

ФГБОУ ВО «Казанский национальный исследовательский технологический университет» (ФГБОУ ВО «КНИТУ»)

Институт управления, автоматизации и информационных технологий

**ОСНОВНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ВЫСШЕГО ОБРАЗОВАНИЯ ТОП-УРОВНЯ**

Наименование: **Разработка программного обеспечения и DevOps (ТОП ИТ)**

Уровень: Бакалавриат

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль): Разработка программного обеспечения и DevOps (ТОП ИТ)

Квалификация: Бакалавр

Форма обучения: Очная

Срок получения образования: 4 года

Объем программы: 240 зачетных единиц

Якорный индустриальный партнер: ООО «ГК «Иннотех» (ИНН/КПП
9703073496/771401001, ОГРН 1227700086460)

Согласовано с индустриальным партнером:

Зам. директора департамента корпоративного обучения и развития талантов –
руководитель управления, разработки образовательных
проектов и операционных процессов обучения

_____/Дядюра С.С./

«28» июля 2026 г.



**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 00D28DBE397EF6634C5A6372E2F0F1EE0F
Владелец: Казаков Юрий Михайлович
Действителен: с 16.01.2026 до 11.04.2027

Казань, 2026

Содержание

1. Общая характеристика образовательной программы	3
2. Объем и структура образовательной программы	3
3. Нормативные акты	4
4. Проектирование логики формирования компетенций	4
4.1. Планируемые результаты освоения образовательной программы	4
4.2. Логика последовательности изучения дисциплин	9
5. Взаимодействие с индустрией	26
5.1. Договоренности с партнерами:	26
5.2. Заказ на программу: целевой образ выпускника и требования индустрии	26
5.3. Кейсы от индустриального партнера	27
5.4. Практическая подготовка	31
5.5. Система стажировок как механизм углубления практической подготовки	35
6. Внутренний отбор студентов	38
7. Государственная итоговая аттестация	38
8. Подходы к обучению и используемые инструменты	40
Перечень Приложений	46

КОНЦЕПЦИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ТОП-УРОВНЯ

1. Общая характеристика образовательной программы

Цель реализации программы:

Подготовка конкурентоспособных, готовых к немедленному включению в индустриальную разработку специалистов в области DevOps-инженерии и промышленной разработки программного обеспечения, владеющих полным стеком технологий и компетенций для автоматизации жизненного цикла ИТ-продуктов, управления инфраструктурой, обеспечения надёжности и безопасности сервисов, а также обладающих потенциалом организационного и технического лидерства для роста до руководителя группы разработки в условиях цифровой трансформации крупных предприятий.

Задачи реализации образовательной программы топ-уровня:

1. Сформировать фундамент инженерных знаний и навыков в области программирования, алгоритмизации, архитектуры ЭВМ, математического моделирования и систем хранения данных.

2. Сформировать профессиональные компетенции инженера DevOps в области автоматизации жизненного цикла программного обеспечения, включая CI/CD, контейнеризацию, оркестрацию, управление инфраструктурой как кодом и облачные технологии.

3. Сформировать компетенции в области мониторинга, обеспечения надёжности, безопасности и оптимизации производительности ИТ-сервисов.

4. Сформировать управленческие компетенции для опережающего карьерного роста, включая организацию процессов разработки, управление проектами и командами, внедрение DevOps-культуры и непрерывное улучшение на основе метрик.

5. Обеспечить практическую подготовку через интеграцию в учебный процесс реальных индустриальных кейсов, практик и стажировок в партнёрских компаниях с применением системы двойного наставничества.

6. Сформировать управленческие компетенции для вертикального карьерного роста: планирование инфраструктурных изменений, приоритизация задач, управление командой разработки, технический лиднинг, взаимодействие с бизнес-заказчиками, HR-процессы.

2. Объем и структура образовательной программы

Общий объем образовательной программы топ-уровня составляет **240 зачетных единиц**.

Блок	Объем (з.е.)
Блок 1. Дисциплины (модули)	207
Блок 2. Практики	24
Блок 3. Государственная итоговая аттестация	9
Итого	240

Распределение по типам учебной работы (в часах)

Курс	Контактная работа	Лекции	Лабораторные работы	Практические работы	Контроль самостоятельной работы	Контроль	Самостоятельная работа студентов	Практика	Итоговая аттестация
1	1197	279	396	315	261	252	603	108	0
2	1152	288	531	171	216	162	630	216	0
3	995	214	617	70	146	162	715	324	0
4	873	171	585	-	117	108	675	216	324
Итого	4217	952	2129	556	740	684	2623	864	324

Структура учебного плана спроектирована с учетом количественных критериев положения о проведении экспертизы:

- Доля практических занятий (лабораторные работы, семинары, практикумы) в общем объеме контактной работы составляет более 50% (63% - к общему объему контактной работы с КСР, 77% - к общему объему контактной работы без КСР).
- Доля профильных дисциплин на 1-м курсе обучения составляет 51% от общего объема программы на 1-м курсе.

Полный учебный план с распределением дисциплин по семестрам и календарный график являются основными документами ООП. Матрица компетенций (Приложение 2) отражает распределение компетенций по дисциплинам и практикам. Полные паспорта всех профильных дисциплин с указанием пререквизитов, постреквизитов и формируемых индикаторов приведены в Приложении 3.

3. Нормативные акты

Программа разработана на основе:

1. ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии (Приказ Министерства образования и науки РФ от 19 сентября 2017 г. N 926, Редакция с изменениями N 1456 от 26.11.2020).
2. Компетентностно-ролевой модели (КРМ) в сфере информационных технологий (утверждена Оператором – АНО «Аналитический центр при Правительстве РФ»).
3. Профессиональных стандартов: 06.001 «Программист», 06.026 «Системный администратор информационно-коммуникационных систем».
4. Заказа промышленных партнёров (ООО «ГК «Иннотех», АО «Лаборатория Касперского»), зафиксированного в протоколах совместных рабочих встреч и дорожной карте взаимодействия (Приложение 5.).

4. Проектирование логики формирования компетенций

4.1. Планируемые результаты освоения образовательной программы

ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии рекомендует к включению в учебные планы, следующие обязательные универсальные и общепрофессиональные компетенции.

Универсальные компетенции выпускников и индикаторы их достижения в соответствии с ФГОС ВО

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2. Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3. Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4. Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5. Определяет и оценивает практические последствия возможных вариантов решения задачи
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1. Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
	УК-2.2. Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из

	действующих правовых норм и имеющихся ресурсов и ограничений
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1. Способен устанавливать разные виды коммуникации (учебную, научную, деловую, неформальную и др.) УК-3.2. Взаимодействует с другими членами команды для достижения поставленной задачи
УК-4 Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)	УК-4.1. Взаимодействует с другими членами команды для достижения поставленной задачи УК-4.2. Использует современные информационно-коммуникативные средства для коммуникации
УК-5 Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах	УК-5.1. Знает основные категории философии, законы исторического развития, основы межкультурной коммуникации УК-5.2. Имеет представление о системах этических и интеллектуальных ценностей и норм, их значении в истории общества
УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.1. Определяет приоритеты профессиональной деятельности и способы ее совершенствования на основе самооценки УК-6.2. Способен планировать самостоятельную деятельность в решении профессиональных задач; подвергать критическому анализу проделанную работу; находить и творчески использовать имеющийся опыт в соответствии с задачами саморазвития
УК-7 Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности	УК-7.1. Знает основы здорового образа жизни, здоровьесберегающих технологий, физической культуры УК-7.2. Понимает влияние оздоровительных систем физического воспитания на укрепление здоровья, профилактику профессиональных заболеваний УК-7.3. Понимает влияние оздоровительных систем физического воспитания на укрепление здоровья, профилактику профессиональных заболеваний
УК-8 Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов	УК-8.1. Знает классификацию и источники чрезвычайных ситуаций природного и техногенного происхождения; причины, признаки и последствия опасностей, способы защиты от чрезвычайных ситуаций УК-8.2. Умеет поддерживать безопасные условия жизнедеятельности; выявлять признаки, причины и условия возникновения чрезвычайных ситуаций; оценивать вероятность возникновения потенциальной опасности и принимать меры по ее предупреждению УК-8.3. Владеет методами прогнозирования возникновения опасных или чрезвычайных ситуаций; навыками по применению основных методов защиты в условиях чрезвычайных ситуаций
УК-9 Способен принимать обоснованные экономические решения в различных областях жизнедеятельности	УК-9.1. Знает базовые принципы функционирования экономики и экономического развития, цели и формы участия государства в экономике УК-9.2. Умеет использовать экономические знания в различных сферах деятельности, анализировать и обобщать экономическую информацию для принятия обоснованных управленческих решений УК-9.3. Владеет навыками использования методов экономического и финансового планирования для достижения финансовых целей, а также инструментами управления личными финансами и финансовыми рисками
УК-10 Способен формировать нетерпимое отношение к проявлениям экстремизма, терроризма, коррупционному поведению и противодействовать им в профессиональной деятельности	УК-10.1. Знает сущность, понятие и задачи противодействия экстремизму, терроризму и коррупции; требования законодательства в области противодействия экстремизма, терроризма и коррупции

	УК-10.2. Умеет предупреждать экстремистские, террористические и коррупционные риски в профессиональной деятельности; исключать необоснованное вмешательство в профессиональную деятельность в целях склонения к экстремистским, террористическим и коррупционным правонарушениям УК-10.3. Владеет навыками нетерпимого отношения к экстремистскому, террористическому и коррупционному поведению, уважительного отношения к праву и закону
--	--

В рамках реализации программы бакалавриата на основе компетентностно-ролевой модели (КРМ) в состав компетенций включена надпрофессиональная компетенция, предусмотренная КРМ УНК-02.

Надпрофессиональная компетенция

Универсальные надпрофессиональные компетенции из КРМ:	Индикаторы универсальных надпрофессиональных компетенций из КРМ:	Результаты обучения по индикаторам универсальных надпрофессиональных компетенций из КРМ:
УНК-02 – Способен применять методы коллективной работы в профессиональной деятельности	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.
	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

Перечень индикаторов УК-3 ориентирован на базовые аспекты социального взаимодействия: знание норм и принципов командной работы, умение выстраивать коммуникацию и обеспечивать взаимодействие в коллективе, владение навыками командной роли. В то же время индикаторы УНК-02 дополняют эту логику за счет акцента

на более сложные и вариативные проявления – лидерство, мотивацию, наставничество, принятие ответственных решений и развитие других участников команды.

Таким образом, несмотря на содержательную близость и частичное пересечение, объединение УНК-02 с УК-3 технически возможно, но методически нецелесообразно.

УК-3 фиксирует базовые навыки командного взаимодействия, тогда как УНК-02 включает более продвинутое проявления (лидерство, наставничество). Их объединение приводит к перегрузке универсальной компетенции и выходу за рамки требований ФГОС.

Кроме того, возникает дублирование индикаторов и смешение уровней сложности (базовые коммуникации и управленческие функции), что затрудняет оценивание и снижает прозрачность результатов обучения.

Сохранение УНК-02 как отдельной компетенции позволяет корректно выделить и обосновать развитие лидерских и наставнических навыков без искажения структуры УК-3.

Общепрофессиональные компетенции выпускников и индикаторы их достижения в соответствии с ФГОС ВО

В рамках УГСН выделяются общие ОПК для всех направлений подготовки и ОПК, специфичные для направления подготовки 09.03.02 «Информационные системы и технологии».

