

УТВЕРЖДЕНА

Приказом Ректора АНО ВО
«Центральный университет»
Е.В. Ивашкевич
от «26» июня 2025 г. № 0626.32

**Рабочая программа дисциплины (модуля)
«Основы промышленной разработки»
дополнительной профессиональной программы – программы
профессиональной переподготовки «Академия data science»**

Траектория: Машинное обучение

**Москва
2025**

Содержание

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

1. Краткая характеристика дисциплины (модуля)

Изучение дисциплины (модуля) «Основы промышленной разработки» обеспечивает слушателей пониманием ключевых процессов и методов, используемых в промышленной разработке, что является основой для успешной карьеры в инженерных и технических областях. Освоение основ промышленной разработки способствует развитию критического мышления и способности к решению сложных задач.

Цель изучения дисциплины (модуля): формирование у слушателей знаний и навыков для эффективной разработки, тестирования и интеграции веб-приложений в командной среде с использованием клиент-серверной архитектуры и баз данных.

Задачи изучения дисциплины (модуля):

- изучить принципы организации и управления процессами командной разработки программного обеспечения;
- освоить методы декомпозиции сложных проектов на технологические компоненты и задачи;
- научиться создавать и применять различные виды тестирования для обеспечения качества кода;
- развить умения программного взаимодействия с базами данных для хранения и обработки информации;
- приобрести навыки интеграции индивидуальных модулей в общий проект с учетом особенностей клиент-серверных приложений.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- механизмы клиент-серверного взаимодействия приложений;
- процессы разработки в команде.

уметь:

- интегрировать свою часть кода в командный проект;
- покрывать код различными видами тестов;
- взаимодействовать с базами данных программным образом.

владеть:

- навыком декомпозиции проекта по используемым технологиям и подзадам;
- навыком реализации полноценного веб-приложения с хранением данных в базе данных.

2. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы			ТКУ (текущий контроль успеваемости)
		Очная форма			
		Аудиторная работа	Контроль	Самостоятельная работа	
Семинары (практические занятия)					
1	Командная работа	11		11	Подготовка элементов командного проекта 1
2	Объектно-ориентированное программирование (ООП)	11		11	Подготовка элементов командного проекта 1
3	Backend	11		11	Защита командного проекта 1
4	Базы данных	11		11	Подготовка элементов командного проекта 2
5	ORM	11		11	Подготовка элементов командного проекта 2
6	Тестирование	11		11	Защита командного проекта 2
7	DevOps	11		12	Подготовка элементов командного проекта 3
8	Observability	11		12	Подготовка элементов командного проекта 3
9	Frontend	11		12	Защита командного проекта 3
10	Документация/Диаграммы	11		12	Подготовка элементов индивидуального проекта
11	Инфраструктура и инструменты	12		12	Подготовка элементов индивидуального проекта
	<i>Зачет с оценкой</i>		4		Защита индивидуального проекта
Итого:		78	4	82	
Объем дисциплины (модуля) (в ак. ч.)		162			

3. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Командная работа	Введение и командная работа
2	Объектно-ориентированное программирование (ООП)	Основы ООП
3	Backend	REST API основы. Swagger / OpenAPI
4	Базы данных	Базы данных и SQL
5	ORM	ORM (SQLAlchemy)
6	Тестирование	Unit-тесты: unittest + mock. Pytest и coverage
7	DevOps	Docker. CI/CD (GitLab)
8	Observability	Observability: Logging, Metrics, Monitoring
9	Frontend	HTML/CSS/JS
10	Документация/Диаграммы	Diagrams as Code (Mermaid, PlantUML)
11	Инфраструктура и инструменты	Инфраструктура и современные инструменты

4. Учебно-методическое обеспечение

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

Каждый слушатель 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.

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

Основная литература:

1. Чакон С., Штрауб Б. Git для профессионального программиста. — СПб.: Питер, 2016. — 496 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-496-01763-3.

