
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол № 2

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2025

**Москва
2025**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	7
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

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

Изучение дисциплины (модуля) «Инструменты разработчика» помогает студентам овладеть современными инструментами разработчика, позволяет повысить качество кода, ускорить процесс разработки и обеспечить удобное сотрудничество в команде. Это способствует созданию надежных и масштабируемых программных продуктов, соответствующих требованиям рынка.

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 Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование практических навыков использования современных инструментов и технологий для эффективной разработки, контроля версий, сетевого взаимодействия и отладки программного обеспечения.

Задачи изучения дисциплины (модуля):

- изучить архитектуру и основные компоненты операционной системы Linux для понимания её устройства и возможностей;
- освоить работу с командной строкой bash для выполнения повседневных задач и автоматизации процессов;
- освоить принципы работы системы контроля версий Git и научиться управлять репозиториями в удалённых сервисах;
- изучить виды сетевого взаимодействия и протокол HTTP для диагностики и решения проблем клиент-серверного взаимодействия;
- развить навыки написания чистого и поддерживаемого программного кода с учётом практик отладки и тестирования.

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.

уметь:

- работать с командной строкой bash в Linux;
- использовать систему контроля версий Git;
- диагностировать и решать проблемы клиент-серверного взаимодействия веб-приложений.

владеть:

- навыками подключения к удалённым серверам и работы с ними;
- навыками работы с удалёнными Git-репозиториями в GitHub, GitLab и других системах;
- навыками отладки клиент-серверного взаимодействия приложений;

— навыками 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-2.	Способен создавать и исследовать новые математические модели в естественных науках, совершенствовать и разрабатывать концепции, теории и методы	ОПК-2.1.	Знает основные математические модели и методы, используемые в естественных науках, включая статистическое моделирование, дифференциальные уравнения и численные методы, а также современные подходы к исследованию и анализу данных
		ОПК-2.2.	Умеет разрабатывать и адаптировать математические модели для решения конкретных проблем в естественных науках, проводить их анализ и верификацию, а также интерпретировать полученные результаты в контексте научных исследований
		ОПК-2.3.	Имеет практический опыт создания и исследования математических моделей в рамках научных проектов или исследований, включая участие в публикациях, конференциях или коллаборациях, где были разработаны и апробированы новые концепции и методы
ПК-1.	Способен определять общие формы и закономерности области разработки	ПК-1.1.	Знает основные теоретические концепции и принципы, относящиеся к области разработки, 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 разработки, 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

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контр оль	Самосто ятельна я работа	
Лекции	Семинары (Практическ ие занятия)					
1	Система контроля версий Git	4	4		15	Домашнее задание
2	Основы ОС Linux, работы с командной строкой и языком bash	6	6		25	Домашнее задание
3	Система контроля версий Git	2	2		9	Домашнее задание
4	Работа с удаленным сервером по SSH	2	2		9	Домашнее задание
5	Системы сборки проектов на основе программного кода	2	2		9	Домашнее задание
6	Основы сетевого взаимодействия, протокол HTTP	6	6		25	Домашнее задание
7	Принципы написания программного кода	2	2		9	Домашнее задание
8	Контейнеризация и Docker	6	6		25	Домашнее задание
	<i>Зачет с оценкой</i>			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	Система контроля версий Git	Введение. Репозитории, коммиты, ветки, состояния файлов Удалённые репозитории и синхронизация с ними
2	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
3	Система контроля версий Git	Отмена изменений. Стратегии слияния. Git Workflows
4	Работа с удаленным сервером по SSH	Удалённое подключение к серверам. SSH, SCP, FTP, SFTP. Работа через VNC
5	Системы сборки проектов на основе программного кода	Виды систем сборки. Механизмы работы. Сборка Java-проектов. Maven и Gradle
6	Основы сетевого взаимодействия, протокол HTTP	Сетевой стек ISO/OSI и TCP/IP. Сокеты. Маршрутизация. Работа с DNS Протокол HTTP. REST и RESTful API. Аутентификация и авторизация. Отладка API Принципы работы веб-серверов. Настройка Nginx. Запуск Java-приложений на сервере

7	Принципы написания программного кода	Паттерны проектирования. Антипаттерны. Инструменты анализа кода
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

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>.

Дополнительная литература:

1. Таненбаум, Э. С. Компьютерные сети / Э. С. Таненбаум, Д. Уэзеролл. - 5-е изд. - Санкт-Петербург : Питер, 2021. - 960 с. - ISBN 978-5-4461-1248-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2141427>.

