МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ – МСХА имени К.А. ТИМИРЯЗЕВА»

(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)

Институт экономики и управления АПК
Кафедра прикладной информатики

УТВЕРЖДАЮ:
Директор института
экономики и управления АПК
_____ Л.И. Хоружий
“__28__”_____08_____2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.11 API-технологии

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.03 Прикладная информатика
Направленность: Системы искусственного интеллекта

Курс 2
Семестр 3

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчик (и): Лапшин М.С., ассистент
(ФИО, ученая степень, ученое звание) (подпись)

(ФИО, ученая степень, ученое звание) (подпись)

« 28 » августа 2025 г.

Рецензент: Ивашова О.Н., к.с.-х.н., доцент
(ФИО, ученая степень, ученое звание) (подпись)

« 28 » августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО, профессионального стандарта и учебного плана по направлению подготовки 09.03.03 «Прикладная информатика»

Программа обсуждена на заседании кафедры прикладной информатики
протокол №1 от « 28 » августа 2025 г.

И.о. заведующего кафедрой
прикладной информатики Худякова Е.В., д.э.н., профессор
(ФИО, ученая степень, ученое звание) (подпись)

« 28 » августа 2025 г.

Согласовано:

Председатель учебно-методической комиссии
института экономики и управления АПК
Гупалова Т.Н., к.э.н., доцент
(ФИО, ученая степень, ученое звание) (подпись)

« 28 » августа 2025 г.

И.о. заведующего выпускающей кафедрой
прикладной информатики Худякова Е.В., д.э.н., профессор
(ФИО, ученая степень, ученое звание) (подпись)

« 28 » августа 2025 г.

Заведующий отделом комплектования ЦНБ
(подпись)

СОДЕРЖАНИЕ

Аннотация

рабочей программы учебной дисциплины
Б1.В.11 «API-технологии»
для подготовки бакалавров по направлению 09.03.03 «Прикладная информатика», направленность «Системы искусственного интеллекта»

Цель освоения дисциплины: сформировать у обучающихся компетенции по проектированию, разработке, документированию, тестированию и сопровождению API для программных систем на основе ИИ, обеспечивая интеграцию AI/ML-компонентов (обучение/инференс, сервисы данных, внешние платформы) и высокую производительность решений за счёт применения **C/C++**, многопоточности и оптимизации под целевые аппаратно-программные платформы.

Место дисциплины в учебном плане: дисциплина включена в формируемую участниками образовательных отношений часть учебного плана по направлению подготовки09.03.03 «Прикладная информатика».

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы) их достижения: ПК-16 (PL-3).1. ПК-16 (PL-3).2. ПК-16 (PL-3).3.

Краткое содержание дисциплины: Дисциплина посвящена проектированию и разработке API-интерфейсов и сервисов для компонентов искусственного интеллекта, включая построение контрактов взаимодействия, обмен данными и интеграцию AI/ML-модулей в прикладные системы. Рассматриваются основы клиент–серверного взаимодействия и протоколы передачи данных, подходы к проектированию REST/gRPC API, версионирование и документирование интерфейсов, валидация данных и обработка ошибок, а также обеспечение безопасности (аутентификация/авторизация, защита API). Изучаются практики тестирования API (модульное, интеграционное, контрактное), мониторинга и трассировки, организация CI/CD для сервисов. Отдельное внимание уделяется производительности и оптимизации API-сервисов в контексте ИИ: многопоточность и распараллеливание, снижение латентности инференса (батчинг, кэширование, эффективная сериализация), профилирование и нагрузочное тестирование, оптимизация под аппаратно-программные платформы и ускорители, анализ результатов и сравнение конфигураций. Практическая часть ориентирована на разработку и сопровождение прототипа API-сервиса для инференса/обработки данных с подготовкой спецификации, тестов и отчёта по производительности.

Общая трудоемкость дисциплины/в т.ч. практическая подготовка: 108/4 (часы/зач. ед.)

Промежуточный контроль: зачет.

1. Цель освоения дисциплины

Целью освоения дисциплины «API-технологии» является формирование у обучающихся компетенций, обеспечивающих способность к проектированию, реализации и сопровождению API-интерфейсов и сервисов взаимодействия ком-

понентов искусственного интеллекта, включая разработку высокопроизводительных программных модулей на языке C/C++ в различных парадигмах программирования, применение средств распараллеливания и многопоточности при обработке запросов и выполнении вычислений, оптимизацию программного обеспечения под различные аппаратно-программные платформы (в том числе с использованием ускорителей), а также анализ и интерпретацию результатов профилирования и нагрузочного тестирования при решении задач ИИ.

