

УТВЕРЖДЕНА

Приказом Ректора АНО ВО
«Центральный университет»
Ивашкевич Е.В.
от «19» января 2024 г. № 0119.37

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

Траектория: Машинное обучение

**Москва
2024**

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, необходимых для проектирования, разработки и внедрения промышленных систем и технологий в условиях современного производства.

Задачи изучения дисциплины (модуля):

— формирование знаний: концепции объектно-ориентированного программирования (ООП) и их применение на практике; механизмы клиент-серверного взаимодействия приложений; процессы разработки в команде;

— формирование умений: интегрировать свою часть кода в командный проект; покрывать код различными видами тестов; разрабатывать свои классы с использованием наследования; взаимодействовать с базами данных программным образом;

— формирование навыка: декомпозировать проект по используемым технологиям и подзадам; реализовать полноценное веб-приложение с хранением данных и разграничением доступа.

2. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Современные процессы разработки	1	4		21	Домашние задания
2	Объектно-ориентированное программирование	1	4		21	Домашние задания Тест
3	Использование и реализация API поверх HTTP	1	4		21	Домашние задания
4	Современные процессы разработки	1	4		21	Домашние задания
5	Взаимодействие с реляционными базами данных	1	4		21	Домашние задания
6	Фронтенд-разработка	2	4		21	Домашние задания
7	Проектная работа	2	5		22	Домашние задания
	<i>Зачет с оценкой</i>			4		Проект
	Итого:	9	29	4	148	
	Объем дисциплины (модуля) (в ак. ч.)	190				

3. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Современные процессы разработки	Система контроля версий git и процессы разработки в GitLab. Юнит-тестирование и инструменты контроля качества кода
2	Объектно-ориентированное программирование	Концепции объектно-ориентированного программирования. Обработка ошибок и создание своих классов исключений
3	Использование и реализация API поверх HTTP	Протокол HTTP для доступа к API. Реализация API через протокол HTTP
4	Современные процессы разработки	Docker-контейнеры и их использование в CI. Интеграционное тестирование
5	Взаимодействие с реляционными базами данных	Введение в SQL и выполнение простых запросов к БД. Работа с данными из таблиц БД как с объектами с помощью ORM
6	Фронтенд-разработка	Основы HTML, CSS и JavaScript. Инструменты разработки браузера; аутентификация и авторизация. Концепция одностраничных приложений (SPA)
7	Проектная работа	Реализация полноценного проекта, состоящего из серверной (backend) и клиентской (frontend) частей.

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 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

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 лекции, практические занятия, домашние задания, тест, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре (практическом занятии) – активная работа слушателей на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре слушателям 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 дисциплине (модулю) 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.

Тест – особая форма проверки знаний. Проводится после освоения одной или нескольких тем и свидетельствует о качестве понимания основных понятий изучаемого материала. Тестовые задания составлены к ключевым понятиям, основным разделам, важным терминологическим категориям изучаемой дисциплины (модуля).

Для подготовки к тесту необходимо знать терминологический аппарат дисциплины (модуля), понимать смысл научных категорий и уметь их использовать в профессиональной

лексике. Владение понятийным аппаратом, включённым в тестовые задания, позволяет преподавателю быстро проверить уровень понимания слушателями важных методологических категорий.

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
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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. 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
Домашние задания	50%	Набор задач по темам недели
Тест	20%	Набор заданий по теме на проверку знаний
Проект	30%	Исследовательская работа по дисциплине (модулю) и презентация результатов

Формула расчёта итоговой оценки по дисциплине (модулю) «Основы промышленной разработки»: « $0,5 \times$ среднее за домашние задания + $0,3 \times$ проект + $0,2 \times$ среднее за тесты».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание: Использование и реализация API поверх HTTP

1. Опишите основные принципы работы протокола HTTP. Как он используется для доступа к API?
2. Приведите примеры различных методов HTTP (GET, POST, PUT, DELETE) и объясните, в каких случаях каждый из них применяется.
3. Напишите простой пример реализации RESTful API с использованием языка программирования по вашему выбору (например, Python, JavaScript).
4. Как осуществляется обработка ошибок в API? Приведите примеры кодов состояния HTTP и их значения.
5. Объясните, что такое аутентификация и авторизация в контексте API. Как они могут быть реализованы?

Домашнее задание: Современные процессы разработки

1. Что такое Docker и как он используется в современном процессе разработки?
2. Опишите основные команды Docker, которые необходимы для создания и управления контейнерами.
3. Как интеграционное тестирование помогает в процессе разработки? Приведите примеры инструментов для его реализации.
4. Создайте Dockerfile для простого веб-приложения. Опишите основные шаги, которые вы включили в этот файл.
5. Каковы преимущества использования контейнеров в CI/CD процессах? Объясните, как они помогают в автоматизации развертывания приложений.