2. Керриск, М. Linux API. Исчерпывающее руководство : практическое руководство / М. Керриск. - Санкт-Петербург : Питер, 2018. - 1248 с. - (Серия «Библиотека программиста»). - ISBN 978-5-496-02689-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1760842>.

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	Катастрофы, стихийные бедствия, аварии, эпидемии. Солнечная и геомагнитная активность. /ежедневный обзор	http://www.disasters.chat.ru
2	Каталог по безопасности жизнедеятельности	http://www.eun.chat.ru
3	Научная электронная библиотека eLibrary.ru библиотека	https://elibrary.ru/defaultx.asp
4	База данных для IT-специалистов	https://habr.com
5	База данных ScienceDirect	https://www.sciencedirect.com
6	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
7	Федеральный портал «Российское образование»	https://www.edu.ru/
8	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
9	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
10	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/
11	Сайт различных плагинов	https://maven.apache.org/plugin/s/
12	Maven central repository - хранилище библиотек и фреймворков	https://mvnrepository.com/repos/central

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
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 заданий 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) «Инструменты разработчика».

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

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (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 практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать,
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Инструменты разработчика» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	70%	8 домашних заданий, которые оцениваются по критериям. Максимально можно набрать 10 баллов за каждое из заданий.
Зачет с оценкой	30%	Контрольная работа, на которой оценивается процент правильных ответов и конвертируется в количество набранных баллов (так, за 100% правильных ответов студент получает 10 баллов).

Формула расчёта итоговой оценки по дисциплине (модулю) «Инструменты разработчика»: $\langle 0,7 \times \text{среднее за домашние задания} + 0,3 \times \text{зачет с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

Необходимо написать аналог утилиты **du** с опцией **-bs**

Предполагается, что утилита всегда работает с файлами и директориями, в поддиректориях нет ссылок на другие файлы, все файлы при этом могут быть прочитаны пользователем.

Можно пользоваться только командами `stat` (или `ls -l`) и `awk`

Usage:

`./du.sh <path to file or directory>`

Процедура сдачи задания

1. Создайте приватный репозиторий
2. Добавьте akhtyamov Pavel в коллабораторы репозитория (желательно это делать на gitlab.akhcheck.ru)
3. Добавьте пользователей в репозиторий:
 - o techprogchecker - для GitHub
 - o checker - для gitlab.akhcheck.ru
4. Создайте ветку tasks-bash
5. В ветке tasks-bash создайте папку tasks-bash
6. В папке tasks-bash создайте папку du
7. В папке du создайте скрипт run.sh, который необходим для сдачи задания

Домашнее задание

Напишите скрипт, который позволяет разбить датасет в формате csv на train и val выборку.

Можно протестировать работу на датасете.

Принимаемые параметры

- `--input` ... (путь к датасету)
- `--train_ratio` ... (доля объектов в обучающей выборке - от 0 до 100)
- `--shuffle` (Флаг есть, если необходимо перемешать датасет)
- `--train_file` ... (путь к train данным)
- `--val_file` ... (путь к val данным)

Процедура сдачи задания

1. Создайте приватный репозиторий
2. Добавьте akhtyamov Pavel в коллабораторы репозитория (желательно это делать на gitlab.akhcheck.ru)
3. Добавьте пользователей в репозиторий:
 - o techprogchecker - для GitHub
 - o checker - для gitlab.akhcheck.ru
4. Создайте ветку tasks-bash
5. В ветке tasks-bash создайте папку tasks-bash
6. В папке tasks-bash создайте папку train-val-split

Домашнее задание

Описание задания

Вам предложен готовый проект на java, необходимо научиться его собирать с помощью maven.

Общие требования к выполнению задания:

- Вносить правки в код и скрипты запрещено
- Необходимо указать, что для сборки необходимо использовать 11 версию языка Java для компилятора

