Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ястребов Олег Александрович
Должность: Ректор
Дата подписания: 22.05.2026 14:55:10
Уникальный программный ключ:
ca953a012[illegible]

Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет дружбы народов имени Патриса Лумумбы»

Факультет искусственного интеллекта

(наименование основного учебного подразделения (ОУП)-разработчика ОП ВО)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

МЕТОДЫ РАЗРАБОТКИ РЕШЕНИЙ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (GIT, DOCKER)

(наименование дисциплины/модуля)

Рекомендована МССН для направлений подготовки:

02.03.02 ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ;
09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

(код и наименование направления подготовки/специальности)

Освоение дисциплины ведется в рамках реализации основной профессиональной образовательной программы высшего образования (ОП ВО):

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: РАЗРАБОТКА И ОБУЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

(наименование (профиль/специализация) ОП ВО)

2026 г.

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Методы разработки решений на основе искусственного интеллекта (Git, Docker)» входит в программу бакалавриата «Искусственный интеллект: разработка и обучение интеллектуальных систем» по направлениям подготовки 02.03.02 Фундаментальная информатика и информационные технологии и 09.03.03 Прикладная информатика, и изучается в 4 семестре 2 курса. Дисциплину реализует Кафедра прикладного искусственного интеллекта. Дисциплина состоит из 4 разделов и 34 тем и направлена на изучение инструментов и практик промышленной разработки программных решений в области искусственного интеллекта: системы управления версиями Git (ветвление, слияние, разрешение конфликтов, модели ветвления, код-ревью, pull requests), контейнеризации с помощью Docker (образы, контейнеры, Dockerfile, Docker Compose, реестры образов), автоматизации процессов сборки, тестирования и развёртывания (CI/CD) для проектов машинного обучения, версионирования данных и моделей, а также практик командной разработки ИИ-решений, включая документирование, планирование и организацию воспроизводимых рабочих окружений.

Целью освоения дисциплины является формирование у студентов практических навыков использования инструментов Git и Docker для организации командной разработки, обеспечения воспроизводимости экспериментов, создания контейнеризированных сред разработки и эксплуатации ИИ-систем, автоматизации тестирования и развёртывания ML-пайплайнов, а также навыков документирования процессов разработки, версионирования кода, данных и моделей, и планирования этапов проекта с учётом практик DevOps и MLOps.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Методы разработки решений на основе искусственного интеллекта (Git, Docker)» направлено на формирование у обучающихся следующих компетенций (части компетенций):

Таблица 2.1. Перечень компетенций, формируемых у обучающихся при освоении дисциплины (результаты освоения дисциплины)

Шифр	**Компетенция**	**Индикаторы достижения компетенции** (в рамках данной дисциплины)
УК-12	Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных	УК-12.1 Способен искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач;
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы	УК-2.2 Умеет анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
	их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	направления работ;
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	ОПК-3.3 Владеет навыками создания, тестирования и отладки алгоритмических и программных решений для систем ИИ, включая разработку пайплайнов обработки данных и обучения моделей;
ОПК-4	Способен участвовать в разработке технической документации, стандартов, норм и правил, а также в управлении проектами создания информационных систем и систем ИИ на стадиях жизненного цикла	ОПК-4.2 Умеет разрабатывать техническую документацию (ТЗ, описание архитектуры, пользовательскую документацию) для систем ИИ, планировать этапы проекта с учётом MLOps-практик;
ОПК-5	Способен инсталлировать и сопровождать программное и аппаратное обеспечение информационных систем и систем ИИ, в том числе отечественного происхождения, с учётом требований информационной безопасности	ОПК-5.2 Умеет развёртывать и сопровождать среды разработки и эксплуатации систем ИИ (контейнеризация, оркестрация, CI/CD), обеспечивать информационную безопасность данных и моделей;
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	ПК-1.2 Разрабатывает технические спецификации на программные компоненты систем ИИ и описывает их взаимодействие;
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и базы данных таких систем	ПК-2.1 Проектирует архитектуру ИС с компонентами ИИ, выбирает архитектурные паттерны и технологический стек;
ПК-3	Способен разрабатывать и реализовывать стратегии тестирования и контроля качества программного обеспечения систем ИИ	ПК-3.2 Разрабатывает план тестирования и организационные документы для тестирования ПО систем ИИ;
BD-2	Способен определять требования к наборам данных для решения задач машинного обучения, проводить разметку и анализ наборов данных, оценивать качество данных, обеспечивать непрерывную интеграцию данных	BD-2.3 Применяет инструменты и практики непрерывной интеграции данных (DataOps);
LC-5	Способен применять и (или) проектировать различные инструменты и инженерные практики промышленной разработки, развертывания,	LC-5.2 Осуществляет выбор инструментов и инженерных практик по управлению данными с необходимым уровнем доступа, контроля качества, резервирования и скоростью выполнения запросов;

