
УТВЕРЖДЕНА

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

**Рабочая программа дисциплины (модуля)
«Инструменты разработчика»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Машинное обучение в промышленности

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	3
3. Тематический план.....	8
4. Содержание дисциплины (модуля).....	8
5. Учебно-методическое обеспечение	10
6. Материально-техническое обеспечение	10
7. Методические и оценочные материалы	12

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Инструменты разработчика» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Машинное обучение в промышленности, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Инструменты разработчика» позволяет сформировать у студентов практические навыки работы с ключевыми инструментами, используемыми в современной разработке программного обеспечения. Дисциплина (модуль) направлена на формирование устойчивых компетенций в области командной строки, систем контроля версий, сетевого взаимодействия, контейнеризации и эффективного.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Машинное обучение в промышленности и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору. Дисциплина (модуль) доступна к изучению на 2 курсе в 3 и 4 семестре на выбор.

Цель изучения дисциплины (модуля): подготовить студентов к уверенной и эффективной работе с основными инструментами разработчика: операционной системой Linux, Git, Docker, HTTP, а также научить использовать ИИ-инструменты для написания и отладки кода, обеспечивая готовность к участию в реальных проектах.

Задачи изучения дисциплины (модуля):

- изучить архитектуру и основные компоненты операционной системы Linux;
- освоить работу с командной строкой bash для навигации, управления файлами и автоматизации задач;
- освоить принципы и практику использования системы контроля версий Git и удалённых репозиториях;
- понять основы сетевого взаимодействия, работу протокола HTTP и диагностику клиент-серверных проблем;
- научиться контейнеризировать приложения с помощью Docker и писать чистый, поддерживаемый код с использованием ИИ-ассистентов.

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;
- виды сетевого взаимодействия и принципы протокола HTTP;
- принципы написания программного кода (в том числе, при помощи ИИ);
- отличия контейнеризации и виртуализации.

уметь:

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

- диагностировать и решать проблемы клиент-серверного взаимодействия веб-приложений;
- диагностировать и решать проблемы, связанные с генерацией кода при помощи ИИ;
- собирать контейнеризированные приложения при помощи инструмента Docker.

владеть:

- навыком подключаться к удалённым серверам и работать с ними;
- навыком работы с удалёнными Git-репозиториями в GitHub, GitLab и других системах;
- навыком отлаживать клиент-серверное взаимодействие приложений;
- навыком писать "чистый" программный код;
- навыком контейнеризировать собранные приложения.

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)
УК-6.	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1.	Знает основные методы самооценки и анализа своей деятельности, а также принципы управления временем и целеполагания
		УК-6.2.	Умеет ставить реалистичные и достижимые цели, определять приоритеты в своей деятельности, а также разрабатывать и внедрять планы по совершенствованию своих навыков и компетенций на основе полученной самооценки
		УК-6.3.	Имеет практический опыт применения методов самооценки в своей профессиональной деятельности, включая участие в тренингах, семинарах и проектах, направленных на развитие личной эффективности и профессионального роста
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области
		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и

			представлять результаты в понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи
ПК-1.	Способен определять общие формы и закономерности области машинного обучения в индустрии	ПК-1.1.	Знает основные теоретические концепции и принципы, относящиеся к области машинного обучения в индустрии, а также ключевые закономерности и модели, которые помогают в анализе и интерпретации данных
		ПК-1.2.	Умеет проводить систематический анализ области разработки, выявлять и формулировать общие закономерности и тенденции, а также применять методы исследования для получения новых знаний и понимания
		ПК-1.3.	Имеет практический опыт работы в области машинного обучения в индустрии, включая участие в научных проектах, исследованиях или практических заданиях, где были выявлены и описаны общие формы и закономерности
ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач,

			формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений
ПК-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере машинного обучения в индустрии	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в машинном обучении в индустрии
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Основы ОС Linux, работа с командной строкой	2	2		12	Домашние задания
2	Система контроля версий Git	2	3		12	Домашние задания
3	Продвинутые инструменты с разработкой командной строки, язык bash	2	3		12	Домашние задания
4	Система контроля версий Git	3	3		12	Домашние задания
5	Работа с удаленным сервером по SSH	3	3		12	Домашние задания
6	Системы сборки проектов на основе программного кода	4	4		15	Домашние задания
7	Основы сетевого взаимодействия	3	3	2	12	Контрольная работа
8	Контейнеризация и Docker	4	4		15	Домашние задания
9	Сетевое взаимодействие	4	4		15	Домашние задания
10	Принципы написания программного кода	3	3		12	Домашние задания
	Зачет с оценкой			4		
Итого:		30	28	6	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, работа с командной строкой	Установка Linux. Командная строка и основные команды
2	Система контроля версий Git	Введение. Репозитории, коммиты, ветки, состояния файлов Удалённые репозитории и синхронизация с ними

3	Продвинутые инструменты с разработкой командной строки, язык bash	Файловая система. Права доступа. Процессы. Устройство systemd Переменные, операторы. Конвейеры. Продвинутые команды
4	Система контроля версий Git	Отмена изменений. Стратегии слияния. Git Workflows
5	Работа с удаленным сервером по SSH	Удалённое подключение к серверам. SSH, SCP, FTP, SFTP. Работа через VNC
6	Системы сборки проектов на основе программного кода	Виды систем сборки. Механизмы работы систем сборки Сборка Java-проектов. Работа Maven и Gradle Сборка Python-проектов: pybind11, пакетные менеджеры poetry, uv
7	Основы сетевого взаимодействия	Сетевой стек ISO/OSI и TCP/IP. Сокеты. Маршрутизация. Работа с DNS.
8	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
9	Сетевое взаимодействие	Протокол HTTP. REST и RESTful API. Аутентификация и авторизация. Отладка API Принципы работы веб-серверов. Настройка Nginx. Протокол gRPC и интеграция с веб-серверами
10	Принципы написания программного кода	ИИ-разработка: провайдеры, безопасность, аудит. Понятие skills и agents

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. Менцер Филлиппо, Фортунато Санто, Дэвис Клейтон - Наука о сетях: вводный курс. ДМК Пресс, 2022 г., 338 с. <https://znanium.ru/catalog/document?id=446466>.
2. Гадзурас Эммануил - Docker Compose для разработчика - ДМК Пресс, 2023 г., 222 с. <https://znanium.ru/catalog/document?id=445334>.

Дополнительная литература:

1. Сمارт Джон Фергюсон, Молак Ян - BDD в действии - ДМК Пресс, 2023 г., 554 с. <https://znanium.ru/catalog/document?id=462885>.

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
CorollaSim	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) «Инструменты разработчика» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контрольная работа, 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 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) «Инструменты разработчика»

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 силлабусе 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 (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. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		практическими задачами.
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. 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.
6	Хорошо	
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) «Инструменты разработчика» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	50%	7	Набор задач по темам недели
Контрольная работа	20%	1	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Инструменты разработчика»: $0,5 \times \text{среднее за домашние задания} + 0,2 \times \text{контрольная работа} + 0,3 \times \text{зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1. Работа с Linux и Git

1. Установите Linux (или WSL) и выполните:
 - Навигацию по файловой системе (cd, ls)
 - Создание, копирование, удаление файлов и папок
 - Поиск файлов по имени (find) и содержимому (grep)
2. Создайте локальный Git-репозиторий:
 - Инициализируйте git init
 - Создайте файл readme.md, добавьте в индекс, сделайте коммит
 - Создайте ветку feature/login, внесите изменения, слейте с main
3. Подключитесь к GitHub, создайте удалённый репозиторий и выполните git push.
4. Оформите отчёт: скриншоты команд и краткие пояснения.

Домашнее задание 2. Работа с SSH, системами сборки и сетями

1. Подключитесь к удалённому серверу по SSH:
 - Сгенерируйте ключ (ssh-keygen)
 - Скопируйте публичный ключ (ssh-copy-id)
 - Выполните команды: whoami, df -h, ps aux
2. Соберите Python-проект:
 - Используйте poetry или uv для инициализации проекта
 - Добавьте зависимости (например, requests, flask)
 - Создайте pyproject.toml
3. Проверьте доступность сайта:
 - Используйте ping, nslookup, curl -I
 - Проанализируйте HTTP-заголовки
4. Оформите лог выполнения.