- Укажите формат кодировки: **UTF-8**

Алгоритм действий

1. Склонировать себе проект
 - Его можно найти [здесь](#)
2. Написать `pom.xml` для подпроекта `easterPreparation`
 - Необходимо осуществить компиляцию файлов и запуск тестовПроверка будет осуществляться командой `mvn test`
2. Написать сборку для всего проекта `easter`
 - Требования к подпроекту `easterPreparation` изложены в пункте 1
 - На каждый подпроект должен быть свой `pom.xml`
 - Для успешной сборки подпроекта `easterParty` требуется подпроект `easterPreparation`, а также файл `attenders.json`, который генерируется скриптом `create_people.sh`
 - Скрипт должен быть вызван с параметрами 3 и 4
 - Должен существовать `pom.xml` на весь проект, лежащий в директории `easter`
 - В результате сборки подпроекта `easterPreparation` должен получиться `jar`-архив. Его следует положить в папку `task-maven/jar/`, необходимые для его работы зависимости следует положить рядом с ним в отдельную папку, назвав ее `libs`
 - Подпроект `easterParty` должен иметь `artifactId: easter-party, a version: 1.0.0`Сборка будет осуществляться с помощью команды `mvn package`, запуск `jar java -jar [jar-name]`
3. Доработать сборку
 - Осуществить автоматическую генерацию `javadoc`-а: он за вас уже написан, поэтому настройте автоматическую генерацию сайта по существующим описаниям. Не забудьте собрать единый джава док из всех модулей! Переносить в другую директорию не нужно: все должно остаться в `task-maven/target/site/`
 - Настроить очистку проекта: необходимо почистить все папки для сборки, папку `jar` и файл `attenders.json`Данный этап будет проверяться следующей последовательностью команд: `mvn install -> mvn javadoc:aggregate -> mvn clean`

Критерии оценивания

За первый пункт дается 3 балла, за второй - 4, за последний - оставшиеся 3 балла.

Обратите внимание, что для выполнения какого-либо пункта необходимо перед этим закрыть предыдущие.

Процедура сдачи задания

1. Создайте в репозитории ветку `task-maven`
2. Скопируйте проект в ветку `task-maven`. В ветке `task-maven` должна быть папка `task-maven`. В папке `task-maven` должно быть содержимое проекта.
3. Создайте `merge request` из ветки `task-maven` в ветку `master/main`, добавьте ревьюера в `merge request` и не сливайте этот `merge request`!

Примерные задания по контрольной работе

Контрольная работа по системам контроля версий и Git

Задание 1: Виды систем контроля версий

1. Опишите основные виды систем контроля версий. В чем их отличия и преимущества?

Задание 2: Базовые команды в Git

2. Объясните, что делает команда `git commit`. Какова её основная цель и какие параметры можно использовать?

Задание 3: Работа с удаленным репозиторием

3. Как вы можете подключиться к удаленному репозиторию в Git? Приведите пример команды.

Задание 4: Команды для работы с ветками

4. Каковы основные отличия между командами `git checkout` и `git switch`? Когда следует использовать каждую из них?

Задание 5: Слияние и ребейз

5. Объясните разницу между `merge` и `rebase`. В каких случаях предпочтительнее использовать каждую из этих операций?

Задание 6: Контроль за процессами слияния

6. Как вы можете разрешить конфликты, возникающие при слиянии веток в Git? Опишите шаги, которые необходимо предпринять.

Задание 7: Git Workflows

7. Опишите один из популярных Git Workflows (например, Git Flow или GitHub Flow). Каковы его основные этапы?

Задание 8: Git Hooks

8. Что такое Git Hooks? Приведите примеры ситуаций, в которых их можно использовать.

Задание 9: Основные команды и Shortcuts

9. Назовите и опишите три основных команды Git, которые вы используете чаще всего. Как они помогают в вашей работе?

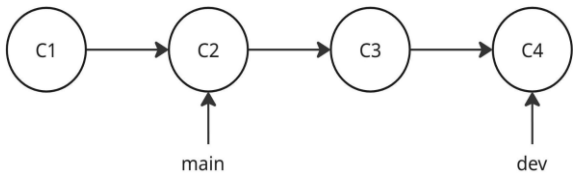
Задание 10: Сценарий использования

Представьте, что вы работаете над проектом с командой. Опишите процесс, начиная с создания новой ветки для новой функции и заканчивая слиянием этой ветки обратно в основную. Укажите, какие команды Git вы будете использовать на каждом этапе.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Из файла <code>foo.md</code> создали два объекта: <ul style="list-style-type: none">• символическую ссылку: <code>foo-soft-link.md</code>• жесткую ссылку: <code>foo-hard-link.md</code> Далее файл <code>foo.md</code> переместили в файл <code>foo2.md</code> . Укажи название файла, применение команды <code>cat</code> к которому завершиться с ошибкой.	<code>foo-soft-link.md</code>	ПК-1
2.	Какие атрибуты прав имеет файл с правами <code>0735</code> ? А. <code>rwX-wXr-X</code> Б. <code>r-xrw-rwX</code> В. <code>rwXr-xrw-</code> Г. <code>r-xrwXrw-</code>	А	ПК-3
3.	В репозитории Git воссоздали следующую структуру первой фиксации (<code>commit-ом</code>). <pre>. ├── docs │ └── TEST.md ├── README.md └── tmp.txt</pre>	2	ПК-3

