Django наиболее подходящим решением является:</p> <p>A) Выполнение задач в основном потоке запроса</p> <p>B) Celery или Dramatiq с брокером сообщений (Redis/RabbitMQ)</p> <p>C) Использование threading внутри view</p> <p>D) Хранение задач в глобальной переменной</p>                                                                                    |
| 20. | <p>Предположим, вы проектируете обработку загрузки больших файлов (видео) через FastAPI. Какая архитектура минимизирует блокировку event loop?</p> <p>A) Чтение всего файла в память в синхронном режиме</p> <p>B) Использование UploadFile с асинхронным чтением по частям (await file.read())</p> <p>C) Вынос обработки в отдельный синхронный поток через run_in_executor</p> <p>D) Сохранение файла на диск через стандартный open без await</p> |

**Ключ к тесту:** 1-B; 2-B; 3-C; 4-B; 5-B; 6-B; 7-B; 8-B; 9-A; 10-C; 11-B; 12-A; 13-C; 14-D; 15-B; 16-B; 17-B; 18-A; 19-B; 20-C