
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: Прикладной искусственный интеллект

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	14
4. Содержание дисциплины (модуля)	14
5. Учебно-методическое обеспечение	15
6. Материально-техническое обеспечение	15
7. Методические и оценочные материалы	17

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 «Прикладной искусственный интеллект», 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 «Прикладной искусственный интеллект» и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре.

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:

Код и содержание компетенции	Индикатор компетенции и перечень планируемых результатов обучения по дисциплине (модулю)
УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде (соответствует УНК-02)	УК-3.1. (УНК-02.01.) Принимает участие в групповом взаимодействии в ходе профессиональной деятельности УНК-02.01.01. У-1. Проявляет понимание своей роли в команде. УНК-02.01.01. У-2. Умеет ясно выражать свои мысли в группе. УНК-02.01.01. У-3. Способен слушать и учитывать мнения других участников команды. УНК-02.01.01. У-4. Умеет давать конструктивную обратную связь. УНК-02.01.01. У-5. Умеет работать в команде, проявлять инициативу и поддерживать коллег. УНК-02.01.01. У-6. Обладает навыками разрешения конфликтных ситуаций и поиска компромиссов. УНК-02.01.01. З-1. Знает основы командной работы, роли и ответственности каждого участника. УК-3.2. (УНК-02.02.) Проявляет лидерство и осуществляет наставничество УНК-02.02.01. У-1. Умеет мотивировать и вдохновлять команду для достижения общих целей. УНК-02.02.01. У-2. Способен организовывать работу группы, распределять обязанности и контролировать выполнение задач. УНК-02.02.01. У-3. Умеет делиться знаниями и опытом, оказывать поддержку и осуществлять наставничество коллегам. УНК-02.02.01. У-4. Способен принимать ответственные решения и демонстрировать пример профессионализма. УНК-02.02.01. З-1. Знает основы лидерства, мотивации и командообразования. УНК-02.02.01. З-2. Знает принципы эффективного наставничества и развития персонала. УНК-02.02.01. З-3. Знает методы оценки эффективности работы команды и индивидуальных участников.
УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни (соответствует УНК-04)	УК-6.1. (УНК-04.01.) Планирует и осуществляет самообучение УНК-04.01.01. У-1. Умеет самостоятельно определять области для профессионального развития и формулировать цели обучения. УНК-04.01.01. У-2. Способен разрабатывать план самообучения, выбирать подходящие источники и методы обучения. УНК-04.01.01. У-3. Умеет организовывать свое время для регулярного повышения квалификации. УНК-04.01.01. У-4. Способен оценивать эффективность проведенного обучения и корректировать план при необходимости. УНК-04.01.01. З-1. Знает современные методы и ресурсы для самостоятельного обучения. УНК-04.01.01. З-2. Понимает принципы постановки целей и планирования личностного развития. УНК-04.01.01. З-3. Знает основы саморегуляции и мотивации для поддержания постоянного профессионального роста.

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 основные классы математических алгоритмов, методы их анализа, критерии выбора и оценки вычислительной сложности, а также принципы программной реализации.
	ОПК-4.2. Умеет находить и анализировать математические алгоритмы, выбирать подходящий алгоритм для поставленной задачи и реализовывать его с использованием современных вычислительных систем.
	ОПК-4.3. Имеет практический опыт программной реализации, тестирования и оценки корректности и эффективности математических алгоритмов.
ППК-Р2. Способен проверять работоспособность и проводить рефакторинг кода программного обеспечения	ППК-Р2.1. Разрабатывает тестовые наборы данных для проверки работоспособности компьютерного программного обеспечения (базовый) ППК-Р2.1. З-1. Знает методы создания и документирования контрольных примеров и тестовых наборов данных. ППК-Р2.1. З-2. Знает требования к структуре и форматам хранения тестовых наборов данных. ППК-Р2.1. З-3. Знает правила, алгоритмы и технологии создания тестовых наборов данных. ППК-Р2.1. У-1. Умеет разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения. ППК-Р2.1. У-2. Умеет готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения.
	ППК-Р2.2. Проверяет работоспособность компьютерного программного обеспечения (базовый) ППК-Р2.2. З-1. Знает методы и средства проверки работоспособности компьютерного программного обеспечения. ППК-Р2.2. З-2. Знает государственные стандарты испытания автоматизированных систем. ППК-Р2.2. З-3. Знает руководящие документы по стандартизации требований к документам автоматизированных систем. ППК-Р2.2. У-1. Умеет применять методы и средства проверки работоспособности компьютерного программного обеспечения. ППК-Р2.2. У-2. Умеет интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения. ППК-Р2.2. У-3. Умеет анализировать значения полученных характеристик компьютерного программного обеспечения.
	ППК-Р2.3. Исправляет дефекты программного кода, зафиксированные в базе данных дефектов (средний) ППК-Р2.3. З-1. Знает типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления. ППК-Р2.3. З-1. Знает методы и приемы отладки программного кода. ППК-Р2.3. У-1. Умеет воспроизводить дефекты программного кода, зафиксированные в базе данных дефектов. ППК-Р2.3. У-2. Умеет выяснять причины возникновения дефектов программного кода.