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
	эксплуатации и мониторинга систем ИИ	
SS-2	Способен к эффективной коммуникации и командной работе в междисциплинарных проектах в области ИИ	SS-2.1 Эффективно коммуницирует с участниками проектной команды при планировании, реализации и анализе результатов работы в контексте гибридной команды "Человек+ИИ", включая постановку задач людям и ИИ-агентам, фиксацию договорённостей и критериев качества;

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Методы разработки решений на основе искусственного интеллекта (Git, Docker)» относится к обязательной части блока 1 «Дисциплины (модули)» образовательной программы высшего образования.

В рамках образовательной программы высшего образования обучающиеся также осваивают другие дисциплины и/или практики, способствующие достижению запланированных результатов освоения дисциплины «Методы разработки решений на основе искусственного интеллекта (Git, Docker)».

Таблица 3.1. Перечень компонентов ОП ВО, способствующих достижению запланированных результатов освоения дисциплины

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
УК-12	Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных	Программирование на языке Python; Введение в искусственный интеллект; Введение в базы данных; Статистические методы и первичный анализ данных; Технологическая (проектно-технологическая) практика (учебная);	*Вайб-кодинг***; Методы машинного обучения;
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	Правоведение; Этика и безопасность использования искусственного интеллекта;	*Оптимизация моделей машинного обучения;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;* *Проектирование и разработка систем компьютерного зрения;* *Практикум по обработке естественного языка (NLP);* *MLOps и промышленная разработка систем*

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
			искусственного интеллекта; *Технологическая (проектно-технологическая) практика (производственная);*
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	Дискретная математика; История и теория программирования; Алгоритмы и структуры данных; Введение в базы данных; Программирование на языке Python; Технологическая (проектно-технологическая) практика (учебная);	*Программирование на языке С++;* *Параллельное и распределенное программирование;* *Методы машинного обучения;* *Нейронные сети;* *Технологическая (проектно-технологическая) практика (производственная);*
ОПК-4	Способен участвовать в разработке технической документации, стандартов, норм и правил, а также в управлении проектами создания информационных систем и систем ИИ на стадиях жизненного цикла	Этика и безопасность использования искусственного интеллекта;	*Технологическая (проектно-технологическая) практика (производственная);* *Безопасность систем искусственного интеллекта;* *MLOps и промышленная разработка систем искусственного интеллекта;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;*
ОПК-5	Способен инсталлировать и сопровождать программное и аппаратное обеспечение информационных систем и систем ИИ, в том числе отечественного происхождения, с учётом требований информационной безопасности	Введение в базы данных; Технологическая (проектно-технологическая) практика (учебная);	*Безопасность систем искусственного интеллекта;* *MLOps и промышленная разработка систем искусственного интеллекта;* *Массово-параллельные вычисления в машинном обучении (GPU);* *Эксплуатационная практика (производственная);*
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	Технологическая (проектно-технологическая) практика (учебная); Правоведение; Введение в искусственный интеллект; Искусственный интеллект и когнитивная психология; Этика и безопасность использования искусственного интеллекта; История и теория программирования; Введение в базы данных;	*Эксплуатационная практика (производственная);* *Преддипломная практика;* *Технологическая (проектно-технологическая) практика (производственная);* *Параллельное и распределенное программирование;* *Методы машинного обучения;* *Массово-параллельные вычисления в машинном обучении (GPU);* *Оптимизация моделей машинного обучения;*