Общие ОПК укрупненной группы 09.00.00 Информатика и вычислительная техника

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1 Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1. Способен анализировать поставленную задачу, намечать пути ее решения ОПК-1.2. Способен строить математические модели, производить количественные расчеты и оценки ОПК-1.3. Способен определять границы применимости полученных результатов
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1. Обладает навыками разработки архитектуры программных систем и компонентов с учетом требований к производительности, надежности и безопасности ОПК-3.2. Умеет применять информационно-коммуникационные технологии для поиска и анализа профессиональной информации, выделения в ней главного, структурирования, оформления и представления в виде аналитических обзоров с обоснованными выводами и рекомендациями ОПК-3.3. Знает основы информационной безопасности и методы защиты программного обеспечения от угроз и атак
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1. Знает основные правила оформления технической документации на различных стадиях жизненного цикла информационной системы ОПК-4.2. Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1. Владеет на практике методологией составления научно-технических отчетов (проектов) ОПК-5.2. Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства

	использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества ОПК-5.3. Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
--	---

Специфичные ОПК для направления подготовки 09.03.02 Информационные системы и технологии

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ОПК-6.1 Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-6.2 Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-6.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
ОПК-7 Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	ОПК-7.1 Знает основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.2 Умеет применять современные технологии для реализации информационных систем ОПК-7.3 Владеет навыками применения инструментальных программно-аппаратных средств реализации информационных систем
ОПК-8 Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем	ОПК-8.1 Знает математику, методологию и основные методы математического моделирования, классификацию и условия применения моделей, методы и средства проектирования информационных и автоматизированных систем, инструментальные средства моделирования и проектирования ОПК-8.2 Умеет проводить моделирование процессов и систем с применением современных инструментальных средств ОПК-8.3 Владеет навыками моделирования и проектирования информационных и автоматизированных систем

Анализ компетенций ОПК и ПК из КРМ показал, что ППК-Р1 частично пересекается с ОПК-1, а ППК-Р3 – с ОПК-6. Их объединение методически нецелесообразно, поскольку совпадение здесь носит лишь частичный характер и не отменяет различий в уровне и назначении компетенций. В части ППК-Р1 и ОПК-1 видно, что ОПК-1 задаёт общий фундамент. В то же время ППК-Р1 раскрывает этот блок значительно более предметно и технологично: здесь речь идёт о формализации и алгоритмизации задач, переводе задач ИТ-отрасли в язык естественнонаучных дисциплин, выборе методов и алгоритмов для программной реализации математических моделей, сравнительном анализе алгоритмов и оценке их эффективности. То есть ОПК-1 фиксирует общую способность применять научные знания, а ППК-Р1 – прикладную инженерную конкретизацию этой способности под профиль программы.

Аналогично, ОПК-6 формулирует базовую общепрофессиональную способность разрабатывать алгоритмы и программы, пригодные для практического применения. ППК-Р3, в свою очередь, уточняет этот результат до важнейшего для практики направления – интеграции программных модулей и проверки работоспособности выпусков программного продукта. Её индикаторы уже не просто про написание и отладку кода, а про разработку процедур интеграции, объединение компонентов, проверку корректности взаимодействия модулей и выпусков продукта. То есть ОПК-6 описывает широкую основу программной деятельности, а ППК-Р3 – специализированный этап жизненного цикла разработки,

значимый именно для нашей программы. Поэтому, несмотря на содержательную близость, включение индикаторов ППК в ОПК привело бы к размыванию структуры компетентностной модели. В одной компетенции оказались бы смешаны базовые требования ФГОС и профильные, более детализированные результаты обучения, что затруднило бы их оценивание, снизило бы прозрачность связи между ФГОС и программой и ослабило бы акцент на профессиональной специфике подготовки.

Методически более корректно сохранить ОПК-1 и ОПК-6 как базовые общепрофессиональные компетенции, а ППК-Р1 и ППК-Р3 – как самостоятельные профильные компетенции.

Профессиональные компетенции, соответствующие роли инженера DevOps и опережающей роли руководителя группы разработки из КРМ, формируемые в рамках образовательной программы, представлены в ЦКМВ (Приложение 1).

4.2. Логика последовательности изучения дисциплин

Логика последовательности изучения дисциплин (принцип «от простого к сложному»):

Программа построена по принципу поступательного наращивания сложности и формирования профессиональных компетенций от базовых навыков программирования до проектирования сложных архитектурных решений и анализа больших данных.

В таблице представлены все дисциплины профессионального цикла (включая обязательную часть и часть, формируемую участниками образовательных отношений, а также факультативы), которые формируют профессиональные компетенции.

Полный перечень профильных дисциплин с распределением по курсам, семестрам и привязкой к результатам обучения образовательной программы

№	Дисциплина (индекс)	Семестр	з.е.	Форма контроля	Пререквизиты	Постреквизиты	Формируемые индикаторы (из ЦКМВ)	Краткое обоснование
1.	Б1.О.09 Информатика*	1	4	Экзамен	Школьный курс информатики	Основы программирования	-	Базовые навыки работы с ИТ, поиск информации, офисное ПО.
2.	Б1.О.15 Линейная и векторная алгебра	1	4	Экзамен	Школьный курс математики	Алгоритмы и структуры данных, Машинное обучение и анализ данных, Дифференциальные уравнения и численные методы, Теория вероятностей и математическая статистика	ППК-Р1.1, ППК-Р1.2	Дисциплина формирует математический фундамент для формализации и моделирования прикладных задач, что является основой для алгоритмизации и программной реализации. Инженер DevOps должен уметь формализовать задачи предметной области и выбирать адекватный математический аппарат для описания сущностей и отношений. Это соответствует ЦКМВ в части способности к формализации и алгоритмизации поставленных задач, что является основой для всех последующих инженерных дисциплин.
3.	Б1.О.21 Основы информационной безопасности	1	4	Зачет с оценкой	Школьный курс информатики, базовые навыки работы с операционной системой (Linux/Windows)	Кибербезопасность информационных систем, Администрирование операционных систем, Кибериммунная разработка, Объектно-ориентированное программирование, Учебная практика	ППК-ОС2.1	Дисциплина закладывает базовое понимание информационной безопасности – критически важного аспекта DevOps-культуры. Инженер DevOps должен понимать виды политик управления доступом, архитектуру

						(ознакомительная практика)		подсистем защиты и принципы противодействия угрозам. В ЦКМВ это соответствует формированию фундамента для DevSecOps-практик, которые являются неотъемлемой частью современных DevOps-процессов.
4.	Б1.О.10 Математический анализ*	1,2	8	Зачет, Экзамен	Школьный курс математики	Теория вероятностей и математическая статистика, Дифференциальные уравнения и численные методы	-	Необходим для статистики и численных методов.
5.	Б1.О.13 Основы программирования	1,2	10	Зачет с оценкой, Курсовая работа, Экзамен	Школьный курс информатики	Системы контроля версий, Объектно-ориентированное программирование, Базы данных, Алгоритмы и структуры данных, Автоматизация сборки, тестирования и доставки ПО, Тестирование программного обеспечения, Web-разработка и API-интеграция, Управление IT-проектами, Учебная практика (ознакомительная практика), Производственная практика (технологическая (проектно технологическая) практика) – 4 семестр	ППК-Р1.1, ППК-Р1.4, ППК-Р1.5, ППК-Р7.1, УНК-02.01	Дисциплина формирует базовые навыки разработки программного кода, алгоритмизации и работы со средой разработки. Инженер DevOps должен уметь разрабатывать алгоритмы, писать код с соблюдением стандартов и использовать системы контроля версий. В ЦКМВ это соответствует формированию фундаментальных компетенций в области разработки и отладки программного обеспечения.
6.	Б1.О.14 Дискретная математика и математическая логика	2	5	Экзамен	Школьный курс математики	Алгоритмы и структуры данных, Базы данных, Теория вероятностей и математическая статистика,	ППК-Р1.1, ППК-Р1.2, ППК-Р1.3	Дисциплина формирует логическое мышление и навыки формализации, необходимые для построения корректных

						Архитектура ЭВМ и операционные системы		алгоритмов и верификации программных решений. Инженер DevOps должен применять методы дискретной математики для анализа и выбора алгоритмов. В ЦКМВ это соответствует компетенциям, связанным с алгоритмизацией и обоснованным выбором методов реализации.
7.	Б1.О.22 Архитектура ЭВМ и операционные системы	2	4	Зачет с оценкой	Дискретная математика и математическая логика	Администрирование операционных систем, Сетевые технологии, Оркестрация контейнеров и Kubernetes, Учебная практика (ознакомительная практика)	ППК-ОС1.1, ППК-ОС1.2, ППК-ОС2.1	Дисциплина даёт понимание аппаратной и системной основы для оптимизации ПО, диагностики инцидентов и администрирования. Инженер DevOps должен знать архитектуру средств ОС, принципы функционирования и уметь контролировать работу систем. В ЦКМВ это основа для компетенций по обслуживанию операционных систем и администрированию средств защиты информации.
8.	Б1.В.02 Системы контроля версий	2	4	Экзамен	Основы программирования	Автоматизация сборки, тестирования и доставки ПО, Web-разработка и API-интеграция, Программная инженерия и DevOps, Непрерывная интеграция, доставка и развертывание (CI/CD/DevOps); Автоматизация DevOps	ППК-Р1.4, ППК-Р1.6, ДИ-1.1, УНК-02.01	Дисциплина формирует культуру командной разработки и понимание важности контроля версий как основы DevOps-процессов. Инженер DevOps должен работать с системами управления версиями, применять

						процессов, Разработка микросервисов, Учебная практика (ознакомительная практика), Производственная практика (технологическая (проектно технологическая) практика)-4 семестр		стратегии ветвления и соблюдать регламенты командной работы. В ЦКМВ это соответствует формированию культуры командной разработки и пониманию ролей в DevOps-цепочке.
9.	Б1.О.12 Базы данных	3	4	Экзамен	Основы программирования, Дискретная математика и математическая логика, Учебная практика (ознакомительная практика)	Web-разработка и API-интеграция, Проектирование информационных систем, Управление данными и DataOps, Производственная практика (технологическая (проектно технологическая) практика)-4 семестр, Производственная практика (преддипломная практика)	ППК-Р4.1, ППК-Р4.2	Дисциплина формирует навыки работы с данными и их хранением, что необходимо для автоматизации CI/CD, управления конфигурациями и мониторинга. Инженер DevOps должен проектировать структуры БД и оптимизировать запросы. В ЦКМВ это соответствует компетенциям по использованию баз данных при создании программных модулей и компонентов.
10.	Б1.О.17 Алгоритмы и структуры данных	3	4	Экзамен	Линейная и векторная алгебра, Основы программирования, Дискретная математика и математическая логика	Машинное обучение и анализ данных, Распределённые системы и микросервисная архитектура, Проектирование информационных систем, Производственная практика (технологическая (проектно-технологическая) практика)-4 семестр	ППК-Р1.1, ППК-Р1.3, ППК-Р1.4	Дисциплина формирует фундаментальное понимание эффективности кода – основу для оптимизации производительности и масштабирования систем. Инженер DevOps должен уметь выбирать оптимальные алгоритмы и оценивать их эффективность. В ЦКМВ это соответствует компетенциям по формализации и алгоритмизации задач и

								обоснованному выбору методов реализации.
11.	Б1.О.18 Сетевые технологии	3	4	Зачет с оценкой	Архитектура ЭВМ и операционные системы	Распределённые системы и микросервисная архитектура, Web-разработка и API-интеграция, DevOps-администрирование систем, Кибербезопасность информационных систем	ППК-Р3.1, ППК-Р3.2, ППК-ОС1.1	Дисциплина даёт понимание сетевой инфраструктуры, необходимое для настройки CI/CD, оркестрации, мониторинга и диагностики. Инженер DevOps должен знать интерфейсы взаимодействия и уметь разрабатывать процедуры интеграции. В ЦКМВ это основа для компетенций по интеграции программных модулей и обслуживанию операционных систем.
12.	Б1.О.19 Объектно-ориентированное программирование	3	4	Зачет с оценкой	Основы программирования, Учебная практика (ознакомительная практика)	Web-разработка и API-интеграция, Программная инженерия и DevOps, Разработка микросервисов, Проектирование информационных систем, Производственная практика (технологическая (проектно технологическая) практика)-4 семестр	ППК-Р1.4, ППК-Р1.5, ППК-Р1.7	Дисциплина формирует навыки промышленной разработки, модульного кода и тестируемости. Инженер DevOps должен разрабатывать код с использованием ООП-парадигмы и методологий разработки. В ЦКМВ это соответствует компетенциям по разработке и отладке программного кода.
13.	Б1.В.03 Автоматизация сборки, тестирования и доставки ПО	3	3	Зачет	Основы программирования, Системы контроля версий, Учебная практика (ознакомительная практика)	Тестирование программного обеспечения, Методы CI/CD и управление релизами, Оркестрация контейнеров и Kubernetes, Инфраструктура разработки ПО, Непрерывная интеграция, доставка и развертывание	ППК-Р1.6, ППК-Р1.7, ППК-Р3.1, ППК-Р6.1, ППК-Р6.2, ДИ-2.1	Дисциплина закладывает основы DevOps-инженерии – автоматизацию жизненного цикла ПО. Инженер DevOps должен настраивать CI/CD-пайплайны и интегрировать их с системами контроля версий. В ЦКМВ это основа для всех компетенций,