	ППК-Р2.3. У-3. Умеет вносить изменений в программный код для устранения выявленных дефектов.
	ППК-Р2.4. Выполняет рефакторинг и инспекцию программного кода (средний) ППК-Р2.4. З-1. Знает методы и средства рефакторинга и инспекции программного кода. ППК-Р2.4. З-1. Знает нормативно-технические документы (стандарты и регламенты), регламентирующие требования к программному коду, порядок отражения изменений в системе управления версиями, порядок отражения результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний. ППК-Р2.4. У-1. Умеет анализировать программный код на соответствие требованиям по читаемости и производительности. ППК-Р2.4. У-2. Умеет проводить инспекцию программного кода для поиска не обнаруженных на ранних стадиях разработки компьютерного программного обеспечения ошибок и критических мест. ППК-Р2.4. У-3. Умеет применять методы и средства рефакторинга и инспекции программного кода. ППК-Р2.4. У-4. Умеет публиковать результаты рефакторинга и инспекции в коллективной базе знаний.
ППК-Р6. Способен участвовать в промышленной разработке программного обеспечения	ППК-Р6.1. Работает в соответствии с промышленными методологиями разработки (средний) ППК-Р6.1. З-1. Знает принципы Agile и их применение в промышленных проектах. ППК-Р6.1. З-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership). ППК-Р6.1. У-1. Умеет оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов. ППК-Р6.1. У-2. Умеет работать в команде с использованием инструментов управления проектами.
	ППК-Р6.2. Использует инструменты промышленной разработки (средний) ППК-Р6.2. З-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD). ППК-Р6.2. З-2. Знает системы мониторинга и логирования в продуктивной среде. ППК-Р6.2. У-1. Умеет настраивать потоки работ CI/CD. ППК-Р6.2. У-2. Умеет работать с контейнеризацией и оркестрацией. ППК-Р6.2. У-2. Умеет настраивать мониторинг в продуктивной среде.
	ППК-Р6.3. Разрабатывает масштабируемый и поддерживаемый код (продвинутый) ППК-Р6.3. З-1. Знает принципы чистого кода, SOLID, DRY, KISS и др. ППК-Р6.3. З-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения. ППК-Р6.3. З-3. Знает паттерны проектирования и антипаттерны. ППК-Р6.3. У-1. Умеет разрабатывать модульный и тестируемый программный код. ППК-Р6.3. У-2. Умеет выполнять модульное, интеграционное и нагрузочное тестирование.