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
			Основы глубокого обучения; *Безопасность систем искусственного интеллекта;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;* *Большие языковые модели**;* *Программирование на языке C++;* *MLOps и промышленная разработка систем искусственного интеллекта;* *Нейронные сети;* *Онтология и графы знаний;* *Проектирование и разработка систем компьютерного зрения;* *Практикум по обработке естественного языка (NLP);*
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и базы данных таких систем	Алгоритмы и структуры данных; Программирование на языке Python; Введение в базы данных; Технологическая (проектно-технологическая) практика (учебная);	*Программирование на языке C++;* *Параллельное и распределенное программирование;* *Hadoop, SPARK;* *Массово-параллельные вычисления в машинном обучении (GPU);* *MLOps и промышленная разработка систем искусственного интеллекта;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;* *Проектирование и разработка систем компьютерного зрения;* *Практикум по обработке естественного языка (NLP);* *Основы глубокого обучения;* *Вайб-кодинг**;* *Онтология и графы знаний;* *Эксплуатационная практика (производственная);* *Преддипломная практика;* *Технологическая (проектно-технологическая) практика (производственная);*
ПК-3	Способен разрабатывать и реализовывать стратегии тестирования и контроля качества программного обеспечения систем ИИ	Технологическая (проектно-технологическая) практика (учебная); Теория вероятностей и математическая статистика; Этика и безопасность использования искусственного	*Преддипломная практика;* *Технологическая (проектно-технологическая) практика (производственная);* *Эксплуатационная практика (производственная);*

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
		интеллекта; Статистические методы и первичный анализ данных; Программирование на языке Python;	*Методы машинного обучения;* *Нейронные сети;* *Безопасность систем искусственного интеллекта;* *Обработка и анализ изображений и видео с помощью методов искусственного интеллекта;* *Анализ естественного языка с помощью методов искусственного интеллекта;* *MLOps и промышленная разработка систем искусственного интеллекта;* *Проектирование и разработка систем компьютерного зрения;* *Практикум по обработке естественного языка (NLP);* *Оптимизация моделей машинного обучения;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;*
SS-2	Способен к эффективной коммуникации и командной работе в междисциплинарных проектах в области ИИ	Технологическая (проектно-технологическая) практика (учебная); Программирование на языке Python; *Иностранный язык***; *Русский язык (как иностранный)***;	*Эксплуатационная практика (производственная);* *Технологическая (проектно-технологическая) практика (производственная);* *MLOps и промышленная разработка систем искусственного интеллекта;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;* *Проектирование и разработка систем компьютерного зрения;* *Практикум по обработке естественного языка (NLP);* *Большие языковые модели**;* *Вайб-кодинг**;* *Иностранный язык в профессиональной деятельности**;* *Русский язык (как иностранный) в профессиональной деятельности**;*
BD-2	Способен определять требования к наборам данных для решения задач	Технологическая (проектно-технологическая) практика (учебная);	*Эксплуатационная практика (производственная);*

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
	машинного обучения, проводить разметку и анализ наборов данных, оценивать качество данных, обеспечивать непрерывную интеграцию данных	Статистические методы и первичный анализ данных; Введение в базы данных;	*Методы машинного обучения;* *Практическая подготовка на проектах отраслевых индустриальных партнеров;* *MLOps и промышленная разработка систем искусственного интеллекта;*
LC-5	Способен применять и (или) проектировать различные инструменты и инженерные практики промышленной разработки, развертывания, эксплуатации и мониторинга систем ИИ		*Технологическая (проектно-технологическая) практика (производственная);* *Преддипломная практика;* *MLOps и промышленная разработка систем искусственного интеллекта;*

* - заполняется в соответствии с матрицей компетенций и СУП ОП ВО

** - элективные дисциплины /практики

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Методы разработки решений на основе искусственного интеллекта (Git, Docker)» составляет «4» зачетные единицы.

Таблица 4.1. Виды учебной работы по периодам освоения образовательной программы высшего образования для очной формы обучения.

Вид учебной работы	**ВСЕГО,** ак.ч.		**Семестр(-ы)**
			4
Контактная работа, ак.ч.	*68*		*68*
Лекции (**ЛК**)	17		17
Лабораторные работы (**ЛР**)	0		0
Практические/семинарские занятия (**СЗ**)	51		51
Самостоятельная работа обучающихся, ак.ч.	*49*		*49*
Контроль (экзамен/зачет с оценкой), ак.ч.	*27*		*27*
Общая трудоемкость дисциплины	**ак.ч.**	***144***	*144*
	зач.ед.	***4***	*4*

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Таблица 5.1. Содержание дисциплины (модуля) по видам учебной работы