Дополнительная литература:

1. Russell, Stuart, Norvig, Peter. Artificial Intelligence: A Modern Approach. 3 : Prentice Hall, 2010.

5. 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;
- механическими калькуляторами;
- 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 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

6. 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 практические занятия, командные проекты, индивидуальный проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Практическое занятие — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где слушатели активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к практическому занятию рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Командный проект – это коллективная форма учебной деятельности, при которой слушатели в группе разрабатывают и реализуют задание по дисциплине, подготавливая части проекта поэтапно в соответствии с пройденными темами, и демонстрируя навыки командной работы и интеграции индивидуальных вкладов.

Для успешной подготовки к командному 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.

Индивидуальный проект – это форма учебной деятельности, где слушатель самостоятельно выполняет работу над исследованием, демонстрируя личные знания и навыки в решении поставленных задач.

Для успешной подготовки к индивидуальному проекту: тщательно спланируйте этапы работы, определите необходимые ресурсы и сроки, а также регулярно оценивайте свой прогресс. Обратитесь за консультацией к преподавателю при возникновении трудностей, чтобы своевременно скорректировать подход.

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

В процессе самостоятельной работы слушатели взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи слушателя 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)

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*.

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	Слушатель полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). 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. Слушатель 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	Слушатель 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. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи.
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Слушатель 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. Слушатель способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Слушатель 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	Описание
Командные проекты	50%	Групповая исследовательская работа по пройденным темам и презентация результатов
Индивидуальный проект	50%	Самостоятельная исследовательская работа по дисциплине (модулю) и презентация результатов

Формула расчёта итоговой оценки по дисциплине (модулю) «Основы промышленной разработки»: « $0,5 \times$ командные проекты + $0,5 \times$ индивидуальный проект».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для командных проектов

Командный проект 1.

Описание проекта:

Этот командный проект направлен на создание базового веб-приложения, демонстрирующего принципы командной работы, объектно-ориентированного программирования (ООП) и основ REST API с документацией через Swagger/OpenAPI. Слушатели разрабатывают приложение, такое как простой блог или система управления задачами, где команды применяют навыки введения в командную работу, основ ООП, создания REST API и его документирования.

Этапы работы над проектом:

1. **Планирование и распределение ролей:** Команда определяет цели проекта, декомпозирует задачи (например, модели данных, endpoints API), распределяет роли

(разработчики backend, тестировщики) и устанавливает сроки с учетом пройденных тем по командной работе и ООП.

2. Разработка моделей и логики: Реализация классов и объектов на основе ООП для обработки данных, создание базовых REST API endpoints (GET, POST и т.д.) с использованием фреймворков, таких как Flask или FastAPI.

3. Документирование API: Интеграция Swagger/OpenAPI для автоматической генерации документации API, включая примеры запросов и ответов.

4. Тестирование и интеграция: Проведение ручного тестирования функциональности, объединение компонентов в единое приложение и демонстрация командной работы через регулярные встречи.

5. Презентация и рефлексия: Представление готового проекта, обсуждение трудностей и уроков, полученных в процессе командной разработки.

Командный проект 2.

Описание проекта:

Этот командный проект фокусируется на разработке системы для работы с данными, включая создание баз данных, использование ORM и написание тестов. Слушатели строят приложение, такое как система учета товаров или пользователей, применяя знания по базам данных и SQL, ORM (например, SQLAlchemy), а также unit-тестам с unittest, mock, Pytest и coverage.

Этапы работы над проектом:

1. Анализ требований и проектирование базы данных: Команда определяет структуру данных, проектирует схему БД с использованием SQL, распределяет задачи по работе с ORM и тестированием.

2. Реализация ORM и взаимодействия с БД: Написание моделей с помощью SQLAlchemy для CRUD-операций, интеграция с приложением для хранения и извлечения данных.

3. Разработка и написание тестов: Создание unit-тестов с unittest и mock для проверки функций, а также интеграция Pytest с coverage для оценки покрытия кода тестами.

4. Интеграция и отладка: Сбор компонентов в рабочее приложение, исправление ошибок на основе результатов тестирования и командное обсуждение прогресса.

5. Финализация и демонстрация: Завершение проекта, проведение финального тестирования и презентация результатов с анализом эффективности тестирования и работы команды.

Командный проект 3.