						(CI/CD/DevOps); Автоматизация DevOps процессов		связанных с непрерывной интеграцией и доставкой.
14.	Б1.О.16 Теория вероятностей и математическая статистика	4	4	Экзамен	Математический анализ, Линейная и векторная алгебра, Дискретная математика и математическая логика	Машинное обучение и анализ данных, Производительность и масштабирование высоконагруженных систем, SRE и надёжность IT сервисов / FinOps и эксплуатация облачных сервисов	ППК-P1.2, ППК-P1.3	Дисциплина даёт математическую основу для анализа данных мониторинга, оценки производительности и обеспечения надёжности систем. Инженер DevOps должен использовать статистические методы для анализа метрик и оценки рисков. В ЦКМБ это основа для SRE-компетенций.
15.	Б1.О.20 Дифференциальные уравнения и численные методы	4	4	Экзамен	Математический анализ, Линейная алгебра и векторная алгебра	Производительность и масштабирование высоконагруженных систем, Инфраструктура разработки программного обеспечения, SRE и надёжность IT сервисов	ППК-P1.2, ППК-P1.3	Дисциплина формирует навыки численного моделирования, необходимые для оптимизации производительности и прогнозирования нагрузки. Инженер DevOps должен выбирать адекватные численные методы для решения прикладных задач. В ЦКМБ это соответствует компетенциям по формализации задач и выбору методов реализации.
16.	Б1.В.04 Кибербезопасность информационных систем	4	4	Экзамен	Основы информационной безопасности, Сетевые технологии	DevOps-администрирование систем, DevOps-администрирование систем, Управление конфигурациями, Версионирование и контейнеризация,	ППК-OC2.1, ППК-OC2.2	Дисциплина формирует компетенции в области безопасности, необходимые для внедрения DevSecOps-практик. Инженер DevOps должен уметь настраивать политики безопасности и администрировать средства

						Производственная практика (преддипломная практика)		защиты. В ЦКМВ это соответствует компетенциям по администрированию средств защиты информации.
17.	Б1.В.05 Администрирование операционных систем	4	5	Зачет с оценкой, Курсовая работа	Основы информационной безопасности, Архитектура ЭВМ и операционные системы, Учебная практика (ознакомительная практика)	DevOps- администрирование систем, Управление конфигурациями, Кибериммунная разработка, Производственная практика (технологическая (проектно технологическая) практика) – 4 семестр, Производственная практика (преддипломная практика)	ППК-ОС1.1, ППК-ОС1.2, ППК-ОС2.1	Дисциплина формирует практические навыки администрирования инфраструктуры – основы эксплуатации и поддержки систем. Инженер DevOps должен уметь выполнять обновления, выявлять инциденты и настраивать политики безопасности. В ЦКМВ это соответствует компетенциям по обслуживанию операционных систем.
18.	Б1.В.06 Тестирование программного обеспечения	4	3	Зачет с оценкой	Основы программирования, Автоматизация сборки, тестирования и доставки ПО	Программная инженерия и DevOps, Разработка микросервисов, DevOps-тестирование, Производственная практика (технологическая (проектно технологическая) практика) – 6 семестр	ППК-Р1.7, ППК-Р3.2	Дисциплина формирует культуру качества кода и тестирования, необходимую для обеспечения надёжности и непрерывной доставки. Инженер DevOps должен уметь отлаживать код и проверять работоспособность продукта. В ЦКМВ это соответствует компетенциям по проверке и отладке программного кода и проверке работоспособности выпусков.
19.	Б1.О.23 Управление IT-проектами	5	4	Зачет с оценкой	Основы программирования, Объектно-ориентированное программирование,	Архитектура информационных систем и DevOps-трансформация,	ППК-Р6.1, РГ-2.1, РГ-2.2,	Дисциплина формирует управленческие компетенции для перехода к опережающей роли

					Алгоритмы и структуры данных	Программная инженерия и DevOps	УНК-02.01, УНК-02.02	«Руководитель группы разработки». Инженер DevOps должен применять Agile-методологии, использовать DORA-метрики и управлять процессами разработки. В ЦКМВ это соответствует компетенциям по организации процессов разработки и управлению IT-проектами.
20.	Б1.В.07 Web-разработка и API-интеграция	5	4	Экзамен	Основы программирования, Системы контроля версий, Объектно-ориентированное программирование, Базы данных, Сетевые технологии, Производственная практика (технологическая (проектно технологическая) практика)-4 семестр	Распределённые системы и микросервисная архитектура, Разработка микросервисов, Производственная практика (технологическая (проектно технологическая) практика)-6 семестр	ППК-Р3.1, ППК-Р3.2, ППК-Р4.2, ППК-Р4.3	Дисциплина формирует навыки разработки веб-приложений и интеграции через API. Инженер DevOps должен разрабатывать процедуры интеграции и проектировать API. В ЦКМВ это соответствует компетенциям по интеграции программных модулей и использованию баз данных.
21.	Б1.В.08 Методы CI/CD и управление релизами	5	4	Экзамен	Автоматизация сборки, тестирования и доставки ПО, Производственная практика (технологическая (проектно технологическая) практика) – 4 семестр	Оркестрация контейнеров и Kubernetes, Архитектура информационных систем и DevOps-трансформация, Непрерывная интеграция, доставка и развертывание, (CI/CD/DevOps)/ Автоматизация DevOps процессов, Облачные технологии и платформы, Производственная практика (технологическая (проектно технологическая) практика) – 6 семестр,	ППК-Р6.2, ДИ-2.1, ДИ-2.2, ДИ-2.3	Дисциплина формирует продвинутое навыки CI/CD и управления релизами. Инженер DevOps должен настраивать пайплайны, управлять окружениями и стратегиями развертывания. В ЦКМВ это соответствует компетенциям по промышленной разработке и проектированию потоков работ CI/CD.

						Производственная практика (преддипломная практика)		
22.	Б1.В.09 Машинное обучение и анализ данных	5	5	Экзамен	Теория вероятностей и математическая статистика, Линейная и векторная алгебра, Алгоритмы и структуры данных	DevOps/MLOps для больших данных, Методы искусственного интеллекта для разработки ПО и AIOps	ППК-Р7.1, ДИ-4.1, ДИ-4.2, ДИ-4.3	Дисциплина формирует компетенции в области ИИ, необходимые для внедрения AIOps и автоматизации разработки. Инженер DevOps должен применять ИИ-инструменты для генерации кода и анализа данных. В ЦКМВ это соответствует компетенциям по применению ИИ для генерации кода и оптимизации DevOps- процессов.
23.	Б1.В.10 Инфраструктура разработки программного обеспечения	5	4	Зачет с оценкой	Системы контроля версий, Сетевые технологии, Автоматизация сборки, тестирования и доставки ПО, Дифференциальные уравнения и численные методы, Производственная практика (технологическая (проектно технологическая) практика)-4 семестр	Версионирование и контейнеризация, Управление конфигурациями, Оркестрация контейнеров и Kubernetes	ППК-ОС3.1, ППК-ОС3.2, ППК-ОС5.1, ДИ-3.1, ДИ-3.2, ДИ-3.3	Дисциплина формирует ключевые навыки контейнеризации и управления инфраструктурой как кодом. Инженер DevOps должен создавать образы контейнеров и разрабатывать модули IaC. В ЦКМВ это соответствует компетенциям по управлению контейнерными средами и управлению инфраструктурой как кодом.
24.	Б1.В.11 Версионирование и контейнеризация	5	4	Зачет с оценкой	Алгоритмы и структуры данных, Инфраструктура разработки программного обеспечения, Кибербезопасность информационных систем, Производственная практика (технологическая	Оркестрация контейнеров и Kubernetes, Облачные технологии и платформы, Производственная практика (технологическая (проектно технологическая) практика)-6 семестр	ППК-Р1.6, ППК-ОС3.1, ППК-ОС3.2, ППК-ОС3.3	Дисциплина формирует практические навыки контейнеризации и управления версиями артефактов. Инженер DevOps должен создавать и оптимизировать Docker- образы, сканировать их на

					(проектно-технологическая) практика)-4 семестр			уязвимости. В ЦКМВ это соответствует компетенциям по созданию и оптимизации образов контейнеров.
25.	Б1.О.24 Проектирование информационных систем	5,6	9	Экзамен, Зачет с оценкой, Курсовой проект	Базы данных, Объектно-ориентированное программирование, Сетевые технологии, Производственная практика (технологическая (проектно технологическая) практика)-4 семестр	Архитектура информационных систем и DevOps-трансформация, Программная инженерия и DevOps, Производственная практика (технологическая (проектно технологическая) практика)-6 семестр	ППК-Р1.5, ППК-Р6.1, ДИ-1.1, РГ-2.1, УНК-02.01, УНК-02.02	Дисциплина формирует архитектурное мышление и навыки управления проектированием. Инженер DevOps должен управлять проектированием ПО и оформлять техническую документацию. В ЦКМВ это соответствует компетенциям по управлению проектированием и оформлению программного кода.
26.	Б1.В.12 Распределённые системы и микросервисная архитектура	6	4	Экзамен	Web-разработка и API-интеграция, Сетевые технологии	Разработка микросервисов, Производительность и масштабирование высоконагруженных систем, SRE и надёжность IT сервисов / FinOps и эксплуатация облачных сервисов, Производственная практика (технологическая (проектно технологическая) практика) – 6 семестр	ППК-Р6.1, ППК-ОС3.2, ППК-ОС6.1, ДИ-1.3	Дисциплина формирует навыки проектирования распределённых систем и микросервисов. Инженер DevOps должен проектировать отказоустойчивые архитектуры и использовать паттерны. В ЦКМВ это соответствует компетенциям по промышленной разработке, управлению контейнерными средами и надёжности ИТ-сервисов.
27.	Б1.В.13 Архитектура информационных систем и DevOps-трансформация	6	4	Экзамен	Методы CI/CD и управление релизами, Инфраструктура разработки программного обеспечения Проектирование	Программная инженерия и DevOps, Оркестрация контейнеров и Kubernetes	ППК-ОС4.1, ДИ-1.2, ДИ-1.3, ДИ-2.1, РГ-2.1	Дисциплина формирует компетенции для внедрения DevOps-культуры и трансформации процессов. Инженер DevOps должен разрабатывать планы трансформации и