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
Раздел 1	Система управления версиями Git	1.1	Введение в управление версиями. Архитектура Git	Проблемы разработки без контроля версий: потеря кода, невоспроизводимость, конфликты. Централизованные (SVN) и распределённые (Git) системы. Архитектура Git: рабочая копия, индекс (staging area), локальный репозиторий, удалённый репозиторий. Объектная модель Git: blob, tree, commit, tag. Установка и настройка Git (git config)	ЛК	ОПК-3.3, УК-12.1
		1.2	Модели ветвления и командная работа в Git	Ветвление: создание, переключение, удаление. Стратегии ветвления: Git Flow, GitHub Flow, trunk-based development. Слияние (merge) и перебазирование (rebase): различия, когда использовать. Разрешение конфликтов. Pull requests и код-ревью: назначение, процесс, лучшие практики. Защита веток (branch protection rules)	ЛК	ОПК-3.3, SS-2.1, ОПК-4.2
		1.3	Практикум: базовые операции Git	Инициализация репозитория (git init). Цикл: add → commit → status → log → diff. Работа с .gitignore: исключение данных, моделей, окружений из трекинга. Откат изменений: git restore, git reset, git revert. Просмотр истории: git log, git show, git blame. Работа с тегами: маркировка релизов	СЗ	ОПК-3.3, УК-12.1
		1.4	Практикум: ветвление и слияние	Создание feature-ветки. Работа в параллельных ветках. Слияние через merge: fast-forward и three-way merge. Перебазирование (rebase): линейная история. Разрешение конфликтов вручную и с помощью инструментов (VS Code, meld). Практика на учебном ML-проекте	СЗ	ОПК-3.3, SS-2.1
		1.5	Практикум: работа с удалённым репозиторием (GitHub/GitLab)	Создание удалённого репозитория. git clone, push, pull, fetch. Настройка SSH-ключей. Fork и upstream. Создание pull request: описание изменений, назначение ревьюера, обсуждение. Код-ревью: чек-лист качества, комментарии, approve/request changes	СЗ	ОПК-3.3, SS-2.1, ОПК-4.2
		1.6	Практикум: командная работа с Git	Моделирование командной разработки: два-три человека работают над общим репозиторием. Распределение задач через Issues. Создание feature-веток, pull requests, код-ревью, слияние. Разрешение merge-конфликтов в команде.	СЗ	SS-2.1, ОПК-4.2, УК-2.2

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				Соглашения об именовании коммитов (Conventional Commits)		
		1.7	Практикум: документирование проекта в Git	Написание README.md: назначение проекта, установка, использование, структура, лицензия. Ведение CHANGELOG.md. Шаблоны Issues и Pull Requests. Описание архитектуры в docs/. Связь документирования в Git с разработкой технической документации на программные компоненты	СЗ	ОПК-4.2, ПК-1.2
		1.8	Практикум: версионирование данных и моделей (DVC)	Проблема: данные и модели не помещаются в Git. Инструмент DVC (Data Version Control): установка, добавление данных, remote storage (S3, GCS, локальная папка). Связь .dvc-файлов с git-коммитами. Переключение между версиями данных. Воспроизводимость ML-экспериментов	СЗ	BD-2.3, ОПК-3.3, LC-5.2
		1.9	Практикум: Git для ML-проекта — сквозная задача	Организация ML-проекта: структура каталогов (data/, src/, models/, notebooks/, tests/, configs/). Cookiecutter Data Science. Версионирование кода (Git) и данных (DVC). Файл requirements.txt / pyproject.toml. Запуск эксперимента из командной строки. Фиксация результата в коммите	СЗ	ОПК-3.3, BD-2.3, ПК-2.1
Раздел 2	Контейнеризация с Docker	2.1	Основы контейнеризации. Архитектура Docker	Проблема «у меня работает»: зависимости, версии, окружения. Контейнеризация vs. виртуализация: различия, преимущества. Архитектура Docker: Docker Engine, образы (images), контейнеры, реестры (Docker Hub, GitHub Container Registry). Слои образа (layers). Пространства имён (namespaces) и контрольные группы (cgroups)	ЛК	ОПК-5.2, УК-12.1
		2.2	Dockerfile и Docker Compose	Dockerfile: инструкции FROM, RUN, COPY, WORKDIR, ENV, EXPOSE, CMD, ENTRYPOINT. Многоэтапная сборка (multi-stage build). Оптимизация: порядок слоёв, кэширование, минимизация размера образа. Docker Compose: определение многоконтейнерных приложений, файл docker-compose.yml, сервисы, сети, тома (volumes)	ЛК	ОПК-5.2, ПК-2.1
		2.3	Практикум: первые контейнеры	Установка Docker. Команды: docker pull, run, ps, stop, rm, images, rmi. Запуск контейнера из готового образа (python, ubuntu). Интерактивный режим (-it). Проброс портов (-p). Монтирование томов (-v). Переменные окружения (-e)	СЗ	ОПК-5.2, УК-12.1