Описание проекта:

Этот командный проект посвящен созданию и развертыванию веб-приложения с акцентом на DevOps-практики и базовый frontend. Слушатели разрабатывают приложение, такое как дашборд для мониторинга, используя Docker для контейнеризации, CI/CD с GitLab, observability (logging, metrics, monitoring) и основы HTML/CSS/JS для интерфейса.

Этапы работы над проектом:

1. Проектирование архитектуры и контейнеризации: Команда планирует приложение, распределяет задачи по DevOps и frontend, настраивает Docker для контейнеризации компонентов и CI/CD пайплайны в GitLab.

2. Разработка frontend-компонентов: Создание простого интерфейса с HTML/CSS/JS для взаимодействия с пользователем, интеграция с backend через API.

3. Настройка observability и развертывания: Реализация logging, metrics и monitoring для отслеживания производительности, настройка CI/CD для автоматического развертывания с использованием GitLab.

4. **Тестирование и интеграция:** Проведение тестирования компонентов, объединение frontend и backend в контейнерах, проверка работы в развернутой среде.

5. **Мониторинг и презентация:** Финальная настройка мониторинга, демонстрация проекта и обсуждение опыта применения DevOps-практик в команде.

Примерное задания для индивидуального проекта

Задание для проекта: Разработка одностраничного приложения (SPA)

Цель проекта

Создать полноценное одностраничное веб-приложение, которое использует серверную (backend) и клиентскую (frontend) части. Приложение должно включать функционал аутентификации и авторизации пользователей, а также продемонстрировать основные навыки работы с HTML, CSS и JavaScript.

Этапы выполнения проекта

1. **Определение концепции приложения**

- Выберите тему для вашего приложения (например, To-Do List, блог, система управления задачами и т.д.).
- Определите основные функции, которые должны быть реализованы (например, регистрация, вход, создание, редактирование и удаление записей).

2. **Проектирование интерфейса**

- Создайте макет интерфейса приложения с использованием инструментов проектирования (например, Figma, Adobe XD).
- Определите структуру HTML-документа и стилизацию с помощью CSS.

3. **Разработка клиентской части (frontend)**

- Реализуйте HTML-страницы и стили CSS.
- Используйте JavaScript для реализации динамического поведения приложения (например, обработка событий, взаимодействие с API).
- Реализуйте функционал аутентификации (регистрация и вход) с использованием форм и валидации.

4. **Разработка серверной части (backend)**

- Выберите язык программирования и фреймворк для разработки сервера (например, Node.js с Express, Python с Flask).
- Реализуйте API для обработки запросов на регистрацию, вход, создание, редактирование и удаление данных.
- Настройте базу данных для хранения информации о пользователях и данных приложения (например, MongoDB, PostgreSQL).

5. **Интеграция клиентской и серверной частей**

- Настройте взаимодействие между клиентом и сервером с помощью AJAX или Fetch API.
- Обеспечьте защиту API с использованием аутентификации (например, JWT).

6. **Тестирование и отладка**

- Проведите тестирование приложения для выявления и исправления ошибок.
- Используйте инструменты разработчика браузера для отладки и оптимизации производительности.

7. **Документация**

- Создайте документацию для вашего проекта, включая инструкции по установке и запуску приложения.
- Опишите архитектуру приложения и используемые технологии.

8. **Презентация проекта**

- Подготовьте короткую презентацию, где вы расскажете о вашем приложении, его функционале и процессе разработки.

Критерии оценивания

1. **Функциональность (40%)**

- Полнота реализации функционала (аутентификация, CRUD операции).
- Корректная работа клиентской и серверной частей.
- 2. **Качество кода (30%)**
 - Чистота и читаемость кода.
 - Соответствие стандартам и лучшим практикам разработки.
- 3. **Дизайн и пользовательский интерфейс (20%)**
 - Эстетика и удобство использования интерфейса.
 - Адаптивность и кроссбраузерность приложения.
- 4. **Документация (10%)**
 - Наличие и качество документации, включая инструкции по установке и запуску приложения.

Итог:

В результате выполнения проекта слушатели должны продемонстрировать свои навыки в разработке полноценного веб-приложения, используя современные технологии и подходы, а также понять основы работы с клиентом и сервером в контексте SPA. Успех проекта будет оцениваться по функциональности, качеству кода, дизайну и документации.