	ППК-Р6.3. У-3. Умеет проводить рефакторинг для повышения качества кода. ППК-Р6.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах.
	ППК-Р6.4. Участвует в развертывании и поддержке программного обеспечения (продвинутый) ППК-Р6.4. З-1. Знает стратегии развертывания промышленного программного обеспечения. ППК-Р6.4. З-2. Знает основы работы с облачными платформами. ППК-Р6.4. У-1. Умеет развертывать приложения в облаке. ППК-Р6.4. У-2. Умеет обнаруживать и устранять инциденты с работой продуктивного программного обеспечения.
РГ-1. Способен руководить процессами разработки компьютерного программного обеспечения	РГ-1.1. Руководит разработкой программного кода (средний) РГ-1.1. З-1. Знает методы принятия управленческих решений. РГ-1.1. З-2. Знает основные принципы и методы управления персоналом. РГ-1.1. З-1. Знает технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии. РГ-1.1. У-1. Способен распределять задачи на разработку программного кода между исполнителями. РГ-1.1. У-2. Применяет лучшие мировые практики оформления программного кода. РГ-1.1. У-3. Оценивает качество и эффективность программного кода. РГ-1.2. Руководит проверкой работоспособности компьютерного программного обеспечения (средний) РГ-1.2. З-1. Знает основные методы измерения и оценки характеристик компьютерного программного обеспечения. РГ-1.2. З-2. Знает методы и средства проверки работоспособности компьютерного программного обеспечения. РГ-1.2. З-3. Знает методы и средства рефакторинга и оптимизации программного кода. РГ-1.2. З-4. Знает о возможности отказа частей системы в результате воздействий внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов). РГ-1.2. У-1. Умеет распределять задачи на проверку работоспособности компьютерного программного обеспечения между исполнителями. РГ-1.2. У-2. Умеет оценивать результаты проверки работоспособности компьютерного программного обеспечения. РГ-1.2. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности компьютерного программного обеспечения об исправлении ошибок, рефакторинге, оптимизации и инспекции кода. РГ-1.2. У-4. Умеет включать поиск возможных отказов из воздействий внутреннего или внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) в процедуры инспекции программного кода.

	РГ-1.3. Руководит интеграцией программных модулей и компонентов компьютерного программного обеспечения (продвинутый) РГ-1.3. З-1. Знает методы и средства сборки модулей и компонентов компьютерного программного обеспечения. РГ-1.3. З-2. Знает методы и программные интерфейсы взаимодействия компьютерного программного обеспечения с внешними программными компонентами. РГ-1.3. З-3. Знает методы проектирования и разработки программных интерфейсов взаимодействия внутренних модулей компьютерного программного обеспечения. РГ-1.3. З-4. Знает методы и средства разработки процедур для развертывания компьютерного программного обеспечения. РГ-1.3. У-1. Умеет назначать задания на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта. РГ-1.3. У-2. Умеет оценивать результаты выполнения назначенных заданий на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта. РГ-1.3. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности выпусков программного продукта (решение о выпуске/невыпуске версии, отправка задач на доработку, добавление новых задач, передача на тестирование).
РГ-2. Способен организовать процессы разработки компьютерного программного обеспечения	РГ-2.1. Управляет проектированием компьютерного программного обеспечения (средний) РГ-2.1. З-1. Знает принципы построения архитектуры компьютерного программного обеспечения и виды архитектуры программного обеспечения. РГ-2.1. З-2. Знает методологии и средства проектирования компьютерного программного обеспечения. РГ-2.1. У-1. Умеет анализировать архитектуру компьютерного программного обеспечения и согласовывать ее с заинтересованными сторонами. РГ-2.1. У-2. Умеет распределять задания на проектирование компьютерного программного обеспечения, структуры базы данных, программных интерфейсов. РГ-2.1. У-2. Умеет оценивать качество проектирования компьютерного программного обеспечения, структуры базы данных, программных интерфейсов.
	РГ-2.2. Управляет процессом разработки компьютерного программного обеспечения (продвинутый) РГ-2.2. З-1. Знает методы и средства планирования и контроля (мониторинга) исполнения планов. РГ-2.2. З-2. Знает методы оценки качества плана разработки программного продукта (ресурсы, сроки, риски). РГ-2.2. З-3. Знает основные принципы и методы управления персоналом. РГ-2.2. У-1. Умеет составлять планы процесса разработки программного продукта. РГ-2.2. У-2. Умеет оценивать качество плана разработки программного продукта (ресурсы, сроки, риски). РГ-2.2. У-3. Умеет наблюдать за исполнением планов разработки программного продукта.