					информационных систем, Управление IT-проектами			автоматизировать ручные процессы. В ЦКМВ это соответствует компетенциям по применению философии DevOps и проектированию потоков работ CI/CD.
28.	Б1.В.14 Кибериммунная разработка	6	3	Зачет с оценкой	Основы информационной безопасности, Кибербезопасность информационных систем, Администрирование операционных систем	DevOps- администрирование систем, Управление конфигурациями	ППК-ОС1.2, ППК-ОС2.3, ДИ-3.2	Дисциплина формирует компетенции в области безопасности прикладного ПО. Инженер DevOps должен анализировать ПО на уязвимости и противодействовать угрозам. В ЦКМВ это соответствует компетенциям по администрированию средств защиты информации прикладного и системного программного обеспечения.
29.	Б1.В.ДВ.01 Дисциплины по выбору 1: Непрерывная интеграция, доставка и развертывание (CI/CD/DevOps) / Автоматизация DevOps процессов	6	4	Зачет с оценкой	Автоматизация сборки, тестирования и доставки ПО; Методы CI/CD и управление релизами; DevOps-разработка мобильных приложений, Системы контроля версий	Оркестрация контейнеров и Kubernetes; Облачные технологии и платформы, Производственная практика (проектная)	ППК-Р6.2, ППК-ОС3.3, ППК-ОС4.2, ДИ-2.1, ДИ-2.3, УНК-02.01, УНК-02.02	Дисциплина формирует продвинутые навыки CI/CD и автоматизации. Инженер DevOps должен настраивать пайплайны и управлять стратегиями развертывания. В ЦКМВ это соответствует компетенциям по промышленной разработке, контейнерным средам, облачной инфраструктуре и CI/CD.
30.	ФТД.01 Управление данными и DataOps	6	1	Зачет	Базы данных	DevOps/MLOps для больших данных	ППК-Р4.2, ППК-Р4.3, ППК-Р7.2	Дисциплина формирует навыки DataOps и управления данными. Инженер DevOps должен управлять схемами данных и использовать ИИ-инструменты для анализа. В

								ЦКМВ это соответствует компетенциям по использованию баз данных и применению ИИ для анализа и отладки кода.
31.	Б1.В.15 Оркестрация контейнеров и Kubernetes	7	4	Зачет с оценкой	Архитектура ЭВМ и операционные системы, Автоматизация сборки, тестирования и доставки ПО, Версионирование и контейнеризация, Инфраструктура разработки программного обеспечения, Производственная практика (технологическая (проектно технологическая) практика) – 4 семестр, 6 семестр	Облачные технологии и платформы, Производственная практика (преддипломная практика), ВКР	ППК-Р6.4, ППК-ОС3.2, ППК-ОС3.3, ДИ-2.2, ДИ-2.3, ДИ-3.1, УНК-02.01, УНК-02.02	Дисциплина формирует ключевые компетенции по управлению контейнерными средами. Инженер DevOps должен разворачивать кластеры Kubernetes и настраивать GitOps. В ЦКМВ это соответствует компетенциям по промышленной разработке, контейнерным средам, CI/CD и инфраструктуре как код.
32.	Б1.В.16 DevOps/MLOps для больших данных	7	4	Зачет с оценкой	Машинное обучение и анализ данных, Управление данными и DataOps	Облачные технологии и платформы, Производственная практика (преддипломная практика), ВКР	ППК-Р7.2, ДИ-4.1	Дисциплина формирует компетенции в области MLOps и AIOps. Инженер DevOps должен настраивать ИИ-инструменты для анализа и отладки кода. В ЦКМВ это соответствует компетенциям по применению ИИ для анализа и отладки кода и по внедрению ИИ-инструментов в DevOps-процессы.
33.	Б1.В.17 Программная инженерия и DevOps	7	6	Экзамен, Курсовой проект	Системы контроля версий, Объектно-ориентированное программирование, Проектирование информационных систем, Архитектура информационных систем и	DevOps-разработка мобильных приложений, Разработка микросервисов, Производственная практика (преддипломная практика), ВКР	ППК-Р6.3, ППК-Р6.4, ППК-Р7.2, ППК-Р7.3, ППК-ОС6.2, ППК-ОС6.3, РГ-2.1,	Дисциплина является интегральной вершиной программы, объединяющей все аспекты промышленной разработки и DevOps. Инженер DevOps должен управлять качеством,

					DevOps-трансформация, Тестирование программного обеспечения; Управление IT-проектами, Производственная практика (технологическая (проектно технологическая) практика)-6 семестр		РГ-2.2, РГ-2.3, УНК-02.01, УНК-02.02	процессами и архитектурой, готовясь к опережающей роли. В ЦКМВ это соответствует формированию всех ключевых компетенций на продвинутом уровне, готовя к опережающей роли руководителя группы разработки.
34.	Б1.В.18 Методы искусственного интеллекта для разработки ПО и AIOps	7	4	Зачет с оценкой	Машинное обучение и анализ данных	Производственная практика (преддипломная практика), ВКР	ППК-Р6.3, ППК-Р6.4, ППК-Р7.3, ДИ-4.2, ДИ-4.3	Дисциплина формирует передовые компетенции в области применения ИИ в DevOps. Инженер DevOps должен использовать ИИ для рефакторинга и оптимизации инфраструктуры. В ЦКМВ это соответствует компетенциям по оптимизации кода с помощью ИИ, применению AIOps и генерации инфраструктурного кода с применением ИИ.
35.	Б1.В.19 DevOps-администрирование систем	7	5	Экзамен	Сетевые технологии, Администрирование операционных систем, Кибербезопасность информационных систем, Киберимунная разработка	Облачные технологии и платформы, Производственная практика (преддипломная практика), ВКР	ППК-ОС1.3, ППК-ОС3.2	Дисциплина формирует продвинутые навыки администрирования и безопасности. Инженер DevOps должен планировать нагрузку и восстанавливать системы при сбоях. В ЦКМВ это соответствует компетенциям по обслуживанию операционных систем и администрированию средств защиты информации.

36.	Б1.В.20 Управление конфигурациями	7	3	Зачет	Инфраструктура разработки программного обеспечения, Администрирование операционных систем, Кибербезопасность информационных систем, Производственная практика (технологическая (проектно технологическая) практика)-6 семестр	Облачные технологии и платформы, Производственная практика (преддипломная практика), ВКР	ДИ-2.2, ДИ-3.2, ДИ-3.3, РГ-2.4	Дисциплина формирует продвинутые навыки управления инфраструктурой и конфигурациями. Инженер DevOps должен внедрять практики тестирования и безопасности IaC. В ЦКМВ это соответствует компетенциям по CI/CD, управлению инфраструктурой как кодом и управлению конфигурациями и выпусками.
37.	Б1.В.ДВ.02 Дисциплины по выбору 2: DevOps- тестирование / DevOps-поддержка и сопровождение	7	3	Зачет	Тестирование программного обеспечения; Автоматизация сборки, тестирования и доставки ПО; Непрерывная интеграция, доставка и развертывание	Тестирование программного обеспечения; Автоматизация сборки, тестирования и доставки ПО; Непрерывная интеграция, доставка и развертывание	ППК-Р3.2, ППК-ОС6.3	Дисциплина формирует навыки тестирования, поддержки и инцидент- менеджмента. Инженер DevOps должен организовывать процессы инцидент-менеджмента и проводить постмортемы. В ЦКМВ это соответствует компетенциям по интеграции и проверке работоспособности и по организации процессов инцидент-менеджмента и непрерывного улучшения.
38.	ФТД.02 Производительность и масштабирование высоконагруженных систем	7	1	Зачет	Теория вероятностей и математическая статистика, Дифференциальные уравнения и численные методы, Распределённые системы и микросервисная архитектура	SRE и надёжность IT сервисов / FinOps и эксплуатация облачных сервисов, Производственная практика (преддипломная); Выпускная квалификационная работа	ППК-ОС4.3, ППК-ОС5.2, ППК-ОС5.3, ППК-ОС6.1, ДИ-4.2	Дисциплина формирует экспертные навыки в области производительности и масштабирования. Инженер DevOps должен проводить нагрузочное тестирование и выявлять узкие места. В ЦКМВ это

								соответствует компетенциям по облачной инфраструктуре, мониторингу, надёжности ИТ-сервисов и применению ИИ в DevOps на продвинутом уровне.
39.	Б1.В.21 Разработка микросервисов	8	4	Зачет с оценкой	Системы контроля версий, Объектно-ориентированное программирование, Web-разработка и API-интеграция, Распределённые системы и микросервисная архитектура, Программная инженерия и DevOps, Производственная практика (технологическая (проектно технологическая) практика) – 6 семестр	Производственная практика (преддипломная); Выпускная квалификационная работа	ППК-ОС6.1, РГ-2.2, РГ-2.3, УНК-02.01, УНК-02.02	Дисциплина формирует практические навыки разработки микросервисов и управления процессами. Инженер DevOps должен управлять процессом разработки и запросами на изменения. В ЦКМВ это соответствует компетенциям по надёжности ИТ-сервисов и организации процессов разработки на продвинутом уровне, готова к роли руководителя группы разработки.
40.	Б1.В.22 Облачные технологии и платформы	8	4	Экзамен	Оркестрация контейнеров и Kubernetes, DevOps/MLOps для больших данных, Методы CI/CD и управление релизами; Управление конфигурациями; DevOps-администрирование систем, Производственная практика (технологическая (проектно технологическая) практика) – 6 семестр	Производственная практика (преддипломная); Выпускная квалификационная работа	ППК-ОС4.1, ППК-ОС4.2, ППК-ОС4.3, ДИ-4.2	Дисциплина формирует ключевые компетенции по работе с облачными платформами. Инженер DevOps должен выбирать облачные сервисы и управлять развертыванием. В ЦКМВ это соответствует компетенциям по облачной инфраструктуре на продвинутом уровне.
41.	Б1.В.23 DevOps-разработка мобильных приложений	8	4	Зачет с оценкой	Программная инженерия и DevOps, Непрерывная интеграция, доставка и	Производственная практика (преддипломная); Выпускная квалификационная работа	ДИ-2.2, РГ-2.4	Дисциплина формирует специализированные навыки DevOps в мобильной разработке. Инженер DevOps

					развертывание (CI/CD/DevOps)			должен настраивать CI/CD для мобильных приложений и управлять выпусками. В ЦКМВ это соответствует компетенциям по CI/CD и управлению конфигурациями и выпусками программного продукта в контексте мобильной разработки.
42.	Б1.В.ДВ.03 Дисциплины по выбору 3: SRE и надёжность ИТ-сервисов / FinOps и эксплуатация облачных сервисов	8	4	Зачет с оценкой	Теория вероятностей и математическая статистика, Дифференциальные уравнения и численные методы, Производительность и масштабирование высоконагруженных систем; Распределённые системы и микросервисная архитектура; Методы CI/CD и управление релизами	Производственная практика (преддипломная); Выпускная квалификационная работа	ППК-ОС5.1, ППК-ОС5.2, ППК-ОС5.3, ППК-ОС6.1, ППК-ОС6.2	Дисциплина формирует экспертные компетенции в области SRE и FinOps. Инженер DevOps должен внедрять SLO/SLI, управлять бюджетом ошибок и оптимизировать затраты. В ЦКМВ это соответствует компетенциям по мониторингу и надёжности ИТ-сервисов на продвинутом уровне, что является вершиной SRE-траектории.

Примечание: *Дисциплины «Информатика» (Б1.О.09), «Математический анализ» (Б1.О.10) с точки зрения образовательного процесса относятся к фундаментальной подготовке и обычно рассматриваются как профильные для соответствующих направлений. Однако в рамках гранта по фундаментальным дисциплинам, который выиграл КНИТУ, где компетенции заранее определены условиями гранта и не подлежат изменению, указанные дисциплины в ЦКМВ не выделяются как профильные; при этом они обеспечивают формирование общепрофессиональных компетенций (ОПК) и закладывают математический фундамент для дальнейшего обучения.

5. Взаимодействие с индустрией

Якорный партнёр – ООО «ГК «Иннотех» – участвует в разработке и экспертизе ОП, отборе студентов, руководстве практиками, предоставлении кейсов и данных, а также входит в состав ГЭК. Дополнительный партнёр – АО «Лаборатория Касперского» – обеспечивает методологическое сопровождение по кибериммунной разработке и предоставляет сопровождение ПО KasperskyOS. Детали взаимодействия зафиксированы в протоколах совместных встреч и дорожной карте. (Приложение 5).

5.1. Договоренности с партнерами:

- Соглашение о сотрудничестве № GKI25-12-11-SPC от 29.12.2025.
- Договор о софинансировании № GKI26-05-359-LLC от 26.05.2026.
- Представитель партнера входит в конкурсную комиссию для внутреннего отбора студентов.
- Представитель партнера входит в состав Государственной экзаменационной комиссии.
- Темы ВКР согласовываются с партнером, и студенты старших курсов распределяются между наставниками от партнера для экспертизы.
- Наставники от партнера участвуют в руководстве практиками и стажировками
- Представители партнера ведут профильные дисциплины.