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
		2.4	Практикум: написание Dockerfile для Python-проекта	Создание Dockerfile для ML-проекта: базовый образ python:3.11-slim, установка зависимостей (requirements.txt), копирование кода, определение точки входа. Сборка образа (docker build). Запуск контейнера. Оптимизация: порядок COPY, использование .dockerignore	СЗ	ОПК-5.2, ОПК-3.3
		2.5	Практикум: многоэтапная сборка и оптимизация	Многоэтапная сборка: этап сборки (установка зависимостей, компиляция) и этап выполнения (минимальный образ). Уменьшение размера образа: slim/alpine базы, удаление кэша pip. Анализ слоёв (docker history). Публикация образа в Docker Hub / GitHub Container Registry	СЗ	ОПК-5.2, ПК-2.1
		2.6	Практикум: Docker Compose — многоконтейнерное приложение	Создание docker-compose.yml для ML-сервиса: контейнер приложения (FastAPI), контейнер базы данных (PostgreSQL), контейнер мониторинга (Prometheus, обзор). Определение сетей, томов, зависимостей (depends_on). Запуск и остановка (docker compose up/down)	СЗ	ОПК-5.2, ПК-2.1
		2.7	Практикум: контейнеризация Jupyter-окружения для ML	Создание Docker-образа с Jupyter Notebook/Lab, установленными ML-библиотеками (NumPy, Pandas, scikit-learn, PyTorch). Проброс порта для доступа к Jupyter. Монтирование локальной папки с ноутбуками. Воспроизводимость: любой участник команды получает идентичное окружение	СЗ	ОПК-5.2, ОПК-3.3, LC-5.2
		2.8	Практикум: контейнеризация ML-сервиса инференса	Обучение простой модели (scikit-learn), сериализация (joblib/pickle). Создание FastAPI-сервиса предсказаний. Контейнеризация сервиса: Dockerfile, health check. Тестирование API через curl / httpie. Документирование API (OpenAPI/Swagger, автогенерация)	СЗ	ОПК-5.2, ОПК-3.3, ПК-1.2
		2.9	Практикум: Docker для GPU-вычислений (обзор	NVIDIA Container Toolkit: доступ к GPU из контейнера. Базовые образы nvidia/cuda. Запуск контейнера с --gpus all. Пример: контейнер с PyTorch + CUDA для обучения модели. Ограничения и лучшие практики. Связь с инфраструктурой ИИ-разработки	СЗ	ОПК-5.2, ПК-2.1
Раздел 3	CI/CD для проектов машинного обучения	3.1	Принципы непрерывной интеграции и доставки (CI/CD)	Непрерывная интеграция (CI): автоматическая сборка и тестирование при каждом коммите. Непрерывная доставка (CD): автоматическое развёртывание. Пайплайн CI/CD:	ЛК	ОПК-3.3, ПК-3.2