Домашнее задание 3. Контейнеризация и веб-сервер

1. Напишите Dockerfile для простого Flask-приложения:
 - Установите зависимости через pip
 - Скопируйте код, откройте порт 5000
 - Запустите приложение
2. Создайте docker-compose.yml:
 - Сервис приложения
 - Сервис PostgreSQL
 - Настройте healthcheck для БД
3. Настройте Nginx как обратный прокси:
 - Напишите конфиг, пробрасывающий запросы на localhost:5000
 - Запустите Nginx в контейнере
4. Проверьте работоспособность через браузер и curl.
5. Оформите README с инструкцией по запуску.

Примерные задания для контрольной работы

1. Напишите команду, которая находит все файлы с расширением `conf` в каталоге `etc`, сортирует их по дате изменения и выводит только имена файлов.
2. Объясните, как проверить, какие процессы используют сетевой порт 8080 на Linux-системе.
3. Как остановить все запущенные Docker-контейнеры и удалить их? Напишите соответствующие команды.
4. В чём разница между протоколами HTTP и HTTPS? Какую роль играет SSL/TLS в этом различии?
5. Что такое Docker-образ и как он отличается от Docker-контейнера?
6. Как настроить автоматический запуск службы в Linux с помощью `systemd`? Назовите основные шаги.
7. Объясните, зачем используется файл `docker-compose.yml` и какие элементы он может содержать.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Как определить приоритеты задач в проекте?	На основе влияния на бизнес, сроков и ресурсов (матрица приоритетов)	УК-6
2.	Как оценить свою эффективность на позиции разработчика?	По достижению целей, качеству кода, обратной связи команды	УК-6
3.	Как спланировать свое профессиональное развитие?	Через анализ навыков, постановку целей, обучение (курсы, книги)	УК-6
4.	Как управлять временем при работе над задачей?	Тайм-менеджмент, декомпозиция, фокус, исключение отвлечений	УК-6
5.	Как посмотреть список запущенных процессов в системе?	<code>ps aux</code> или <code>top</code>	ОПК-1
6.	Как отменить последний коммит, сохранив изменения в рабочей директории?	<code>git reset --soft HEAD~1</code>	ОПК-1
7.	Как подключиться к удалённому серверу по SSH под пользователем <code>dev</code> и адресом 192.168.1.10?	<code>ssh dev@192.168.1.10</code>	ОПК-1
8.	Как скопировать файл <code>app.log</code> на удалённый сервер через SCP?	<code>scp app.log dev@192.168.1.10 :/home/dev/</code>	ОПК-1
9.	Как собрать Java-проект с помощью Maven?	<code>mvn clean install</code>	ПК-1

10.	Как инициализировать Python-проект с помощью poetry?	poetry init, затем poetry install для установки зависимостей	ПК-1
11.	Что такое DNS и какова его основная функция?	Система доменных имён; преобразует понятные имена, например, example.com, в IP-адреса	ПК-1
12.	Что означает HTTP-статус код 200?	Успешное выполнение запроса, состояние ОК	ПК-1
13.	Какая команда в Linux показывает текущую директорию?	pwd	ПК-3
14.	Что делает команда ls -la?	Выводит список файлов в текущей директории, включая скрытые, с детальной информацией	ПК-3
15.	Как создать новый коммит в Git с сообщением fix bug?	git commit -m "fix bug"	ПК-3
16.	Как добавить файл в индекс перед коммитом?	git add имя_файла или git add . для всех изменений	ПК-3
17.	Как запустить Docker-контейнер в фоновом режиме?	docker run -d	ПК-6
18.	Что такое Docker-образ?	Шаблон, используемый для создания контейнера; содержит файловую систему и метаданные	ПК-6
19.	Как остановить контейнер по его идентификатору?	docker stop идентификатор	ПК-6
20.	Для чего используется файл docker-compose.yml?	Для описания и запуска многоконтейнерных приложений; определяет сервисы, сети, тома и их конфигурацию	ПК-6