	РГ-2.2. У-4. Умеет корректировать план разработки программного продукта.
	РГ-2.3. Управляет запросами на изменения, дефектами и проблемами в компьютерном программном обеспечении (продвинутый) РГ-2.3. 3-1. Знает методы и средства выявления дефектов, проблем и причин их возникновения в компьютерном программном обеспечении. РГ-2.3. 3-2. Знает методы и средства управления запросами на изменения в компьютерном программном обеспечении. РГ-2.3. 3-3. Знает методы планирования и документирования вносимых изменений в компьютерное программное обеспечение. РГ-2.3. У-1. Умеет оценивать запросы на изменения и предложенных решений по их осуществлению (по стоимости, трудоемкости, эффективности) в компьютерном программном обеспечении. РГ-2.3. У-2. Умеет принимать управленческие решения о реализации запросов на изменения (решений о необходимости и сроках внесения изменений в программное обеспечение и документацию). РГ-2.3. У-3. Умеет контролировать исполнение принятых управленческих решений.
	РГ-2.4. Управляет конфигурациями и выпусками программного продукта (продвинутый) РГ-2.3. 3-1. Знает нормативно-технические документы (стандарты и регламенты) по процессам управления конфигурациями, изменениями и выпусками программного продукта. РГ-2.3. 3-2. Знает состав и методы использования коллективной среды разработки компьютерного программного обеспечения и системы контроля версий. РГ-2.3. 3-3. Знает методы и средства верификации работоспособности выпусков программных продуктов. РГ-2.5. У-1. Умеет управлять версиями отдельных компонентов и программного продукта в целом. РГ-2.5. У-2. Умеет анализировать требования к выпуску новой версии программного продукта. РГ-2.5. У-3. Умеет определять перечень функциональных требований, реализуемых в новой версии программного продукта. РГ-2.5. У-3. Умеет контролировать выполнение разработки версии программного продукта.
РГ-4. Способен организовать промышленную разработку программного обеспечения	РГ-4.1. Внедряет промышленные методологии разработки (средний) РГ-4.1. 3-1. Знает принципы Agile и их применение в промышленных проектах. РГ-4.1. 3-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership). РГ-4.1. У-1. Умеет оценивать объем задач и срок их выполнения. РГ-4.1. У-2. Умеет распределять задачи между участниками команды. РГ-4.1. У-3. Умеет организовать работу в команде с использованием инструментов управления проектами.

	РГ-4.2. Внедряет инструменты промышленной разработки (средний) РГ-4.2. 3-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD). РГ-4.2. 3-2. Знает системы мониторинга и логирования в продуктивной среде. РГ-4.2. У-1. Умеет управлять настройкой потоков работ CI/CD. РГ-4.2. У-2. Умеет управлять работой с контейнеризацией и оркестрацией. РГ-4.2. У-3. Умеет организовать мониторинг в продуктивной среде.
	РГ-4.3. Организует разработку масштабируемого и поддерживаемого кода (продвинутой) РГ-4.3. 3-1. Знает принципы чистого кода, SOLID, DRY, KISS и др. РГ-4.3. 3-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения. РГ-4.3. 3-3. Знает паттерны проектирования и антипаттерны. РГ-4.3. У-1. Умеет управлять разработкой модульного и тестируемого программного кода. РГ-4.3. У-2. Умеет организовать выполнение модульного, интеграционного и нагрузочного тестирования. РГ-4.3. У-3. Умеет управлять техническим долгом, в том числе организовать рефакторинг для повышения качества кода. РГ-4.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения.
	РГ-4.4. Организует развертывание и поддержку программного обеспечения (продвинутой) РГ-4.4. 3-1. Знает стратегии развертывания промышленного программного обеспечения. РГ-4.4. 3-2. Знает основы работы с облачными платформами. РГ-4.4. У-1. Умеет организовать развертывание приложения в облаке. РГ-4.4. У-2. Умеет управлять обнаружением и устранением инцидентов с работой продуктивного программного обеспечения.
РГ-5. Способен организовать процессы применения искусственного интеллекта (ИИ) для генерации и отладки программного кода	РГ-5.1. Организует применение ИИ-инструментов для генерации программного кода (средний) РГ-5.1. 3-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода. РГ-5.1. 3-2. Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование). РГ-5.1. У-1. Умеет выбирать подходящие под задачи ИИ-инструменты. РГ-5.1. У-2. Умеет организовывать процесс применения ИИ для генерации программного кода.
	РГ-5.2. Организует применение ИИ для анализа и отладки кода (продвинутой) РГ-5.2. 3-1. Знает методы ИИ-анализа кода. РГ-5.2. 3-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ. РГ-5.2. У-1. Умеет организовать применение ИИ-инструментов для поиска ошибок и недостатков в коде. РГ-5.2. У-2. Умеет организовать применение ИИ для