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				этапы (lint, test, build, deploy). Инструменты: GitHub Actions, GitLab CI/CD, Jenkins (обзор). Связь CI/CD с качеством ПО и скоростью разработки		
		3.2	CI/CD для ML: от кода к модели	Особенности CI/CD для ML-проектов: тестирование кода + тестирование данных + тестирование модели. Уровни тестирования: линтинг (flake8, black), юнит-тесты (pytest), интеграционные тесты. Тестирование данных: проверка схемы, наличие пропусков, распределения. Тестирование модели: метрики на контрольной выборке, регрессионные тесты	ЛК	ПК-3.2, ОПК-3.3, LC-5.2
		3.3	Практикум: GitHub Actions — первый пайплайн	Структура workflow-файла (.github/workflows/). Триггеры: push, pull_request, schedule. Jobs и steps. Использование готовых actions: checkout, setup-python. Создание пайплайна: установка зависимостей → линтинг (flake8) → запуск тестов (pytest). Просмотр логов в GitHub	СЗ	ОПК-3.3, ПК-3.2
		3.4	Практикум: расширенный CI-пайплайн для ML-проекта	Добавление этапов: проверка форматирования (black --check), проверка типов (mypy, обзор), тестирование данных (great_expectations, обзор). Матричная сборка: тестирование на нескольких версиях Python. Кэширование зависимостей для ускорения. Секреты (GitHub Secrets)	СЗ	ПК-3.2, ОПК-3.3
		3.5	Практикум: автоматическая сборка и публикация Docker-образа	CI-пайплайн: при merge в main автоматически собирается Docker-образ и публикуется в GitHub Container Registry. Тегирование образов (latest, версия из git tag). Использование docker/build-push-action. Связь с практикой развёртывания ML-сервисов	СЗ	ОПК-5.2, ПК-3.2, ПК-2.1
		3.6	Практикум: разработка плана тестирования ML-проекта	Составление плана тестирования для ML-проекта: уровни (unit, integration, system), объекты (код, данные, модель), инструменты, критерии прохождения. Разработка чек-листа приёмки ML-компонента. Документирование в формате Markdown в репозитории. Связь с организационными документами тестирования ПО	СЗ	ПК-3.2, ОПК-4.2, ПК-1.2
		3.7	Практикум: автоматизация проверки качества модели в CI	Добавление этапа CI: обучение модели на маленьком подмножестве данных → вычисление метрик → сравнение с порогом (accuracy > 0.8). Сохранение метрик как артефактов пайплайна. Комментирование PR с	СЗ	ПК-3.2, ОПК-3.3, LC-5.2

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				результатами метрик (GitHub Actions bot). Обсуждение: когда модель «готова»?		
		3.8	Практикум: GitLab CI/CD (обзор) и сравнение инструментов	Файл .gitlab-ci.yml: stages, jobs, artifacts, cache. Сравнение GitHub Actions и GitLab CI/CD: синтаксис, возможности, ограничения. Обзор отечественных платформ: Gitflic, Gitverse. Выбор инструмента CI/CD для проекта: критерии	СЗ	ОПК-3.3, ПК-3.2, УК-12.1
Раздел 4	Интеграция инструментов и DevOps-практики для ИИ-проектов	4.1	Организация командной разработки ИИ-проекта	DevOps для ML: культура, практики, инструменты. Agile и Scrum в ML-проектах: спринты, backlog, ретроспективы. Трекинг задач: GitHub Issues, GitHub Projects, Kanban-доска. Распределение ролей в команде. Планирование спринта для ML-проекта. Матрица RACI для ИИ-проекта	ЛК	УК-2.2, SS-2.1, ОПК-4.2
		4.2	Инфраструктура как код и оркестрация	Принцип Infrastructure as Code (IaC): воспроизводимость окружений. Docker Compose для локальной оркестрации. Kubernetes: основные концепции (pod, service, deployment, namespace) — обзор. Helm charts — обзор. Облачные платформы для ML: Yandex Cloud, AWS SageMaker (обзор). Связь с масштабированием ИИ-систем	ЛК	ОПК-5.2, ПК-2.1
		4.3	Практикум: планирование спринта для ML-проекта	Формулирование User Stories для ML-проекта. Декомпозиция на задачи (Issues). Оценка трудоёмкости. Создание Kanban-доски в GitHub Projects. Распределение задач в команде. Фиксация Definition of Done для ML-задач (код, тесты, документация, код-ревью)	СЗ	УК-2.2, SS-2.1, ОПК-4.2
		4.4	Практикум: составление технической спецификации ML-сервиса	Описание архитектуры: компоненты (API, модель, БД, мониторинг), их взаимодействие. Спецификация API (OpenAPI). Описание Docker-контейнеров и их конфигурации. Требования к ресурсам. Документирование в формате Architecture Decision Records (ADR)	СЗ	ПК-1.2, ПК-2.1, ОПК-4.2
		4.5	Практикум: pre-commit hooks и автоматизация качества кода	Установка и настройка pre-commit: конфигурация .pre-commit-config.yaml. Хуки: black (форматирование), flake8 (линтинг), isort (сортировка импортов), mypy (типы, обзор), detect-secrets (обнаружение секретов). Автоматическое применение при каждом коммите	СЗ	ОПК-3.3, ПК-3.2
		4.6	Практикум: мониторинг контейнеров и логирование	Логирование в контейнерах: docker logs, структурированное логирование (JSON). Мониторинг: проверка здоровья (health checks в docker-compose). Обзор инструментов:	СЗ	ОПК-5.2, LC-5.2