Материалы взаимодействия с партнёрами представлены в Приложении 5.

5.2. Заказ на программу: целевой образ выпускника и требования индустрии

Современная ИТ-индустрия испытывает острый дефицит инженеров DevOps, способных не только писать код, но и выстраивать полный жизненный цикл программного обеспечения – от автоматизации сборки и тестирования до контейнеризации, оркестрации, управления облачной инфраструктурой и мониторинга высоконагруженных систем. В ответ на этот запрос образовательная программа топ-уровня «Разработка программного обеспечения и DevOps» по направлению 09.03.02 «Информационные системы и технологии» формирует выпускника, который по завершении обучения готов немедленно включиться в индустриальную разработку на позиции инженера DevOps.

Целевой портрет выпускника:

- Проектирует, настраивает и сопровождает CI/CD-пайплайны полного цикла, реализуя стратегии развертывания и управляя откатами релизов;
- создаёт и оптимизирует Docker-образы, разворачивает и администрирует кластеры Kubernetes с настройкой мониторинга, логирования, RBAC и сетевых политик, применяя GitOps-подход;
- описывает инфраструктуру через Terraform/Ansible, управляет состоянием и дрейфом конфигураций, внедряет policy-as-code и тестирование инфраструктурного кода;
- применяет методы машинного обучения для предиктивной аналитики и разработки цифровых двойников технологических процессов;
- внедряет системы наблюдаемости (метрики, логи, трейсы), определяет SLO/SLI, настраивает оповещения и проводит анализ инцидентов с последующими постмортемами;
- работает в кросс-функциональной команде по Agile/Scrum, участвует в code review, планировании спринтов и внедрении DORA-метрик для непрерывного улучшения процессов;
- интегрирует SAST/DAST-инструменты, сканирование контейнеров и управление секретами в CI/CD-пайплайны.

Карьерный трек: в перспективе 3–5 лет – возглавить группу разработки, соединяя глубокую техническую экспертизу с управленческими компетенциями.

Основная профессиональная роль (КРМ): инженер-DevOps.

Опережающая профессиональная роль: руководитель группы разработки.

5.3. Кейсы от индустриального партнера

В основу практической подготовки положена система из 11 сквозных кейсов, интегрированных в дисциплины и практики. Кейсы выстроены по принципу нарастания сложности, что обеспечивает поступательное формирование профессиональных компетенций.

№	Наименование кейса	Описание	Дисциплина	Семестр
1.	Автоматизация рутинных задач с помощью Bash	Разработать набор Bash-скриптов для автоматизации рутинных задач, включая резервное копирование файлов, анализ логов, проверку доступности сервисов и очистку временных данных. Обеспечить параметризацию скриптов, обработку ошибок и их запуск по расписанию с использованием cron.	Основы программирования, Учебная практика	2
2.	CI-пайплайн для проверки качества код	Настроить CI-пайплайн в GitLab CI, GitHub Actions или аналогичной системе, обеспечивающий автоматический запуск сборки проекта, модульных тестов, линтеров и статического анализа кода при каждом push или pull request. Обеспечить базовый процесс контроля качества кода перед слиянием изменений.	Автоматизация сборки, тестирования и доставки ПО	3
3.	Linux-сервер для учебного веб-приложения	Развернуть виртуальный сервер на базе Linux, настроить учетные записи пользователей и права доступа, конфигурировать SSH-доступ, базовые правила межсетевого экрана, а также системные службы и механизмы журналирования. Подготовить серверную среду, готовую к размещению простого веб-приложения, и разработать краткую инструкцию по эксплуатации и администрированию.	Администрирование операционных систем	4
4.	Контейнеризация веб-приложения	Выполнить контейнеризацию веб-приложения и базы данных с использованием Docker. Описать окружение с помощью Dockerfile и docker-compose.yml, настроить переменные окружения, тома	Версионирование и контейнеризация	5

		хранения данных (volumes) и сетевое взаимодействие между сервисами.		
5.	Инфраструктура как код для учебного сервиса	Установить и сконфигурировать серверную инфраструктуру учебного сервиса с использованием инструментов «инфраструктура как код» (Ansible или Terraform), включая установку необходимых пакетов, настройку пользователей, межсетевого экрана, Docker, структуры каталогов проекта и системных служб. Обеспечить воспроизводимость процесса подготовки сервера без выполнения ручной настройки.	Инфраструктура разработки программного обеспечения	5
6.	Автоматическая сборка и публикация Docker-образа	Настроить CI-процесс для автоматической сборки Docker-образа приложения, его тегирования, проверки запуска контейнера и публикации в реестре контейнеров. Обеспечить применение принципов версионирования, безопасного хранения секретов и воспроизводимой сборки.	Непрерывная интеграция, доставка и развертывание (CI/CD/DevOps) / Автоматизация DevOps процессов	6
7.	CI-конвейер с контролем качества и политикой слияния изменений	Настроить CI-конвейер для проекта, включающий автоматическую сборку, выполнение модульных и интеграционных тестов, линтеров, статического анализа кода, расчет покрытия и публикацию отчетов. Спроектировать политику ветвления и слияния изменений, обеспечивающую запрет на принятие merge request или pull request без успешного прохождения всех проверок, с целью формирования инженерной дисциплины в командной разработке.	Непрерывная интеграция, доставка и развертывание (CI/CD/DevOps) / Автоматизация DevOps процессов	6
8.	CD-пайплайн для развертывания приложения на сервере	Реализовать CD-пайплайн для автоматического развертывания приложения на удаленном Linux-сервере после успешного прохождения CI, с использованием SSH, Docker Compose и переменных окружения. Обеспечить поддержку сценария отката (rollback) и проверку работоспособности сервиса после деплоя.	Оркестрация контейнеров и Kubernetes	7

9.	Инфраструктура как код для нескольких окружений	Описать и реализовать инфраструктуру учебного сервиса для нескольких окружений (development, staging и production-like) с использованием инструментов «инфраструктура как код» (Ansible, Terraform или их сочетание), автоматизировав подготовку серверов, установку зависимостей, настройку межсетевого экрана, пользователей, Docker, структуры каталогов проекта, переменных окружения и системных служб. Обеспечить идемпотентность, параметризацию и возможность воспроизведения инфраструктуры с нуля для каждого окружения.	Управление конфигурациями	7
10.	Мини-платформа для развертывания микросервисного приложения	Спроектировать и реализовать мини-платформу для развертывания микросервисного приложения, объединяющую контейнеризацию сервисов, процессы CI/CD, управление конфигурацией окружений и секретами, средства мониторинга и эксплуатационную документацию. Обеспечить интеграцию основных практик DevOps в рамках единого индивидуального проекта.	Программная инженерия и DevOps	7
11.	Production-like DevOps-контур для микросервисного приложения	Спроектировать комплексный Production-like DevOps-контур для небольшого микросервисного приложения, включающий контейнеризацию сервисов, CI/CD, реестр образов, несколько окружений, управление конфигурациями и секретами, миграции базы данных, мониторинг, логирование, алертинг и rollback. Обеспечить целостный процесс поставки изменений от коммита до работающего релиза.	Разработка микросервисов	8

Последовательность кейсов опирается на поэтапное раскрытие индикаторов КРМ: от базовых умений работы с кодом и инфраструктурой до продвинутого DevOps-контура.

1. 2–4 семестры: базовые и средние уровни (ППК-Р1, ППК-ОС1).

– Bash-автоматизация и первичные DevOps-задачи формируют умения формализации задач, написания и отладки кода, работы с ОС и скриптами (индикаторы ППК-Р1.1–Р1.4, Р1.7, ППК-ОС1.1–ОС1.2).

– Кейс с Linux-сервером переводит абстрактные знания по администрированию в практику настройки сервисов, прав доступа, журналирования и базовой защиты, закрывая часть индикаторов по эксплуатации ОС и серверов (ППК-ОС1, ППК-ОС2).

2. 5–6 семестры: выход в промышленную разработку и CI/CD (ППК-Р3, ППК-Р4, ППК-Р6, ППК-ОС3)

– Контейнеризация и инфраструктура как код обеспечивают переход от «ручной» настройки к воспроизводимой инфраструктуре и управляемым окружениям, что отвечает индикаторам по интеграции модулей, работе с БД и применению IaC (ППК-Р3.1–Р3.2, ППК-Р4.*, ППК-Р6.2, ППК-ОС3.1).

– Кейсы по автоматической сборке Docker-образов и CI-конвейерам с политикой слияния развивают индикаторы по использованию инструментов промышленной разработки, CI/CD и инженерной дисциплине в командной работе (ППК-Р6.1–Р6.2, РГ-2.3–РГ-2.4, ППК-ОС3.3).

3. 7 семестр: оркестрация, multi-env и интеграция практик (ППК-ОС3, ППК-ОС4, РГ-2)

– CD-пайплайн и инфраструктура как код для нескольких окружений обеспечивают развитие индикаторов по развертыванию контейнеризированных приложений, управлению конфигурациями и поддержке нескольких сред (ППК-ОС3.2–ОС3.3, ППК-ОС4.2, РГ-2.4).

– Мини-платформа для микросервисов интегрирует все предыдущие компетенции: контейнеры, CI/CD, конфигурации, мониторинг и документацию, что соответствует продвинутым индикаторам по организации процессов разработки и эксплуатации (ППК-Р6.3–Р6.4, ППК-ОС5.*, РГ-2.2–РГ-2.4).

4. 8 семестр: продвинутый DevOps-контур (ТОП-уровень, ППК-ОС3, ППК-ОС4, ППК-Р6)

– Production-like DevOps-контур для микросервисов завершает траекторию, комплексно закрывая индикаторы по проектированию архитектуры, развертыванию и оптимизации облачной/контейнерной инфраструктуры, обеспечению надёжности, откатов, мониторинга и FinOps-подходов (ППК-ОС3., ППК-ОС4., ППК-Р6.3–Р6.4).

Таким образом, каждый последующий кейс опирается на уже сформированные знания и умения, последовательно переводя студента от базовых индикаторов КРМ (работа с кодом и ОС) к продвинутым DevOps-компетенциям и комплексному управлению жизненным циклом ИТ-продукта.

5.4. Практическая подготовка

Практики организованы на базе индустриальных партнёров с индивидуальными планами. Общий объём практик – 24 з.е. Практики начинаются не позднее 2-го семестра.