	исправлению кода. РГ-5.2. У-3. Умеет организовать применение ИИ для проведения code review.
	РГ-5.3. Организует оптимизацию кода с помощью ИИ (продвинутый) РГ-5.3. 3-1. Знает методы ИИ-оптимизации. РГ-5.3. 3-2. Знает критерии качества кода, применяемые ИИ-системами. РГ-5.3. У-1. Умеет организовать проведение рефакторинга с использованием ИИ. РГ-5.3. У-2. Умеет управлять проверкой корректности оптимизаций, предложенных ИИ.
	РГ-5.4. Оценивает этические и профессиональные аспекты применения ИИ в разработке (продвинутый) РГ-5.4. 3-1. Знает этические нормы использования ИИ (конфиденциальность, плагиат кода и т.п.). РГ-5.4. 3-2. Знает лицензионные ограничения сгенерированного кода. РГ-5.4. У-1. Умеет организовать проверку кода на соответствие стандартам после ИИ-генерации. РГ-5.4. У-2. Умеет организовать процесс документирования использования ИИ в разработке.
	РГ-5.5. Организует внедрение ИИ в разработку (продвинутый) РГ-5.5. 3-1. Знает методы внедрения ИИ в разработку. РГ-5.5. 3-2. Знает критерии качества внедрения ИИ в разработку. РГ-5.5. У-1. Умеет организовать внедрение ИИ в разработку. РГ-5.5. У-2. Умеет управлять процессом внедрения ИИ в разработку.
ППК-У1. Осуществляет оценку и управление рисками	ППК-У1.1. Осуществляет идентификацию рисков в проекте (средний) ППК-У1.1. У-1. Способен выявлять контекст рисков, их идентификацию и формировать портфель рисков проекта. ППК-У1.1. У-2. Умеет осуществлять мониторинг рисков проекта. ППК-У1.1. У-3. Умеет проводить качественную и количественную статистическую оценку рисков на основе фактических событий базы рисков событий. ППК-У1.1. У-4. Способен осуществлять оценку рисков ситуаций в динамике, тестирование и верификацию методик идентификации рисков с учетом отраслевой специфики и контекста функционирования организации. ППК-У1.1. 3-1. Знает критерии, методы, правила идентификации риска. ППК-У1.1. 3-2. Знает возможности инструментов риск-менеджмента для идентификации рисков организации. ППК-У1.1. 3-3. Знает законодательство Российской Федерации, базовые положения международных стандартов и отраслевые стандарты по управлению рискам. ППК-У1.2. Осуществляет сбор и обработку релевантной аналитической информации для анализа и оценки рисков (средний) ППК-У1.2. У-1. Способен собирать, регистрировать, обрабатывать и систематизировать релевантную информацию для проведения анализа и оценки рисков. ППК-У1.2. У-2. Умеет анализировать и идентифицировать изменения рисков в динамике. ППК-У1.2. 3-1. Знает контекст процесса управления рисками. ППК-У1.2. 3-2. Знает методы, техники, технологии, программные средства и информационные базы для идентификации различных видов риска. ППК-У1.2. 3-3. Знает законодательство Российской Федерации,

	базовые положения международных стандартов и отраслевые стандарты по управлению рискам.
	ППК-У1.3. Разрабатывает комплекс аналитических процедур и методов анализа и оценки рисков с позиции их идентификации по функциональным областям (продвинутый) ППК-У1.3. У-1. Способен осуществлять мониторинг, анализировать и оценивать риски с позиции их идентификации по функциональным областям. ППК-У1.3. У-2. Умеет определять и осуществлять отбор эффективных методов анализа и оценки рисков с позиции их идентификации по функциональным областям. ППК-У1.3. З-1. Знает критерии, методы анализа и оценки рисков с позиции их идентификации по функциональным областям. ППК-У1.3. З-2. Знает методы, техники, технологии, программные средства и информационные базы идентификации различных видов риска.
	ППК-У1.4. Осуществляет оценку уровня (пороговых значений, условных зон) рисков в разрезе отдельных видов (продвинутый) ППК-У1.4. У-1. Способен оценить вероятность события (угроз), пороговые значения (условные зоны), и предельно допустимый уровень рисков с определением индикаторов. ППК-У1.4. У-2. Способен осуществлять мониторинг пороговых значений рисков в разрезе отдельных видов. ППК-У1.4. У-3. Умеет определять критериальные показатели, для которых устанавливаются пороговые значения, в соответствии с внутренней и внешней средой (контекстом) функционирования организации, а также учитывать особые обстоятельства и ограничения. ППК-У1.4. У-4. Умеет осуществлять оценку рискованных ситуаций, тестировать и верифицировать методики оценки уровня (пороговых значений, условных зон) рисков в разрезе отдельных видов. ППК-У1.4. З-1. Знает критерии, применяемые при оценке уровня (пороговых значений, условных зон) рисков в разрезе отдельных видов. ППК-У1.4. З-2. Знает методы, техники, технологии, программные средства и информационные базы идентификации различных видов риска. ППК-У1.4. З-3. Знает законодательство Российской Федерации, базовые положения международных стандартов и отраслевые стандарты по управлению рискам.
	ППК-У1.5. Участвует в выработке мероприятий по воздействию на риск (продвинутый) ППК-У1.5. У-1. Способен разрабатывать мероприятия по управлению рисками, осуществлять мониторинг и оценку их эффективности совместно с ответственными за риск сотрудниками организации — владельцами риска. ППК-У1.5. У-2. Умеет определять и отбирать эффективные методы воздействия на риск, разрабатывать и внедрять планы воздействия на риски (совместно с ответственными за риск сотрудниками — владельцами риска), оказывать помощь ответственным за риск сотрудникам в правильной оценке риска и разработке мероприятий по их управлению. ППК-У1.5. У-3. Умеет осуществлять расчеты, прогнозировать,