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				Prometheus + Grafana для мониторинга метрик сервиса. Связь с мониторингом ML-моделей в продакшене		
		4.7	Практикум: воспроизводимость ML-эксперимента — сквозная демонстрация	Сквозная задача: клонирование репозитория → установка DVC → загрузка данных (dvc pull) → запуск Docker-контейнера → обучение модели → фиксация результатов → коммит + push. Демонстрация: другой участник команды воспроизводит эксперимент идентично	СЗ	BD-2.3, ОПК-5.2, ОПК-3.3
		4.8	Практикум: итоговый мини-проект — полный DevOps-пайплайн для ML	Финальная интеграция: Git-репозиторий с структурой проекта, DVC для данных, Dockerfile и docker-compose для окружения, CI/CD-пайплайн (lint → test → build → publish), README и техническая документация, план тестирования. Командная работа: распределение задач через Issues, код-ревью через PR. Презентация результатов	СЗ	ОПК-3.3, ОПК-5.2, ПК-1.2, ПК-2.1, SS-2.1

* - заполняется только по **ОЧНОЙ** форме обучения: *ЛК – лекции; ЛР – лабораторные работы; СЗ – практические/семинарские занятия.*

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Таблица 6.1. Материально-техническое обеспечение дисциплины

Тип аудитории	**Оснащение аудитории**	**Специализированное учебное/лабораторное оборудование, ПО и материалы для освоения дисциплины (при необходимости)**
Лекционная	Аудитория для проведения занятий лекционного типа, оснащенная комплектом специализированной мебели; доской (экраном) и техническими средствами мультимедиа презентаций.	
Семинарская	Аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная комплектом специализированной мебели и техническими средствами мультимедиа презентаций.	Персональные компьютеры, необходимое ПО
Для самостоятельной работы	Аудитория для самостоятельной работы обучающихся (может использоваться для проведения семинарских занятий и консультаций), оснащенная комплектом специализированной мебели и компьютерами с доступом в ЭИОС.	Персональные компьютеры, необходимое ПО

* - аудитория для самостоятельной работы обучающихся указывается **ОБЯЗАТЕЛЬНО**!

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

1. Кейн, Ш. П. Docker. Вводный курс / Ш. П. Кейн, К. Маттиас ; пер. с англ. — 3-е изд. — Астана : Алист, 2025. — 352 с. — ISBN 978-601-09-7541-5.

2. Scott Chacon, Ben Straub. Pro Git / Версия 2.1.121-1-gabdb461, 27.01.2026 – URL: https://git-scm.com/book/ru/v2

Дополнительная литература:

1. Учебник Git для новичков. – URL: https://code.mu/ru/git/book/prime/?ysclid=mogy45i7l7646604718#

Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. ЭБС РУДН и сторонние ЭБС, к которым студенты университета имеют доступ на основании заключенных договоров

- Электронно-библиотечная система РУДН – ЭБС РУДН https://mega.rudn.ru/MegaPro/Web
- ЭБС «Университетская библиотека онлайн» http://www.biblioclub.ru
- ЭБС «Юрайт» http://www.biblio-online.ru
- ЭБС «Консультант студента» www.studentlibrary.ru
- ЭБС «Знаниум» https://znanium.ru/

2. Базы данных и поисковые системы

- Sage https://journals.sagepub.com/
- Springer Nature Link https://link.springer.com/
- Wiley Journal Database https://onlinelibrary.wiley.com/
- Наукометрическая база данных Lens.org https://www.lens.org

Учебно-методические материалы для самостоятельной работы обучающихся при освоении дисциплины/модуля:*

1. Курс лекций по дисциплине «Методы разработки решений на основе искусственного интеллекта (Git, Docker)».

* - все учебно-методические материалы для самостоятельной работы обучающихся размещаются в соответствии с действующим порядком на странице дисциплины **в ТУИС**!