	Выпишите количество объектов вида tree с их атрибутами, которые были созданы в локальном репозитории Git.		
4.	<p>На рисунке изображена система 파티ций файловой системы. Напишите название партиции с твердотельным накопителем, в котором расположены данные операционной системы, доступные для задач общего назначения</p> <pre>Filesystem Size Used Avail Use% Mounted on udev 32G 0 32G 0% /dev tmpfs 6,3G 3,0M 6,3G 1% /run /dev/nvme0n1p7 230G 187G 32G 86% / tmpfs 32G 161M 32G 1% /dev/shm tmpfs 5,0M 4,0K 5,0M 1% /run/lock tmpfs 32G 0 32G 0% /sys/fs/cgroup /dev/sdb 7,3T 6,9T 27G 100% /8tbsata /dev/nvme0n1p1 646M 73M 574M 12% /boot/efi /dev/sda5 1,5T 1,4T 48G 97% /2tbsata tmpfs 6,3G 36K 6,3G 1% /run/user/1000</pre>	/dev/nvme0n1p7	ПК-1
5.	Напишите основной метод протокола HTTP, в рамках которых тело запроса может быть проигнорировано принимающим сервером.	GET	ПК-1
6.	<p>Команда ведет разработку ЦУ. Такси в общем Git-репозитории. Посмотрите на версии тегов и отсортируйте их в порядке возрастания версий (ответ запиши в последовательности букв, например: ДВГАБ – без знаков препинания и пробелов):</p> <p>Version А) 1.10.2-beta.15 Б) 2.7.0 В) 1.10.2 Г) 1.15.1-alpha Д) 1.10.1-rc.2</p>	ДАВГБ / давгб	ПК-1
7.	<p>Какой код выведет скрипт ./01-script.sh в случае запуска при помощи следующих аргументов: перед запуском скрипта была запущена команда export a=10</p> <pre>#!/bin/bash echo \$a sleep 2 a=1 ./01-script.sh</pre> <p>А. Последовательность из 10. Б. Последовательность из 1 В. Пустой вывод. Г. 10 и последовательность из 1.</p>	Г	ПК-1
8.	Укажите название протокола, при помощи которого происходит отправка в удаленный репозиторий без запроса пароля.	SSH/ssh	ПК-3
9.	Разработчик написал однострочный скрипт, который позволит найти все изображения расширения .png, находящиеся в папке /home/cu	-type,-name/ name,-type	ОПК-2

	Укажите опции команды <code>find</code> , которые использовал разработчик. Перечислите команды через запятую.		
10.	<p>Укажите размер директории со следующей структурой:</p> <pre> . ├── docs │ └── TEST.md ├── README.md └── tmp.txt </pre> <p>Каждый inode вида директории составляет 4 КБ, каждый inode вида файл составляет 2 КБ.</p> <p>Ответ укажите в КБ.</p>	14	ПК-1
11.	В вашем проекте был настроен Git Flow. Однако в удаленном репозитории <code>origin</code> на каждый Merge в Main/Master ветку не был найден тег <code>v1.0</code>	<code>git push origin v1.0</code>	ОПК-2
12.	Назовите стадию сборки проекта, в котором один или несколько объектных файлов собираются в один общий бинарных файлов.	Линковка / линковка	ПК-3
13.	Укажите уровни протокола ISO/OSI, которые переносятся на прикладной уровень TCP/IP.	567	ПК-3
	Перечислите ответы через последовательность индексов, написанных слитно.		
14.	Укажите опцию команды <code>parallel</code> или <code>make</code> , которая позволит указать количество опцию сборки.	<code>-j</code>	ОПК-2
15.	<p>В системе контроля версий изображена следующая диаграмма:</p>  <pre> graph LR C1((C1)) --> C2((C2)) C2 --> C3((C3)) C3 --> C4((C4)) main[main] --> C2 dev[dev] --> C4 </pre> <p>В ветке <code>main</code> выполнили команду <code>git merge dev</code></p> <p>Укажите количество новых созданных коммитов в репозитории.</p>	0	ОПК-2
16.	<p>Перечислите сокращения типов для:</p> <ul style="list-style-type: none"> ● регулярного файла ● директории ● символической ссылки ● сокета <p>Запишите в виде строки из 4 символов.</p>	<code>-dls</code>	ОПК-2