Домашнее задание: Взаимодействие с реляционными базами данных

1. Что такое SQL и для чего он используется? Приведите примеры простых SQL-запросов.
2. Напишите SQL-запрос для создания таблицы с несколькими полями. Опишите, какие типы данных вы использовали.
3. Как можно извлечь данные из таблицы с использованием SQL? Приведите пример запроса с условиями фильтрации.
4. Объясните, что такое ORM (Object-Relational Mapping). Как он упрощает работу с базами данных?
5. Напишите пример кода на языке программирования по вашему выбору, который демонстрирует использование ORM для выполнения CRUD операций с объектами.

Домашнее задание: Фронтенд-разработка

1. Опишите структуру HTML-документа. Какие основные элементы должны быть включены?
2. Как CSS используется для стилизации веб-страниц? Приведите примеры различных селекторов и их применения.
3. Напишите простой JavaScript-код, который изменяет текст на веб-странице при нажатии кнопки.
4. Объясните разницу между аутентификацией и авторизацией. Как они реализуются на фронтенде?
5. Что такое одностраничные приложения (SPA)? Каковы их преимущества и недостатки по сравнению с традиционными веб-приложениями?

Примерные задания для 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 демонстрировать 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.

Примерные тестовые задания

Тест по теме: Git, GitLab, юнит-тестирование, ООП и обработка ошибок

Часть 1. Система контроля версий Git и процессы разработки в GitLab (5 вопросов)

1. Что делает команда `git clone`?
 - a) Создает новый репозиторий
 - b) Копирует удаленный репозиторий на локальную машину
 - c) Добавляет изменения в индекс
 - d) Отправляет изменения на удаленный сервер
2. Какой файл обычно используется для описания правил игнорирования файлов в Git?
 - a) `.gitignore`
 - b) `.gitconfig`
 - c) `README.md`
 - d) `.gitkeep`
3. Что такое `merge request (MR)` в GitLab?
 - a) Запрос на удаление ветки
 - b) Запрос на слияние изменений из одной ветки в другую
 - c) Запрос на создание нового проекта
 - d) Запрос на откат коммита
4. Какой из следующих процессов обеспечивает автоматическую проверку кода при каждом коммите в GitLab?
 - a) Continuous Integration (CI)
 - b) Continuous Deployment (CD)
 - c) Code Review
 - d) Git Rebase
5. Что делает команда `git rebase`?
 - a) Объединяет несколько коммитов в один
 - b) Переносит одну ветку на основе другой
 - c) Создает новую ветку
 - d) Удаляет ветку

Часть 2. Юнит-тестирование и инструменты контроля качества кода (5 вопросов)

6. Что такое юнит-тестирование?
 - a) Тестирование всей системы в целом

- b) Тестирование отдельных модулей или функций программы
 - c) Тестирование пользовательского интерфейса
 - d) Тестирование производительности
7. Какой из следующих инструментов используется для юнит-тестирования в JavaScript?
- a) Jest
 - b) ESLint
 - c) Webpack
 - d) Babel
8. Что из перечисленного является целью статического анализа кода?
- a) Исполнение кода
 - b) Поиск ошибок и потенциальных проблем без запуска программы
 - c) Оптимизация скорости выполнения
 - d) Тестирование пользовательского интерфейса
9. Что означает понятие «code coverage»?
- a) Количество строк кода в проекте
 - b) Процент кода, покрытого тестами
 - c) Количество багов в коде
 - d) Время выполнения тестов
10. Какой из следующих инструментов предназначен для анализа качества кода?
- a) Git
 - b) ESLint
 - c) Docker
 - d) Postman

Часть 3. Концепции ООП, обработка ошибок и создание своих классов исключений (5 вопросов)

11. Какое из перечисленных понятий НЕ относится к основным принципам объектно-ориентированного программирования?
- a) Наследование
 - b) Инкапсуляция
 - c) Полиморфизм
 - d) Рекурсия
12. Что такое инкапсуляция?
- a) Способ создания новых классов на основе существующих
 - b) Скрытие внутренней реализации объекта и предоставление интерфейса для взаимодействия
 - c) Возможность объектов принимать разные формы
 - d) Обработка исключений
13. Как в большинстве языков программирования создаётся собственный класс исключения?
- a) Наследованием от базового класса Exception
 - b) Созданием функции с именем Exception
 - c) Использованием ключевого слова try
 - d) Импортом модуля exceptions

14. Что происходит при выбрасывании исключения (throw)?
- a) Код продолжает выполняться без изменений
 - b) Управление передаётся ближайшему блоку обработки исключений (catch)
 - c) Программа автоматически перезапускается
 - d) Исключение игнорируется
15. Какой блок кода используется для обработки исключений?
- a) try
 - b) catch
 - c) finally
 - d) Все перечисленные