Вклад практики в достижение ЦКМВ

Вид практики	Семестр	з.е.	Форма контроля	Компетенции	Профессиональные задачи	Ожидаемый результат
Учебная практика (ознакомительная практика) Кейс № 1	2	3	Зачёт с оценкой	ППК-Р1.1 – Р1.6; ППК-Р7.1; ППК-ОС1.1, 1.2; ППК-ОС2.1; ДИ-1.1; УНК-02.01	Выполнять формализацию простых задач предметной области и выбирать базовые алгоритмы и структуры данных для их решения на выбранном языке программирования. Разрабатывать и отлаживать несложные программные модули, использовать базовые средства отладки и ведения журналов. Осваивать начальные приёмы работы с системой управления версиями: инициализация репозитория, фиксация изменений, работа с простыми ветками и слиянием. Применять генеративные ИИ-инструменты для поддержки написания и рефакторинга учебного кода с пониманием ограничений и рисков. Выполнять элементарные операции администрирования ОС: создание учётных записей, установка и обновление программного обеспечения, базовые настройки безопасности. Идентифицировать основные угрозы информационной безопасности на уровне ОС и прикладного ПО, определять базовые меры защиты. Соблюдать требования учебной документации и базовые нормы профессиональной этики и командного взаимодействия.	Студент демонстрирует умение формулировать учебные ИТ-задачи в терминах алгоритмов и структур данных и реализовывать их в виде небольших программ. Умеет пользоваться системой контроля версий на базовом уровне в индивидуальной и парной работе. Понимает базовые принципы работы ОС и элементарные меры ИБ. Может осознанно использовать ИИ-инструменты как вспомогательный инструмент при разработке учебных задач. Проявляет ответственное отношение к срокам, качеству учебных заданий и правилам академической

						честности.
Производственная практика (технологическая (проектно-технологическая) практика)	4	6	Зачёт с оценкой	ППК-Р1.1 – Р1.7; ППК-Р3.1, 3.2; ППК-Р4.1, 4.2; ППК-Р7.1; ППК-ОС1.1, 1.2; ППК-ОС2.1, 2.2; ДИ-1.1, 2.1; УНК-02.01	Выполнять анализ и формализацию типовых задач из реальных (или приближенных к реальным) кейсов компании-партнёра, выбирать алгоритмы и технологии реализации. Разрабатывать, тестировать и отлаживать программные компоненты малой и средней сложности, интегрируя их в существующий код. Применять системы контроля версий в составе команды: ветвление, code review, работа с pull/merge-request. Участвовать в разработке процедур интеграции модулей и первичных процедур проверки выпусков программного продукта. Реализовывать простые сценарии работы с базами данных: запросы, транзакции, взаимодействие с БД из кода, участвовать в уточнении структуры БД. Применять ИИ-инструменты для ускорения разработки и документирования, анализируя корректность предложенных решений. Выполнять типовые операции администрирования рабочих станций и серверов, участвовать в настройке базовых политик безопасности, межсетевого экрана и средств защиты в компьютерных сетях. Соблюдать корпоративные регламенты, базовые требования к документации и взаимодействию в проектной команде.	Студент способен включаться в разработку небольших фрагментов реального проекта, соблюдая принятые стандарты кодирования, документооборота и использования СКВ. Участвует в интеграции модулей и начальных стадиях тестирования выпусков. Демонстрирует устойчивые навыки работы с БД и базовые администраторские операции в среде, близкой к производственной. Готов брать ответственность за отдельные задачи в составе команды, соблюдая профессиональную этику и регламенты компании.

Производственная практика (технологическая (проектно-технологическая практика))	6	9	Зачёт с оценкой	ППК-Р1.5 – 1.7; ППК-Р3.1, 3.2; ППК-Р4.2, 4.3; ППК-Р6.2; ППК-Р7.2; ППК-ОС1.1; 1.2; ППК-ОС2.3; ППК-ОС3.2, 3.3; ППК-ОС4.1;4.2; ППК-ОС5.1; ППК-ОС6.1; ДИ-1.1 – 1.3, ДИ-2.1 – 2.3; ДИ-3.1 – 3.3; ДИ-4.1 – 4.3; РГ-2.1; 2.2; УНК-02.01, 02	Применять стандарты оформления кода и технической документации, поддерживать читаемость и сопровождаемость программных модулей. Участвовать в проектировании и оптимизации структур баз данных, анализировать производительность запросов и предлагать варианты улучшения. Реализовывать и развивать CI/CD-конвейеры для сборки, тестирования и базового развёртывания приложений, в том числе контейнеризированных. Настраивать и администрировать компоненты инфраструктуры приложений в кластерах Kubernetes и/или облачной инфраструктуре, работать с несколькими окружениями. Применять ИИ-инструменты для анализа и отладки кода, в том числе для выявления дефектов и уязвимостей. Участвовать в обеспечении безопасности на уровне приложений и инфраструктуры: анализ уязвимостей, настройка средств защиты, реагирование на инциденты (ППК-ОС2.3, ППК-ОС1.1–1.2). Участвовать в проектировании системы мониторинга на уровне метрик и журналов для сервисов, формулировать требования к наблюдаемости и алертам (ППК-ОС5.1, ППК-ОС6.1). Включаться в планирование и декомпозицию задач команды, оценивать трудоёмкость и сроки, вести отчётность по выполнению (РГ-2.1, РГ-2.2; ДИ-*, УНК-02.01, 02.02).	Студент способен выполнять полноценные роли junior-инженера в командах разработки/DevOps: поддерживать и развивать CI/CD, работать с Kubernetes/облаком, участвовать в настройке мониторинга и обеспечении безопасности. Демонстрирует умение оптимизировать работу с БД и инфраструктурой, опираясь на метрики и результаты анализа. Понимает процессы планирования и контроля задач в команде, способен аргументированно оценивать и защищать технические решения.
Производственная практика (преддипломная практика)	8	6	Зачёт с оценкой	ППК-Р6.1 – 6.4; ППК-Р7.1 – 7.3; ППК-ОС3.1 – 3.3; ППК-ОС4.1 – 4.3; ППК-ОС5.1 – 5.3; ППК-ОС6.1 – 6.3; ДИ-1.1 – 1.3,	Проектировать и реализовывать комплексный DevOps-контур для реального (или максимально приближенного к реальному) проекта: контейнеризация, CI/CD, несколько окружений, стратегии деплоя и отката (ППК-Р6.2–Р6.4; ППК-ОС3.1–3.3, ППК-ОС4.2). Применять ИИ-инструменты в полном цикле разработки: генерация кода, анализ и оптимизация, автоматизированное тестирование,	Студент демонстрирует готовность к работе на уровне начинающего DevOps/Software Engineer с элементами технического и организационного лидерства.

				ДИ-2.1 – 2.3; ДИ-3.1 – 3.3; ДИ-4.1 – 4.3; РГ-2.1 – 2.4; УНК-02.01, 02.02	оценка качества (ППК-Р7.1–Р7.3). Обеспечивать архитектурную целостность и масштабируемость решений, участвовать в выборе архитектурных шаблонов, технологий и облачных сервисов (ППК-Р6.1, ППК-ОС4.1–4.3). Проектировать и внедрять системы мониторинга, логирования, трассировки и алертинга, управляющие SLO/SLI, и выстраивать процессы реагирования на инциденты (ППК-ОС5.1–5.3, ППК-ОС6.1–6.3). Организовывать работу небольшой проектной группы: распределение задач, контроль сроков, управление изменениями и конфигурациями, оценка рисков (РГ-2.1–2.4). Интегрировать результаты преддипломной практики в выпускную квалификационную работу: формулировать цель, задачи, обосновывать архитектуру и стек технологий, готовить отчёт и презентацию (ДИ-*, УНК-02.01, 02.02).	Способен спроектировать и реализовать production-like DevOps-контур для микросервисного или облачного приложения, обеспечить его наблюдаемость, надёжность и управляемость. Умеет координировать работу небольшой команды, управлять изменениями и конфигурациями, интегрировать практические результаты в ВКР и профессиональное портфолио.
--	--	--	--	---	--	--

Паспорта практик (Приложение 4) содержат описание профессиональных задач, компетенций, артефактов и форматов участия индустриальных партнёров.

5.5. Система стажировок как механизм углубления практической подготовки

Помимо обязательных практик, студентам предоставляется возможность прохождения стажировок на базе индустриального партнёра – ООО «ГК «Иннотех» – и реализуются в соответствии с Положением о проведении стажировок (Приложение 6). Стажировки являются добровольным механизмом углубления практических навыков, который дополняет обязательные практики и сквозные кейсы позволяют студентам:

- Участвовать в реальных производственных проектах.
- Получать актуальный опыт работы в команде.
- Формировать портфолио и готовить материалы для ВКР.

Структура и содержание стажировок

Стажировки организованы по трёхуровневой модели (базовый, углубленный, расширенный) для четырёх профессиональных ролей: «Разработчик», «Тестирующий», «Аналитик», «Прикладной администратор». Содержание каждого уровня соответствует типовым программам, утверждённым индустриальным партнёром и представленным в Приложении 6. Стажировки реализуются в сквозном режиме, параллельно с освоением дисциплин и прохождением обязательных практик. Они позволяют студенту применить и углубить знания, полученные в рамках образовательной программы, на реальных производственных задачах. Соответствие стажировок профессиональным ролям и обобщённое содержание уровней для каждой роли представлено в таблицах.

Соответствие стажировок профессиональным ролям ООП

Роль стажировки (по Положению)	Соответствующая роль в КРМ / Проф.стандарт	Формируемые компетенции (ЦКМВ)
Разработчик	Инженер DevOps / 06.001 Программист	ППК-Р1, ППК-Р3, ППК-Р4, ППК-Р6, ДИ-1 – ДИ-4
Прикладной администратор	Инженер DevOps / 06.026 Системный администратор информационно-коммуникационных систем	ППК-Р6, ППК-ОС1 – ОС-6, РГ-2
Аналитик	Руководитель группы разработки (опережающая роль)	ДИ-1 – ДИ-4, РГ-2, УНК-02

Формируемые ключевые навыки стажировок по ролям и уровням

Роль	Уровень I (Базовый) (12 недель)	Уровень II (Углубленный) (12 недель)	Уровень III (Расширенный) (12 недель)
Разработчик	- Проектирование и реализация небольшого функционального модуля/компонента - Работа с API - Участие в код-ревью - Интеграционное тестирование	- Проектирование и реализация небольшого функционального модуля/компонента - Работа с API - Участие в код-ревью - Интеграционное тестирование	- Оптимизация и рефакторинг существующего кода - Участие в планировании спринта - Деплой и основы мониторинга - Работа с Docker / очередями (Kafka/RabbitMQ)
Прикладной администратор	- Понимание CI/CD процессов - Основы Linux и работа с командной строкой - Работа с системами контроля версий (Git) - Основы контейнеризации (Docker)	- Углубленная работа с Docker и Docker Compose - Основы оркестрации контейнеров (Kubernetes) - Инфраструктура как код (Terraform/Ansible) – основы - Настройка мониторинга и алертинга	- Настройка отказоустойчивости в Kubernetes - Работа с системами хранения (PV/PVC) - Основы сетевой политики и безопасности - Автоматизация бэкапов

Аналитик	- Понимание предметной области и продукта - Описание простых требований - Составление пользовательских историй (User Stories) - Основы работы с диаграммами (BPMN)	- Проведение интервью с пользователями и экспертами - Анализ процессов «как есть / как будет» - Формирование и приоритизация backlog - Работа с прототипами (Figma)	- Полный цикл работы с требованиями - Коммуникация с техническими и бизнес-стейкхолдерами - Участие в разработке архитектурных решений - Поддержка и актуализация backlog команды
-----------------	---	--	--

В качестве расширения профессиональных навыков студенты так же могут пройти стажировку по дополнительным ролям, не связанным напрямую с профилем ООП.

Роль	Уровень I (Базовый) (12 недель)	Уровень II (Углубленный) (12 недель)	Уровень III (Расширенный) (12 недель)
Тестировщик	- Понимание основ тестирования и жизненного цикла дефекта - Проведение функционального тестирования UI - Составление чек-листов и простых тест-кейсов	- Тестирование API (Postman/Insomnia) - Работа с базами данных (SQL-запросы на выборку) - Анализ логов - Оформление тестовой документации	- Проектирование тестовой стратегии для нового функционала - Проведение комплексного тестирования (дымовое, регрессионное, приёмочное) - Работа со сложными тестовыми сценариями и кросс-функциональным тестированием - Анализ и оптимизация тестового покрытия - Участие в планировании тестирования на спринт.

Индивидуальные планы стажировки и организация

Для каждой стажировки разрабатывается Индивидуальный план стажировки (ИПС), который включает:

- ФИО студента, группа, роль и уровень стажировки.
- Наставника от индустриального партнёра.
- Цели этапа с привязкой к компетенциям ЦКМВ.
- Перечень задач с указанием сроков выполнения и критериев оценки.
- Форму отчётности (код в репозитории, документация, устная защита).

Механизм организации:

1. **Наставничество** – за каждым стажёром закрепляется наставник от предприятия, который проводит код-ревью, консультации и оценивает результаты.

2. **Итоговая аттестация** – по завершении уровня наставник готовит Итоговый отчёт, фиксирующий достигнутые компетенции и выполненный объём работ. Аттестация проводится в форме защиты отчёта перед комиссией с участием представителей кафедры и партнёра.

