
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол №2

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

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Математика и компьютерные науки

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2026

**Москва
2026**

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 компьютерные науки», 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 компьютерные науки» и входит в обязательную часть Блока 1, как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 или 3 курсе в 3,4 или 5 семестре на выбор.

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:

- проектирование и реализация иерархии классов с использованием принципов ООП и наследования, а также написание юнит-тестов для проверки корректности работы методов;
- разработка клиент-серверного веб-приложения с регистрацией пользователей, разграничением доступа и хранением данных в реляционной базе;
- организация командной работы с использованием системы контроля версий Git, распределение задач, интеграция кода и проведение код-ревью.

В результате освоения дисциплины (модуля), обучающийся должен:

знать:

- общее устройство операционной системы Linux;
- принципы работы с командной строкой bash;
- принципы работы системы контроля версий Git;
- принципы написания программного кода.

уметь:

- работать с командной строкой bash в Linux;
- использовать систему контроля версий Git;

владеть:

- навыком подключения к удалённым серверам и работать с ними;
- навыком работы с удалёнными Git-репозиториями в GitHub, GitLab и других системах;
- навыком написания "чистого" программного кода.

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-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности.
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики

	практического применения	ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-1.	Способен формулировать естественнонаучные задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий
		ПК-1.2.	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике
		ПК-1.3.	Имеет опыт работы с естественнонаучными задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач, а также участие в обследовании предметной области и согласовании требований с бизнес-пользователями
ПК-3	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях

3. Тематический план

№п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Практиче ские занятия			
1	Основы ОС Linux, работы с командной строкой и языком bash	10	10		42	Домашние задания
2	Система контроля версий Git	6	6		26	Домашние задания
3	Работа с удаленными серверами	4	4		16	Домашние задания
4	Контейнеризация и Docker	6	6		26	Домашние задания
5	Непрерывная интеграция	4	4		16	Домашние задания
	Зачет с оценкой			4		Контрольная работа
Итого:		30	30	4	126	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы ОС Linux, работы с командной строкой и языком bash	Установка Linux. Командная строка и основные команды. Файловая система. Права доступа. Процессы. Устройство systemd. Переменные, операторы. Конвейеры. Продвинутое команды. Работа с текстами из командной строки. Репозитории ПО и пакеты в Linux
2	Система контроля версий Git	Репозитории, коммиты, ветки, состояния файлов. Удалённые репозитории и синхронизация с ними. Отмена изменений. Стратегии слияния. Git Workflows
3	Работа с удаленными серверами	Удалённое подключение к серверам. SSH, SCP, FTP, SFTP. Работа через VNC. Инструменты командной строки для взаимодействия с сетевыми сервисами
4	Контейнеризация и Docker	Виртуализация и контейнеризация. Docker, podman. Образы и контейнеры. Docker и тома. Слои и их виды. Работа с Docker Registry. Безопасность. Docker Compose. Сервисы и их интеграция. Healthcheck. Безопасность в Compose
5	Непрерывная интеграция	Continuous Integration. Gitlab CI/Github Actions/Jenkins: основные понятия. Настройка агентов. Continuous Delivery/Continuous Deployment на примере Gitlab CI/Github Actions. Хранение секретных данных в CI/CD системах

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. Ганди, Р. Head First. Git : практическое руководство / Р. Ганди. - Санкт-Петербург : БХВ-Петербург, 2023. - 464 с. - ISBN 978-5-9775-1777-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2123358>.

2. Барретт, Д. Linux. Командная строка. Лучшие практики : практическое руководство / Д. Барретт. - Санкт-Петербург : Питер, 2023. - 256 с. - (Бестселлеры O'Reilly). - ISBN 978-5-4461-2300-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2123801>.

3. Дэвис, Д. Философия DevOps. Искусство управления IT : практическое руководство / Д. Дэвис, К. Дэниелс. - Санкт-Петербург : Питер, 2019. - 416 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-1141-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1760824>.

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>.

2. 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>.

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;
- 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

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 лекции, семинары, контрольная работа и домашние задания, 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.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару 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.

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.

Контрольная работа – письменная работа с набором задач, которые нужно решить за

ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

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, если иное не указано в syllabusе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	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	
8	Отлично	
7	Хорошо	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	

Десятибалльная оценка	Пятибалльная оценка	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	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	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

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

Активность	Вес	Описание
Домашние задания	70%	Набор задач по темам недели
Зачёт с оценкой	30%	Контрольная работа - письменная работа с набором задач, которые нужно решить за ограниченное время

Формула расчёта итоговой оценки по дисциплине (модулю) «Основы промышленной разработки»: $0,7 \times \text{среднее за домашние задания} + 0,3 \times \text{за контрольную работу}$.

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 контрольной работы

1. Выполните следующие действия в терминале Linux:

— Создайте директорию ~/lab1 и перейдите в неё.

— Создайте файл data.txt, содержащий строку: Hello, Linux!

— Скопируйте этот файл в backup_data.txt.

— Выведите содержимое обоих файлов на экран с нумерацией строк.

— Установите права на data.txt так, чтобы владелец мог читать и писать, группа — только читать, остальные — не имели доступа.

— Запишите команды, которые вы использовали.