	тестировать и верифицировать методики управления рисками с учетом отраслевой специфики. ППК-У1.5. У-4. Умеет формировать формы отчетности, дорожные карты для целей реализации и мониторинга мероприятий по воздействию на риски. ППК-У1.5. 3-1. Знает критерии, методы и инструменты воздействия на риски в разрезе отдельных их видов. ППК-У1.5. 3-2. Знает план мероприятий по управлению рисками. ППК-У1.5. 3-3. Знает виды, методы, формы и инструменты внутреннего контроля. ППК-У1.5. 3-4. Знает законодательство Российской Федерации, базовые положения международных стандартов и отраслевые стандарты по управлению рискам.
--	--

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 пушиться 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	Индикатор
1.	В разделе «Основы ОС Linux» укажите команду для отображения содержимого директории.	ls	УК-3	УК-3.1-3-1

2.	При групповом взаимодействии укажите инструмент для совместной работы с кодом.	Git / GitHub	УК-3	УК-3.1-У-1
3.	В командной работе укажите способ выражения мыслей при code review.	Комментарии в MR	УК-3	УК-3.1-У-2
4.	При разрешении конфликтов укажите метод поиска компромиссов в Git.	Обсуждение / Merge	УК-3	УК-3.1-У-6
5.	В разделе «Git» укажите роль участника команды при работе с репозиторием.	Разработчик / Ревьюер	УК-3	УК-3.2-3-1
6.	При лидерстве укажите способ мотивации команды для достижения целей.	Постановка задач	УК-3	УК-3.2-У-1
7.	В разделе «Самообучение» укажите источник для изучения Docker.	Официальная документация	УК-6	УК-6.1-3-1
8.	При планировании укажите метод организации времени для повышения квалификации.	Учебный план	УК-6	УК-6.1-У-2
9.	В разделе «Linux» укажите символ разрешения на выполнение файла.	x	ОПК-4	ОПК-4.1-3-1
10.	При анализе укажите метод оценки сложности bash-скриптов.	Profiling / time	ОПК-4	ОПК-4.2-У-1
11.	В разделе «Linux» укажите язык для автоматизации задач в командной строке.	Bash / Shell	ОПК-4	ОПК-4.3-У-1
12.	При тестировании укажите метод проверки корректности bash-скрипта.	Unit testing	ОПК-4	ОПК-4.3-У-2
13.	В разделе «Git» укажите команду для отправки изменений в удалённый репозиторий.	git push	ППК-Р2	ППК-Р2.1-У-1
14.	При разработке тестов укажите формат хранения тестовых данных.	JSON / YAML	ППК-Р2	ППК-Р2.1-3-2
15.	В разделе «CI/CD» укажите методологию тестирования для проверки работоспособности.	Integration testing	ППК-Р2	ППК-Р2.1-У-2
16.	При проверке работоспособности укажите средство автоматизации тестов.	CI Pipeline	ППК-Р2	ППК-Р2.2-У-1
17.	В разделе «CI/CD» укажите формат логов для анализа ошибок.	JSON logs	ППК-Р2	ППК-Р2.2-У-2
18.	При анализе характеристик укажите метрику для оценки производительности.	Время выполнения	ППК-Р2	ППК-Р2.2-У-3
19.	В разделе «Docker» укажите типичную ошибку при сборке образа.	Dependency error	ППК-Р2	ППК-Р2.3-3-1

