

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Основы промышленной разработки»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Математика и искусственный интеллект

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва
2024**

Содержание

| | |
|--|---|
| 1. Краткая характеристика дисциплины (модуля) | 3 |
| 2. Перечень планируемых результатов обучения | 4 |
| 3. Тематический план | 6 |
| 4. Содержание дисциплины (модуля) | 6 |
| 5. Учебно-методическое обеспечение | 7 |
| 6. Материально-техническое обеспечение | 7 |
| 7. Методические и оценочные материалы | 9 |

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Основы промышленной разработки» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Математика и искусственный интеллект, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Основы промышленной разработки» позволяет овладеть ключевыми методами и технологиями, необходимыми для эффективного проектирования и внедрения промышленных решений. Это формирует практические навыки и системное мышление, способствующие успешной профессиональной деятельности в инженерной и производственной сферах.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Математика и искусственный интеллект и входит в обязательную часть Блока 1 как дисциплина по выбору.

Дисциплина (модуль) изучается на 1 курсе во 2 семестре.

Цель изучения дисциплины (модуля): формирование базовых знаний и практических навыков, необходимых для проектирования, анализа и внедрения промышленных технологий и процессов.

Задачи изучения дисциплины (модуля) направлены на формирование у студентов следующий знаний, умений и навыков:

- знание концепций ООП и их применение на практике;
- знание механизмов клиент-серверного взаимодействия приложений;
- знание процессов разработки в команде;
- умение интегрировать свою часть кода в командный проект;
- умение покрывать код различными видами тестов;
- умение разрабатывать свои классы с использованием наследования;
- умение взаимодействовать с базами данных программным образом;
- навык декомпозиции проекта по используемым технологиям и подзадачам;
- навык реализации полноценного веб-приложения с хранением данных и разграничением доступа.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

| Компетенция | Содержание компетенции | Индикатор компетенции | Перечень планируемых результатов обучения по дисциплине (модулю) |
|-------------|---|-----------------------|---|
| ОПК-1. | Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности | ОПК-1.1. | Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики |
| | | ОПК-1.2. | Умеет применять математические методы для решения профессиональных задач |
| | | ОПК-1.3. | Имеет практический опыт разработки и реализации математических моделей в профессиональной деятельности |
| ОПК-2. | Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности | ОПК-2.1. | Знает основные методы и подходы к проведению исследований в области искусственного интеллекта; научные и этические стандарты, применяемые в исследовательской практике |
| | | ОПК-2.2. | Умеет формулировать исследовательские вопросы и гипотезы на основе существующих знаний; анализировать и интерпретировать данные, полученные в ходе исследования |
| | | ОПК-2.3. | Имеет практический опыт участия в исследовательских проектах под научным руководством; применения существующих методов для решения практических задач в рамках исследовательской деятельности в области искусственного интеллекта |
| ОПК-5. | Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности | ОПК-5.1. | Знает технологии, необходимые для прикладного программирования, включая современные функциональные языки программирования, а также основные принципы и |

| | | | |
|--|--|----------|---|
| | | | понятия, применяемыми при использовании компьютерных сетей |
| | | ОПК-5.2. | Умеет пользоваться технологиями прикладного программирования, включая среды высокоуровневого программирования |
| | | ОПК-5.3. | Имеет практический опыт использования технологий прикладного программирования |

3. Тематический план

| №п/п | Наименование раздела дисциплины (модуля) | Трудоемкость, академические часы | | | | ТКУ (текущий контроль успеваемости) |
|--------|---|----------------------------------|-----------|----------|------------------------|-------------------------------------|
| | | <i>Очная форма</i> | | | | |
| | | Контактная работа | | Контроль | Самостоятельная работа | |
| Лекции | Семинары (практические занятия) | | | | | |
| 1 | Основы командной разработки | | 14 | | 23 | Домашние задания, Мини-проекты |
| 2 | Основы проектирования | | 14 | | 23 | Домашние задания, Мини-проекты |
| 3 | Веб-API | | 14 | | 23 | Домашние задания, Мини-проекты |
| 4 | Frontend | | 14 | | 23 | Домашние задания, Мини-проекты |
| | <i>Зачет с оценкой</i> | | | 4 | | Проект |
| | Итого: | | 56 | 4 | 92 | |
| | <i>Объем дисциплины (модуля) (в ак. ч.)</i> | 152 | | | | |
| | <i>Объем дисциплины (модуля) (в зач. ед.)</i> | 4 | | | | |

4. Содержание дисциплины (модуля)

| №п/п | Наименование раздела дисциплины (модуля) | Содержание дисциплины (модуля) по темам |
|------|--|--|
| 1 | Основы командной разработки | Git и GitLab Flow для командной работы. Unit-тесты. |
| 2 | Основы проектирования | ООП и обработка ошибок. Создание mock-объектов в тестах. |
| 3 | Веб-API | Разработка и документирование HTTP API. Работа с базами данных. Работа с ORM. Docker и Docker Compose. Continuous Integration (CI). Интеграционное тестирование. |
| 4 | Frontend | Основы HTML, CSS, JavaScript. DevTools в браузере. Концепция SPA, React. |

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2025. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567946>.