3. **Признание результатов** – результаты стажировки могут быть засчитаны как:

- Выполнение индивидуального задания по практике (при согласовании с руководителем практики).
- Подготовка материалов для ВКР (артефакты, документация, результаты тестирования).
- Дополнительный бонус при формировании рейтинга студента.

Примерная траектория стажировки для студентов ООП

Семестр	Роль / Рекомендуемый уровень стажировки	Связь с учебной деятельностью
После 2-го семестра	Разработчик / Базовый	Стажировка опирается на освоение базовых курсов по программированию, алгоритмам и структурам данных, web-разработке начального уровня, архитектуре ЭВМ и операционным системам, системам контроля версий. На основе этих дисциплин студенты отрабатывают проектирование и реализацию небольших функциональных модулей и компонентов, работу с простыми API, участие в код-ревью под руководством наставника и базовое интеграционное тестирование в рамках учебных и близких к ним производственных задач.
	Прикладной администратор / Базовый	Стажировка опирается на базовые курсы по операционным системам и архитектуре ЭВМ, введению в DevOps и системам контроля версий, в рамках которых студенты знакомятся с принципами сборки и запуска программ, структурой ОС и основами автоматизации. На этой основе формируется понимание CI/CD-процессов, осваиваются основы работы в Linux и командной строке, практикуется использование Git в учебных проектах и выполняются первые шаги в контейнеризации с использованием Docker.
	Аналитик / Базовый)	Стажировка строится на базовых курсах по информатике, введению в профессиональную деятельность и дисциплинам, дающим общее представление о предметной области и сценариях использования цифровых продуктов. На этой основе студенты учатся понимать предметную область и продукт, описывать простые требования в структурированном виде, формулировать пользовательские истории (User Stories) и осваивают основы работы с простыми диаграммами (например, BPMN) для фиксации бизнес-процессов и пользовательских сценариев.
После 4-го семестра	Разработчик / Углубленный	Стажировка продолжает и углубляет результаты курсов по объектно-ориентированному программированию, базам данных, web-разработке и API-интеграции, тестированию программного обеспечения и сетевым технологиям. Студенты выполняют проектирование и реализацию функциональных модулей уже в составе существующих систем, работают с API, участвуют в код-ревью и интеграционном тестировании на реальных проектах, постепенно переходя от учебных примеров к задачам промышленного уровня.
	Прикладной администратор / Углубленный	Стажировка продолжает и углубляет содержание дисциплин по инфраструктуре разработки, системному администрированию, методам CI/CD, версионированию и контейнеризации, а также распределённым системам начального уровня. Студенты переходят от базовой работы с Docker к углублённому использованию Docker и Docker Compose, осваивают основы оркестрации контейнеров в Kubernetes и инфраструктуры как кода (Terraform/Ansible), а также применяют знания по мониторингу и журналированию для настройки базового мониторинга и алертинга сервисов.
	Аналитик / Углубленный	Стажировка опирается на дисциплины по проектированию информационных систем, основам бизнес-анализа, UX/UI и управлению ИТ-проектами, где студенты получают инструменты описания процессов и пользовательских сценариев. На этой базе они осваивают проведение интервью с пользователями и экспертами,

		анализ процессов «как есть / как будет», формирование и приоритизацию backlog, а также работу с прототипами (Figma) для проверки и уточнения требований.
6-й семестр	Разработчик / Расширенный	Стажировка связана с изучением дисциплин по инфраструктуре разработки, DevOps-практикам, распределённым системам и микросервисной архитектуре, архитектуре информационных систем и основам обеспечения надёжности и безопасности. На этой стадии студенты развивают навыки оптимизации и рефакторинга существующего кода, участвуют в планировании спринтов и работе с backlog, осваивают базовый деплой и основы мониторинга, а также работу с Docker и системами очередей (Kafka/RabbitMQ), применяя знания по DevOps и архитектуре в реальной командной разработке.
	Прикладной администратор / Расширенный	Стажировка связана с продвинутыми курсами по оркестрации контейнеров, DevOps/MLOps, архитектуре информационных систем, обеспечению надёжности и безопасности, а также дисциплинам по эксплуатации и поддержке сервисов. На этой стадии студенты закрепляют умения настраивать отказоустойчивость в Kubernetes, работать с системами хранения (PV/PVC), применять основы сетевой политики и безопасности в кластере и автоматизировать бэкапы, тем самым переходя от базового владения инструментами к их использованию в задачах промышленной эксплуатации.
	Аналитик / Расширенный	Стажировка связана с продвинутыми курсами по архитектуре информационных систем, управлению продуктом и проектами, а также дисциплинами, развивающими компетенции ДИ и РГ на старших курсах. На этом этапе студенты переходят к полному циклу работы с требованиями (от сбора и анализа до сопровождения изменений), выстраивают коммуникацию с техническими и бизнес-стейкхолдерами, участвуют в разработке архитектурных решений и обеспечивают поддержку и актуализацию backlog команды в долгосрочном горизонте.

Механизм интеграции с дисциплинами и практиками

Стажировки не заменяют обязательные практики, предусмотренные учебным планом, а расширяют возможности студента для:

1. Углубления компетенций в конкретной профессиональной роли (разработчик, тестировщик, аналитик, прикладной администратор).
2. Участия в реальных производственных проектах на ранних этапах обучения.
3. Подготовки материалов для ВКР на основе реальных задач индустрии.

Прохождение стажировок синхронизировано с календарным графиком учебного процесса и может быть засчитано как часть индивидуальной траектории студента. Результаты стажировок фиксируются в портфолио студента и учитываются при формировании рейтинга.

6. Внутренний отбор студентов

Внутренний отбор проводится на конкурсной основе среди студентов, зачисленных на первый курс.

Правила внутреннего отбора на программу ТОП ИТ отражены в Приложении 7.

7. Государственная итоговая аттестация

Форма ГИА: Защита выпускной квалификационной работы (бакалаврской работы)

Механизмы проведения ГИА:

- государственная итоговая аттестация проводится в 8 семестре после освоения программы в полном объеме;
- государственная аттестационная комиссия включает представителя «якорного» индустриального партнера;
- Защита проходит в формате презентации и демонстрации разработанного продукта.

Согласование тематик ВКР:

- темы ВКР разрабатываются студентом совместно с научным руководителем;
- тематика согласуется с индустриальным партнером не менее чем за 6 месяцев до защиты;
- ВКР должна решать реальную производственную задачу партнера.

Тематики ВКР согласованы с индустриальным «якорным» партнером ООО «ГК «Иннотех» (Приложение 5, протокол № 4 от 24 мая 2026 г.). Темы утверждаются не позднее чем за 6 месяцев до защиты.

Примерные темы ВКР

№	Тема ВКР	Ожидаемый результат
1.	Разработка DevOps-платформы для развертывания микросервисного приложения на базе Kubernetes и GitOps	Платформа, позволяющая развернуть приложение в Kubernetes-кластере одной командой
2.	Разработка модуля автоматизации управления инфраструктурой в облачной среде на основе Infrastructure as Code	Программный модуль для автоматизации развертывания и управления облачной инфраструктурой, обеспечивающий воспроизводимое и идиоматичное развертывание инфраструктуры в облачных средах
3.	Разработка системы наблюдаемости для высоконагруженного веб-сервиса с реализацией SRE-практик	Система наблюдаемости, включающая сбор метрик, логов и трейсов, дашборды для разных ролей, алертинг на основе SLO/SLI, а также сценарий диагностики инцидентов, связывающий все типы данных для ускорения анализа и сокращения MTTR
4.	Разработка модуля контроля качества кода в CI/CD-пайплайне	Программный модуль контроля качества кода для CI/CD: запуск тестов, статический анализ (SonarQube), покрытие, линтеры, SAST/DAST, блокировка слияния при нарушениях и веб-интерфейс отчетов.
5.	Разработка системы мониторинга и наблюдаемости на основе OpenTelemetry	Система наблюдаемости для микросервисного приложения на основе OpenTelemetry, включающая: библиотеку инструментирования кода для сбора метрик, логов и трейсов; модуль агрегации телеметрии (OTEL Collector); хранилища для метрик (Prometheus), логов (Loki) и трейсов (Tempo/Jaeger); модуль корреляции данных с использованием trace_id; набор аналитических дашбордов в Grafana; сценарии диагностики типовых проблем и инструкции по разбору инцидентов.
6.	Разработка программного модуля оптимизации ресурсных затрат облачных платформ на основе методологии FinOps	Система наблюдаемости для микросервисного приложения на основе OpenTelemetry: сбор метрик, логов и трейсов через OTEL Collector, хранение в Prometheus, Loki и Tempo/Jaeger, корреляция по trace_id, дашборды Grafana и сценарии диагностики инцидентов.
7.	Разработка системы резервного копирования и восстановления для распределённого приложения в Kubernetes	Система резервного копирования и восстановления для приложения в Kubernetes: бэкап PostgreSQL с WAL-архивацией и Point-in-Time Recovery, резервирование манифестов, ConfigMaps и Secrets, автоматические бэкапы по расписанию, проверка целостности копий и сценарии восстановления с пошаговой проверкой работоспособности.
8.	Разработка модуля безопасности для CI/CD-пайплайна	Модуль интеграции DevSecOps-практик в CI/CD-пайплайн: сканирование зависимостей и контейнерных образов, SAST и DAST с настраиваемыми правилами безопасности, Policy-as-

		Code на OPA/Rego, а также веб-интерфейс для управления уязвимостями и отчётности.
9.	Разработка модуля AIOps для предиктивной диагностики инцидентов в системе мониторинга	Модуль AIOps для системы мониторинга: автоматический анализ логов и метрик методами машинного обучения, детектирование аномалий, прогнозирование нагрузки и автоскейлинг, рекомендации по устранению инцидентов и сокращение MTTR на основе исторических данных.

8. Подходы к обучению и используемые инструменты

Методы обучения:

Метод	Описание	Применение в программе
Project-based learning (PjBL)	Обучение через выполнение практических проектов	Выполнение проектов на основе реальных задач индустриального партнера (11 кейсов на протяжении всего обучения)
Командная разработка по Agile/Scrum	Работа в командах с распределением ролей	Производственная практика (стажировка), курсовые проекты, участие в хакатонах
Code Review	Обязательная проверка кода наставниками и одногруппниками	Все лабораторные работы, практики и курсовые проекты
Проблемно-ориентированное обучение (PBL)	Решение практических задач с использованием изученных инструментов	Лабораторные работы на основе кейсов от индустриального партнера
Перевернутый класс (Flipped Classroom)	Самостоятельное изучение теории перед практическим занятием	Дисциплины «Основы программирования», «Алгоритмы и структуры данных»
Мастер-классы и воркшопы	Проведение занятий представителями индустриального партнера	Не менее 2-4 мастер-классов в год от ООО «ГК «Иннотех» и Лаборатории Касперского

Инфраструктура и инструментарий

Программа базируется на современном стеке технологий, включающем как отечественные, так и Open Source решения, что соответствует требованиям импортозамещения и КИИ. Выбор каждого инструмента обусловлен его ролью в достижении конкретных результатов обучения и формировании профессиональных компетенций.