2. Дан файл log.txt, содержащий строки вида:

2024-04-01 10:15:23 ERROR Failed to connect

2024-04-01 10:16:01 INFO User logged in

2024-04-01 10:17:30 ERROR Timeout

Напишите команду, которая:

— Выведет количество строк с ошибками (ERROR).

— Извлечёт все строки с ERROR и сохранит их в файл errors.log.

— Извлечёт из этих строк только временные метки (первые два поля) и отсортирует их по убыванию.

3. Ответьте на вопросы, приведя соответствующие команды:

— Как проверить текущее состояние репозитория (изменённые, добавленные,

неотслеживаемые файлы)?

- Как добавить в индекс все изменённые файлы, кроме temp.log?
- Как создать коммит с сообщением "Fix login bug"?
- Как отменить последний коммит, сохранив изменения в рабочей директории?

4. Ответьте на вопросы:

- Как загрузить изменения из ветки main удалённого репозитория?
- Как отправить вашу локальную ветку feature/auth в удалённый репозиторий и установить отслеживание?
- Как слить ветку dev в текущую ветку main, если возник конфликт в файле app.py? Опишите шаги.

5. Приведите команды для выполнения следующих действий:

- Подключиться по SSH к серверу с IP 192.168.1.100, пользователем admin.
- Скопировать файл config.json с локальной машины на сервер в директорию /home/admin/.
- Скачать файл report.log с сервера в текущую директорию.
- Проверить, открыт ли порт 80 на удалённом сервере example.com.

6. Напишите команду Docker, которая:

- Запускает контейнер на основе образа nginx:alpine в фоновом режиме.
- Назначает ему имя web-server.
- Пробрасывает порт 8080 хоста на порт 80 контейнера.
- Монтирует локальную директорию ~/site в /usr/share/nginx/html внутри контейнера.
- Как проверить, что контейнер запущен?

7. Дан фрагмент docker-compose.yml:

```
version: '3.8'

services:
  db:
    image: postgres:14
    environment:
      POSTGRES_DB: myapp
      POSTGRES_USER: user
      POSTGRES_PASSWORD: secret
    volumes:
      - db_data:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    volumes:
      db_data:
```

- Что делает секция volumes на верхнем уровне?
- Почему важно использовать именованный том вместо привязки к хосту?
- Как запустить этот стек и как остановить его с удалением контейнеров?

8. Дан .gitlab-ci.yml:

stages:

test

build

deploy

run-tests:

stage: test

script:

- echo "Running tests..."

- python -m pytest

build-image:

stage: build

script:

- docker build -t myapp:latest .

only:

- main

— На каких ветках будет запускаться задача run-tests?

— Почему задача build-image запускается только при коммитах в ветку main?

— Как добавить переменную DB_PASSWORD безопасным способом, чтобы она не попала в логи?

9. Напишите bash-скрипт, который:

— Объявляет переменную LOG_DIR="/var/log".

— Проверяет, существует ли директория, указанная в LOG_DIR.

— Если существует — выводит количество файлов в ней.

— Если нет — выводит ошибку и завершается с кодом 1.

10. Опишите пошагово, как организовать процесс CI/CD для приложения:

— Исходный код в GitLab.

— При коммите в ветку main должен:

— Запускаться тест (команда: npm test).

— Собираться Docker-образ.

— Образ пушится в GitLab Container Registry.

— На сервере через SSH запускается скрипт обновления контейнера.

— Укажите, какие файлы нужно создать, какие настройки добавить (включая секреты), и приведите пример .gitlab-ci.yml.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Что отображает команда ls в терминале Linux?	содержимое директории / список файлов	ОПК-1
2.	Что обозначает символ x в правах доступа к файлу в Linux?	разрешение на выполнение	ОПК-1

		/ выполнение	
3.	Что позволяет передавать вывод одной команды на вход другой в bash?	пайп /	ОПК-1
4.	По какому протоколу безопасно копируют файлы на удалённый сервер в Linux?	scp	ОПК-4
5.	Что делает команда git push в системе контроля версий?	отправляет изменения в удалённый репозиторий	ОПК-4
6.	Что выполняет команда git pull в Git?	получает и сливает изменения из удалённого репозитория	ОПК-4
7.	Что создаётся командой docker build ?	образ / Docker image	ОПК-6
8.	В каком файле описывается конфигурация сервисов для Docker Compose?	docker-compose.yml	ОПК-6
9.	На каком этапе CI/CD приложение разворачивается на сервере?	deploy	ОПК-6
10.	Что делает команда mkdir в терминале Linux?	создаёт директорию	ПК-1
11.	Что выводит на экран команда cat при работе с файлом?	содержимое файла	ПК-1
12.	Для чего используется команда cd в командной строке?	переход в директорию / смена директории	ПК-1
13.	Что делает команда git reset --soft HEAD~1 ?	отменяет последний коммит, сохраняя изменения	ПК-3
14.	Что происходит с файлом после выполнения git restore ?	восстанавливается до последнего сохранённого состояния	ПК-3

15.	Что позволяет команда <code>git checkout HEAD -- <file></code> ?	вернуть файл к состоянию последнего коммита	ПК-3
-----	--	---	------