20.	При диагностике укажите метод воспроизведения ошибки в контейнере.	docker logs	ППК-P2	ППК-P2.3-У-1
21.	В разделе «Docker» укажите подход к внесению изменений в Dockerfile.	Исправление слоя	ППК-P2	ППК-P2.3-У-3
22.	При рефакторинге укажите метод анализа кода на читаемость.	Code review	ППК-P2	ППК-P2.4-У-1
23.	В разделе «Linux» укажите инструмент для поиска ошибок в bash-скриптах.	shellcheck	ППК-P2	ППК-P2.4-У-2
24.	При оптимизации укажите подход к уменьшению размера образа Docker.	Multi-stage build	ППК-P2	ППК-P2.4-У-3
25.	В разделе «CI/CD» укажите принцип Agile для промышленных проектов.	Непрерывная интеграция	ППК-P6	ППК-P6.1-3-1
26.	При code review укажите принцип коллективного владения кодом.	Collective ownership	ППК-P6	ППК-P6.1-3-2
27.	В разделе «CI/CD» укажите метод оценки объема задачи в pipeline.	Story points	ППК-P6	ППК-P6.1-У-1
28.	При работе с инструментами укажите систему управления проектами.	Jira	ППК-P6	ППК-P6.1-У-2
29.	В разделе «Docker» укажите принцип Continuous Integration.	Автоматизация сборки	ППК-P6	ППК-P6.2-3-1
30.	При мониторинге укажите систему для логирования в продуктивной среде.	ELK Stack	ППК-P6	ППК-P6.2-3-2
31.	В разделе «CI/CD» укажите инструмент для настройки потоков работ.	GitLab CI	ППК-P6	ППК-P6.2-У-1
32.	В разделе «Linux» укажите принцип чистого кода для скриптов.	DRY / KISS	ППК-P6	ППК-P6.3-3-1
33.	При проектировании укажите подход к структурированию скриптов.	Модульность	ППК-P6	ППК-P6.3-3-2
34.	В разделе «Docker» укажите антипаттерн при написании Dockerfile.	Hardcoded paths	ППК-P6	ППК-P6.3-3-3
35.	При разработке укажите подход к тестируемости bash-скриптов.	Функции	ППК-P6	ППК-P6.3-У-1
36.	В разделе «Remote» укажите стратегию развертывания промышленного ПО.	Blue-green	ППК-P6	ППК-P6.4-3-1
37.	При облаках укажите провайдера для развертывания приложений.	AWS / Yandex Cloud	ППК-P6	ППК-P6.4-3-2
38.	В разделе «CI/CD» укажите метод деплоя приложения через pipeline.	Automated deployment	ППК-P6	ППК-P6.4-У-1

39.	При инцидентах укажите подход к обнаружению проблем в продуктиве.	Мониторинг	ППК-Р6	ППК-Р6.4-У-2
40.	В разделе «Git» укажите метод принятия решений при конфликтах слияния.	Code review	РГ-1	РГ-1.1-3-1
41.	При управлении укажите принцип распределения задач между исполнителями.	По компетенциям	РГ-1	РГ-1.1-У-1
42.	В разделе «CI/CD» укажите метрику для эффективности кода в pipeline.	Coverage	РГ-1	РГ-1.1-У-3
43.	При проверке укажите метод оценки результатов тестирования.	Отчеты CI	РГ-1	РГ-1.2-У-2
44.	В разделе «CI/CD» укажите метод сборки модулей и компонентов.	CI Pipeline	РГ-1	РГ-1.3-3-1
45.	В разделе «Linux» укажите принцип построения архитектуры ПО.	Модульность	РГ-2	РГ-2.1-3-1
46.	При проектировании укажите методологию для проектирования ПО.	Agile / DevOps	РГ-2	РГ-2.1-3-2
47.	В разделе «CI/CD» укажите метод контроля исполнения планов.	Спринты	РГ-2	РГ-2.2-3-1
48.	В разделе «Docker» укажите принцип работы генеративных ИИ-моделей.	LLM	РГ-5	РГ-5.1-3-1
49.	При рисках укажите метод идентификации рисков в проекте.	Risk assessment	ППК-У1	ППК-У1.1-У-1