Дополнительная литература:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

| № | Наименование портала (издания, курса, документа) | Ссылка |
|----|--|---|
| 1. | Научная электронная библиотека elibrary.ru библиотека | https://elibrary.ru/defaultx.asp |
| 2. | База данных для IT-специалистов | https://habr.com |
| 3. | База данных ScienceDirect | https://www.sciencedirect.com |
| 4. | Официальный сайт Министерства науки и высшего образования Российской Федерации | https://minobrnauki.gov.ru/ |
| 5. | Федеральный портал «Российское образование» | https://www.edu.ru/ |
| 6. | Информационная система "Единое окно доступа к образовательным ресурсам" | http://window.edu.ru/ |
| 7. | Единая коллекция цифровых образовательных ресурсов | http://school-collection.edu.ru/ |
| 8. | Федеральный центр информационно - образовательных ресурсов | http://fcior.edu.ru/ |

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

| Наименование ПО | Производство | Лицензионное / свободно распространяемое |
|---|---------------|--|
| Операционные системы: | | |
| Microsoft Imagine (Windows Client, Server) | зарубежное | лицензионное |
| Браузеры: | | |
| Яндекс.Браузер | отечественное | свободно распространяемое |
| Google Chrome | зарубежное | свободно распространяемое |
| Офисные приложения: | | |
| Microsoft Imagine (Visio, OneNote) | зарубежное | лицензионное |
| TeXstudio | зарубежное | свободно распространяемое |
| Adobe Acrobat Reader | зарубежное | свободно распространяемое |
| Программное обеспечение для планирования и учета времени: | | |
| Toggle app | зарубежное | свободно распространяемое |
| Системы управления проектами: | | |
| Microsoft Imagine (Project) | зарубежное | лицензионное |
| Системы управления базами данных: | | |
| Microsoft Imagine (SQL Server) | зарубежное | лицензионное |
| Системы резервного копирования (backup): | | |
| Acronis Backup Advanced for HyperV | зарубежное | лицензионное |
| Справочно-правовые системы: | | |
| КонсультантПлюс: справочно-правовая система | отечественное | лицензионное |
| Средства антивирусной защиты: | | |
| Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition | отечественное | лицензионное |
| Среды разработки: | | |
| Visual Studio Code | зарубежное | свободно распространяемое |
| Bash (Unix shell) | зарубежное | свободно распространяемое |
| Anaconda | зарубежное | свободно распространяемое |
| Robotic Operating System | зарубежное | свободно распространяемое |
| CopelliaSim | зарубежное | свободно распространяемое |
| Google Colaboratory | зарубежное | свободно распространяемое |
| Пакеты программных средств и библиотек: | | |

| | | |
|--|------------|---------------------------|
| AutoPsy | зарубежное | свободно распространяемое |
| Interactive Disassembler (IDA) | зарубежное | свободно распространяемое |
| Системы управления библиографической информацией: | | |
| Zotero | зарубежное | свободно распространяемое |
| Сервисы и службы: | | |
| Bind | зарубежное | свободно распространяемое |
| Docker | зарубежное | свободно распространяемое |

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Основы промышленной разработки» в рамках текущего контроля успеваемости используются такие виды учебной работы, как семинары, мини-проекты, домашние задания, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Участие в семинаре – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Мини-проект — это краткосрочная практическая работа в малых группах на семинарах, направленная на применение теоретических знаний для решения конкретных задач.

В ходе мини-проекта студенты проводят анализ проблемы, разрабатывают техническое решение и представляют результаты в виде отчета, модели или презентации. Такой формат способствует развитию навыков самостоятельной работы, критического мышления и командного взаимодействия.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских

заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Основы промышленной разработки»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине. Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|---|
| 10 | Отлично | Зачтено | Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами. |
| 9 | Отлично | Зачтено | |
| 8 | Отлично | Зачтено | |
| 7 | Хорошо | Зачтено | Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. |
| 6 | Хорошо | Зачтено | |

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|---|
| | | | Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами. |
| 5 | Удовлетворительно | Зачтено | Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования. |
| 4 | Удовлетворительно | Зачтено | |
| 3 | Не сдан | Не зачтено | Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы. |
| 2 | Не сдан | Не зачтено | |
| 1 | Не сдан | Не зачтено | |