Операционные системы и среды выполнения

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Astra Linux Special Edition 1.7 («Орёл»)	Основная целевая ОС для системного программирования, разработки драйверов, модулей ядра и отладки. Сертифицирована ФСТЭК, ФСБ и Минобороны России. Включена в реестр отечественного ПО.	Российская (коммерческая, для образования – специальные условия)	Архитектура ЭВМ и операционные системы Инфраструктура разработки программного обеспечения Версионирование и контейнеризация Оркестрация контейнеров и Kubernetes DevOps-администрирование систем
KasperskyOS (Community)	Платформа для разработки кибериммунных систем, изучения политик изоляции и безопасной интеграции	Российская (бесплатная для образования)	Кибериммунная разработка

Linux (дистрибутивы Debian/Ubuntu – для сравнения и экспериментов)	Используется как альтернативная среда для изучения кроссплатформенности, работы с сетевыми инструментами и контейнеризации (в дополнение к Astra Linux).	Open Source (GPL)	Архитектура ЭВМ и операционные системы Web-разработка и API-интеграция Методы CI/CD и управление релизами Инфраструктура разработки программного обеспечения Версионирование и контейнеризация Распределённые системы и микросервисная архитектура Оркестрация контейнеров и Kubernetes
Ansible	Автоматизация конфигурации серверов и развертывания приложений (Configuration Management, IaC)	GNU GPLv3 (Ansible Core), модули и коллекции чаще всего Apache 2.0/MIT	DevOps-администрирование систем; Методы CI/CD и управление релизами; Оркестрация контейнеров и Kubernetes

Среды разработки и инструменты программирования

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
GigaIDE	Российская импортозамещённая среда разработки со встроенным ИИ-ассистентом GigaCode. Позволяет вести разработку на C++, Python, Java и других языках. Встроенный ИИ-ассистент повышает скорость написания кода до 25%.	Российская (бесплатная для образования)	Основы программирования Алгоритмы и структуры данных Объектно-ориентированное программирование
Кибер IDE (ИТ-холдинг Т1)	Альтернативная российская среда разработки, интегрированная с платформой «Сфера».	Российская	Кибериммунная разработка DevOps-администрирование систем
Visual Studio Code / VSCodium	Кроссплатформенный редактор для разработки на C++, Python, Java с поддержкой плагинов для работы с Git, Docker, отладки.	MIT (Open Source)	Основы программирования Web-разработка и API-интеграция Методы CI/CD и управление релизами Инфраструктура разработки программного обеспечения Разработка микросервисов
SourceCraft Code Assistant	ИИ-ассистент для генерации и оптимизации кода, доступный как плагин для VS Code и IDE JetBrains.	Российская (бесплатная)	Методы искусственного интеллекта для разработки ПО и AIOps Программная инженерия и DevOps DevOps-администрирование систем

GCC 13+ / GDB 14+	Компилятор и отладчик для C/C++, необходимые для системного программирования, отладки ядра и bare-metal разработки.	GPL	Основы программирования Алгоритмы и структуры данных
Make / CMake	Системы сборки проектов на C/C++, управление зависимостями и компиляцией.	GPL / BSD	Инфраструктура разработки программного обеспечения Производительность и масштабирование высоконагруженных систем

Фреймворки и библиотеки для прикладной разработки

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Qt Charts / Qt Data Visualization	Модули для создания графиков, трендов, 2D/3D-визуализации данных в реальном времени. Используются для отображения телеметрии и цифровых двойников.	LGPL / GPL	Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps
Qt 3D	Модуль для создания 3D-визуализаций и цифровых двойников с использованием OpenGL.	LGPL	Машинное обучение и анализ данных
PyQt / PySide	Python-привязки для Qt, используемые при разработке прототипов HMI и скриптов для обработки данных.	GPL / LGPL	Web-разработка и API-интеграция Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps

Базы данных и хранение данных

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
PostgreSQL PRO	Основная российская СУБД для промышленных систем. Включена в реестр отечественного ПО. Используется для хранения телеметрии, конфигураций, журналов событий. Поддерживает транзакции ACID, репликацию, партиционирование.	Российская (коммерческая, для образования – специальные условия)	Базы данных Web-разработка и API-интеграция Машинное обучение и анализ данных Управление данными и DataOps
ClickHouse	Колоночная СУБД для аналитической обработки больших объемов телеметрии в реальном времени.	Apache 2.0	Машинное обучение и анализ данных Управление данными и DataOps
Redis	In-memory база данных для кэширования и быстрого доступа к часто используемым данным (например, кэш сессий, конфигураций).	BSD	Web-разработка и API-интеграция Методы CI/CD и управление релизами DevOps-администрирование систем
MongoDB	Документно-ориентированная NoSQL СУБД для хранения неструктурированных данных (например, логов, конфигураций).	SSPL	Web-разработка и API-интеграция Машинное обучение и анализ данных Управление данными и DataOps

Системы контроля версий, CI/CD и управление проектами

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Платформа «Сфера» (Код, Задачи, Конвейер)	Российская платформа для полного цикла разработки: Git-репозиторий (Сфера.Код), управление задачами (Сфера.Задачи), CI/CD (Сфера.Конвейер). Включена в реестр отечественного ПО.	Российская	Инфраструктура разработки программного обеспечения Методы CI/CD и управление релизами DevOps-администрирование систем Управление IT-проектами
Git	Система распределённого контроля версий – основа для командной разработки, кодревью и управления релизами.	GPL	Основы программирования Системы контроля версий Web-разработка и API-интеграция Версионирование и контейнеризация
GitVerse	Российский Git-репозиторий, альтернатива GitHub.	Российская	Системы контроля версий Инфраструктура разработки программного обеспечения
GitLab CI / Jenkins	Инструменты для настройки CI/CD-пайплайнов: автоматическая сборка, тестирование, развёртывание.	MIT / GPL	Методы CI/CD и управление релизами Инфраструктура разработки программного обеспечения DevOps-администрирование систем
Helm	Пакетный менеджер для Kubernetes – управление развёртыванием приложений.	Apache 2.0	Оркестрация контейнеров и Kubernetes
ArgoCD / Flux	Инструменты GitOps – автоматическое синхронизирование состояния кластера с Git-репозиторием.	Apache 2.0	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы

Контейнеризация и оркестрация

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Docker CE / Podman	Контейнеризация приложений – создание образов, изоляция окружений, воспроизводимость сборки.	Apache 2.0 / GPL	Версионирование и контейнеризация Инфраструктура разработки программного обеспечения Методы CI/CD и управление релизами Оркестрация контейнеров и Kubernetes
Kubernetes (k3s)	Оркестрация контейнеров – управление кластерами, автоматическое масштабирование, self-healing. Легковесная версия k3s для edge-устройств и IoT	Apache 2.0	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы

Мониторинг, логирование и наблюдаемость

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Prometheus	Сбор метрик производительности, мониторинг состояния сервисов, хранение временных рядов.	Apache 2.0	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы DevOps-администрирование систем
Grafana	Визуализация метрик, создание дашбордов для	AGPL	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы

	мониторинга, интеграция с Prometheus, Loki, Alertmanager.		DevOps-администрирование систем
Alertmanager	Управление оповещениями на основе метрик, эскалация, интеграция с системами уведомлений.	Apache 2.0	Оркестрация контейнеров и Kubernetes DevOps-администрирование систем
ELK Stack (Elasticsearch, Logstash, Kibana) / Loki + Grafana	Централизованное логирование, сбор и анализ логов, поиск инцидентов.	SSPL / AGPL	DevOps-администрирование систем Методы CI/CD и управление релизами SRE и надёжность IT-сервисов
OpenTelemetry	Сбор телеметрии (метрики, логи, трейсы) для распределённых систем.	Apache 2.0	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы
Jaeger	Распределённая трассировка запросов в микросервисной архитектуре – анализ задержек, диагностика.	Apache 2.0	Оркестрация контейнеров и Kubernetes Облачные технологии и платформы

Инструменты отладки, профилирования и тестирования

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Selenium	Автоматизированное тестирование web-интерфейсов (UI-тесты, end-to-end). habr	Apache 2.0. habr	Web-разработка и API-интеграция Методы CI/CD и управление релизами
QEMU	Эмулятор аппаратных платформ для отладки и тестирования кода без физического оборудования.	GPL	Архитектура ЭВМ и операционные системы DevOps-администрирование систем
Google Test / PyTest	Фреймворки для модульного и интеграционного тестирования кода на C++ и Python.	BSD / MIT	Основы программирования Алгоритмы и структуры данных Объектно-ориентированное программирование Web-разработка и API-интеграция
Postman / Insomnia	Тестирование REST API, отладка HTTP-запросов, создание коллекций тестов.	Freemium / Apache 2.0	Web-разработка и API-интеграция
k6 / JMeter / Locust	Нагрузочное тестирование – проверка производительности и масштабируемости систем.	AGPL / Apache 2.0	Методы CI/CD и управление релизами DevOps-администрирование систем
Wireshark / tcpdump	Анализ сетевого трафика, отладка промышленных протоколов, диагностика сетевых проблем.	GPL	Сетевые технологии Оркестрация контейнеров и Kubernetes

Инструменты для ИИ и анализа данных

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Python (pandas, numpy, scikit-learn, TensorFlow, PyTorch)	Основной стек для обработки данных, машинного обучения и построения нейросетей.	BSD / MIT / Apache 2.0	Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps
Jupyter Notebook / Google Colab	Интерактивные среды для анализа данных, визуализации и прототипирования моделей.	BSD / Free	Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps

MLflow	Платформа для управления жизненным циклом моделей ML: отслеживание экспериментов, версионирование, развёртывание.	Apache 2.0	Методы искусственного интеллекта для разработки ПО и AIOps
Apache Spark	Фреймворк для распределённой обработки больших данных (в факультативе).	Apache 2.0	Управление данными и DataOps Машинное обучение и анализ данных
Библиотеки NLP (spaCy, Transformers)	Инструменты для обработки естественного языка, используемые при анализе документов (например, ЭПБ).	MIT / Apache 2.0	Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps
OpenCV	Библиотека компьютерного зрения для обработки изображений и видео (диагностика резервуаров, распознавание дефектов).	Apache 2.0 / BSD	Машинное обучение и анализ данных Методы искусственного интеллекта для разработки ПО и AIOps

Облачная инфраструктура

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Terraform	Описание и управление облачной инфраструктурой как кодом (IaC). justltd	Business Source License (HashiCorp), для OpenTofu – MPL 2.0 (см. ниже). justltd	DevOps-администрирование систем; Методы CI/CD и управление релизами; Управление данными и DataOps

Дополнительные инструменты

Инструмент / Технология	Назначение в программе	Лицензия	Используется в дисциплинах
Mockaroo / Faker / Mimesis	Генерация синтетических данных для тестирования и обучения моделей.	Freemium / MIT	Управление данными и DataOps Машинное обучение и анализ данных
Synthetic Data Vault (SDV)	Библиотека для генерации синтетических данных на основе реальных датасетов.	MIT	Управление данными и DataOps Машинное обучение и анализ данных
Chaos Mesh / Gremlin	Инструменты Chaos Engineering – проверка устойчивости системы к сбоям.	Apache 2.0 / коммерческая	DevOps-администрирование систем Оркестрация контейнеров и Kubernetes
ROS/ROS2, Gazebo	Фреймворк для разработки роботизированных систем и симуляции.	Apache 2.0 / BSD	Распределённые системы и микросервисная архитектура Проектирование информационных систем
SAP (OData API)	Интеграция с корпоративными системами (выгрузка заказ-нарядов в SAP TOPO).	Коммерческая	Web-разработка и API-интеграция

Перечень Приложений

1. **Приложение 1.** Целевая компетентностная модель выпускника (ЦКМВ) – полная модель с индикаторами и результатами обучения, а также расчёт показателей соответствия требованиям проекта.
2. **Приложение 2.** Матрица компетенций – распределение компетенций по дисциплинам и практикам.
3. **Приложение 3.** Паспорта профильных дисциплин – детальное описание каждой дисциплины (цели, пререквизиты, постреквизиты, формируемые индикаторы, артефакты).
4. **Приложение 4.** Паспорта практик – описание всех практик (задачи, компетенции, артефакты, форматы участия индустриальных партнёров, ИПС).
5. **Приложение 5.** Материалы взаимодействия с индустриальными партнёрами ООО «ГК «Иннотех» («якорный»), АО «Лаборатория Касперского» (методологический в вопросах разработки СКИБ) (протоколы, соглашения, технические задания, дорожная карта, договор о софинансировании).
6. **Приложение 6.** Положение о стажировках ООО «ГК «Иннотех»
7. **Приложение 7.** Положение о внутреннем отборе студентов на обучение по программе ТОП ИТ.
8. **Приложение 8.** Рецензия на ООП от индустриального партнера ООО «ГК «Иннотех».