Дисциплина (модуль) «Основы промышленной разработки» оценивается следующим образом:

| Активность | Вес | Количество | Описание |
|------------------|-----|------------|--|
| Домашние задания | 20% | 12 | Набор задач по темам недели |
| Мини-проекты | 40% | 12 | Практическая работа в малых группах на семинарах |
| Зачет с оценкой | 40% | 1 | Защита итогового проекта |

Формула расчёта итоговой оценки по дисциплине (модулю) «Основы промышленной разработки»: $\langle 0,2 \times \text{среднее за домашние задания} + 0,4 \times \text{среднее за мини-проекты} + 0,4 \times \text{зачет с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине

Примерные домашние задания

Домашнее задание № 1

1. Создайте репозиторий, инициализируйте ветку develop и реализуйте GitLab Flow: создайте feature-ветку, сделайте коммиты, выполните merge request в develop.
2. Настройте .gitignore для проекта, исключите из коммитов временные и сгенерированные файлы.
3. Напишите несколько unit-тестов для простой функции (например, функции вычисления факториала) с использованием выбранного тестового фреймворка.
4. Организуйте code review с помощью Merge Request, прокомментируйте и исправьте

замечания.

5. Смоделируйте конфликт слияния в Git, разрешите его и завершите merge.

Домашнее задание № 2

1. Спроектируйте и реализуйте класс с инкапсуляцией и наследованием (например, базовый класс Animal и наследники Dog, Cat).
2. Добавьте обработку исключений при вводе некорректных данных в методы класса.
3. Напишите unit-тесты для класса с использованием mock-объектов для внешних зависимостей (например, замокайте вызов базы данных или внешнего API).
4. Реализуйте интерфейс и несколько классов, реализующих его, и продемонстрируйте полиморфизм.
5. Создайте тест, который проверяет корректное выбрасывание и обработку исключений.

Домашнее задание № 3

1. Разработайте RESTful API для CRUD-операций с сущностью (например, «заказ»), используя выбранный фреймворк.
2. Создайте и настройте базу данных, реализуйте доступ к ней через ORM.
3. Напишите документацию API (например, с помощью Swagger/OpenAPI).
4. Создайте Dockerfile и docker-compose для запуска приложения и базы данных в контейнерах.
5. Настройте CI-пайплайн, который автоматически запускает unit- и интеграционные тесты при пуше в репозиторий.

Домашнее задание № 4

1. Создайте простую веб-страницу с использованием HTML и CSS, реализуйте адаптивный дизайн.
2. Напишите скрипт на JavaScript, который динамически изменяет содержимое страницы по событию пользователя (например, клик по кнопке).
3. Используйте DevTools для отладки JavaScript-кода, найдите и исправьте ошибку в скрипте.
4. Создайте простое SPA-приложение на React с несколькими компонентами и навигацией без перезагрузки страницы.
5. Реализуйте форму с валидацией на React, отображайте ошибки ввода и успешную отправку.

Примерные задания для мини-проектов

Мини-проект № 1

Проект: Разработка консольного приложения «ToDo List» с использованием Git и GitLab Flow

Описание:

- Команда из 3-4 человек разрабатывает простое консольное приложение для управления списком задач.
- Используется GitLab Flow: основная ветка main, ветки для фич и багфиксов.
- Каждый участник создает отдельные ветки для своих задач, делает коммиты, создает merge request и проходит код-ревью.
- В проекте реализованы unit-тесты для основных функций (добавление, удаление, пометка задачи выполненной).
- Для тестирования используется популярный фреймворк (например, pytest для Python или JUnit для Java).

Цели:

- Освоить базовые команды Git и рабочий процесс GitLab Flow.
- Научиться писать и запускать unit-тесты.

- Понять важность код-ревью и совместной работы.

Мини-проект № 2

Проект: Моделирование системы банковских счетов с обработкой ошибок и тестированием с mock-объектами

Описание:

- Создать классы Account, Transaction, Bank с использованием принципов ООП (инкапсуляция, наследование, полиморфизм).
- Реализовать методы для пополнения счета, снятия средств и перевода между счетами.
- Ввести обработку ошибок: недостаточно средств, неверные параметры и т.п.
- Написать unit-тесты с использованием mock-объектов для имитации внешних сервисов (например, проверки кредитного рейтинга или логирования).
- Использовать фреймворк для моков (например, unittest.mock в Python или Mockito в Java).

Цели:

- Попрактиковаться в проектировании классов и обработке исключений.
- Научиться создавать mock-объекты для изоляции тестируемых компонентов.
- Понять важность тестирования на разных уровнях.

Мини-проект № 3

Проект: REST API для управления библиотекой книг

Описание:

- Разработать RESTful API (например, на Flask/Django/Express) с основными CRUD-операциями для сущности «Книга».
- Использовать ORM (SQLAlchemy, Django ORM или аналог) для работы с базой данных (PostgreSQL или MySQL).
- Документировать API с помощью OpenAPI/Swagger.
- Создать Dockerfile и docker-compose.yml для запуска приложения и БД.
- Настроить CI (например, GitLab CI или GitHub Actions) для автоматического запуска тестов и сборки контейнеров.
- Написать интеграционные тесты, которые проверяют взаимодействие API с базой данных.

Цели:

- Освоить создание и документирование REST API.
- Научиться работать с базой данных через ORM.
- Познакомиться с контейнеризацией и оркестрацией через Docker Compose.
- Настроить CI/CD процесс.
- Практиковаться в интеграционном тестировании.

Мини-проект № 4

Проект: Одностраничное приложение (SPA) «Список задач» на React

Описание:

- Создать SPA с использованием React, реализовать функционал добавления, удаления и пометки задач.
- Использовать базовые HTML и CSS для верстки.
- Реализовать интерактивность на JavaScript/React.
- Использовать DevTools для отладки и профилирования приложения.
- Добавить простую навигацию без перезагрузки страницы (React Router).
- Опционально — подключить локальное хранение задач (localStorage).

Цели:

- Освоить основы HTML, CSS и JavaScript в контексте React.
- Понять концепцию SPA и работу с виртуальным DOM.

- Научиться использовать DevTools для отладки.
- Познакомиться с React Router и управлением состоянием.

Примерное описание и критерии оценивания к итоговому проекту

Описание итогового проекта:

Итоговый проект представляет собой комплексную разработку программного продукта командой студентов, включающую создание веб-приложения с использованием современных технологий и инструментов. В рамках проекта необходимо применить навыки командной разработки с использованием Git и GitLab Flow, реализовать объектно-ориентированное проектирование с обработкой ошибок и модульным тестированием (unit-тесты и mock-объекты), разработать и задокументировать HTTP API с использованием баз данных и ORM, настроить контейнеризацию через Docker и автоматизацию сборки и тестирования с помощью CI, а также реализовать клиентскую часть с использованием HTML, CSS, JavaScript и React, обеспечив SPA-архитектуру и удобство пользовательского интерфейса. По завершении разработки команда должна подготовить презентацию и защитить проект, продемонстрировав технические решения, результаты работы и ответив на вопросы аудитории.

Критерии оценки итогового проекта:

- 1. Командная работа и управление версиями (Git, GitLab Flow):**
 - Использование ветвления и слияния согласно GitLab Flow.
 - Наличие понятной истории коммитов и описаний.
 - Эффективное распределение задач и взаимодействие в команде.
- 2. Качество проектирования и тестирования:**
 - Применение принципов ООП и грамотная обработка ошибок.
 - Наличие unit-тестов с использованием mock-объектов.
 - Покрытие кода тестами и качество тестов.
- 3. Backend-разработка и API:**
 - Корректная реализация HTTP API с полной документацией.
 - Использование баз данных и ORM для управления данными.
 - Настройка Docker и Docker Compose для контейнеризации.
 - Реализация интеграционных тестов.
 - Настройка и использование CI для автоматизации сборки и тестирования.
- 4. Frontend-разработка:**
 - Использование HTML, CSS и JavaScript согласно современным стандартам.
 - Реализация SPA с использованием React.
 - Качество пользовательского интерфейса и удобство взаимодействия.
 - Использование DevTools для отладки и оптимизации.
- 5. Документация и презентация проекта:**
 - Наличие технической документации по проекту.
 - Демонстрация работоспособности и основных функций.
 - Качество и полнота презентации итогов проекта.

6. Защита проекта:

- Умение четко и полно представить цели, задачи и результаты работы.
- Обоснование выбранных технических решений.
- Ответы на вопросы комиссии, демонстрирующие понимание материала и командную работу.
- Способность показать взаимодействие компонентов проекта и практическую пользу решения.

Задания для промежуточной аттестации по дисциплине (модулю)

| № п/п | Задание | Ответ | Компетенция |
|-------|---|--------------------------|-------------|
| 1. | Какой результат выполнения следующего кода? <pre>a = 2.5 b = 4.0 result = a * b + 3 print(result)</pre> | 13/13.0 | ОПК-2 |
| 2. | Напиши команду, с помощью которой можно добавить изменения всех файлов, которые находятся текущей директории, или во вложенных директориях, в git перед коммитом? | git add./ git add - A | ОПК-1 |
| 3. | Какой инструмент используется для скачивания библиотек в Python? <pre>A. pip B. venv C. requests D. Pycharm</pre> | A | ОПК-5 |