Значимость формирования цифровых и алгоритмических компетенций в процессе профессиональной подготовки специалистов в области ИТ обусловлена требованиями цифровой трансформации экономики, а также приоритетами государственной политики Российской Федерации в области внедрения искусственного интеллекта и интеллектуальных технологий. Освоение дисциплины направлено на обеспечение готовности выпускников к практико-ориентированной деятельности в условиях цифрового общества и глобализированных рынков данных.

Дисциплина реализуется в контексте участия образовательных программ в федеральной инициативе подготовки топ-специалистов в области информационных технологий и искусственного интеллекта (ТОП ИИ), осуществляемой с 2025 года в рамках федеральных проектов «Искусственный интеллект» и «Кадры для цифровой трансформации» национального проекта «Экономика данных и цифровая трансформация государства». Программа ориентирована на развитие у студентов продвинутых компетенций в области проектирования и внедрения ИТ-решений и ИИ-моделей, в тесной связи с индустриальными партнёрами, участвующими в образовательном процессе, в том числе в формате проектной деятельности и софинансирования.

2. Место дисциплины в учебном процессе

Дисциплина «API-технологии» относится к части, формируемой участниками образовательных отношений, Блока 1 «Дисциплины (модули)» учебного плана и реализуется в 3 семестре в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению подготовки 09.03.03 «Прикладная информатика», направленность (профиль) «Системы искусственного интеллекта» (уровень образования — бакалавриат).

Содержательно дисциплина занимает важное место в подготовке бакалавра, поскольку обеспечивает переход от базовых знаний программирования, математического аппарата и основ ИИ к инженерной практике построения прикладных API-сервисов в составе ИИ-систем: проектирование контрактов и интерфейсов (REST/gRPC), разработка высокопроизводительных сервисов на C++, организация конкурентной обработки запросов, нагрузочное тестирование и профилирование, а также адаптация решений под ограничения целевых платформ (в том числе встраиваемых) и использование аппаратных ускорителей (GPU/FPGA). Новизна и значимость дисциплины в учебном процессе определяется её практико-ориентированной направленностью на разработку и эксплуатацию API-компонентов ИИ-решений с измеримыми требованиями к качеству (задержка, пропускная способность, устойчивость), что непосредственно соответствует профилю «Системы искусственного интеллекта».

Предшествующими курсами, на которых непосредственно базируется дисциплина «API-технологии», являются: «Программирование на языке Python» (Б1.В.20.03), «Математическая статистика» (Б1.О.07), «Линейная алгебра» (Б1.О.22), «Теоретические основы информатики» (Б1.О.23), «Основы ИИ в АПК» (Б1.В.21). Дисциплина изучается параллельно с дисциплинами «Технологии обработки больших данных в АПК» (Б1.В.13) и «Базы данных» (Б1.О.20.02), что обеспечивает связность формирования компетенций по построению ИИ-сервисов, работающих с данными и инфраструктурой хранения.

Дисциплина «API-технологии» является основополагающей для изучения следующих дисциплин и практик: «Информационные системы и технологии» (Б1.О.20.03), «ИТ-инфраструктура организации АПК» (Б1.В.03), «Управление информационными системами в АПК» (Б1.В.04), «Разработка геоинформационных систем для предприятий АПК» (Б1.В.05), «Разработка распределенных систем» (Б1.В.09), «Программирование в 1С» (Б1.В.10), «Технологии работы с открытыми данными» (Б1.В.14), «Компьютерное зрение» (Б1.В.15), «AIoT-технологии и средства автоматизации в АПК» (Б1.В.16), «Машинное обучение» (Б1.В.17), «Анализ пространственно-временных данных на основе машинного обучения» (Б1.В.18), «Глубокое обучение» (Б1.В.19), «Средства работы в команде» (Б1.В.ДВ.02.01).

Рабочая программа дисциплины «API-технологии» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 108/4 часов, их распределение по видам работ семестрам представлено в таблице 2.

Таблица 1

Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся осваивают следующий уровень:		
				знать	уметь	владеть
	ПК-16 (PL-3)	Способен применять языки программирования C/C++ для решения задач в области ИИ	ПК-16 (PL-3).1 Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений Уровень: Продвинутый Решает проблемы одновременного доступа к данным из нескольких потоков, грамотно применяет атомарные операции и механизм блокировок. Оценивает производительность, умеет профилировать код и устраняет найденные узкие места.	основы построения многопоточных API-сервисов на C++ (жизненный цикл запроса, типовые точки гонок в API: кэш, очереди задач, пул соединений, счётчики/метрики), модели памяти и причины data race/deadlock, средства синхронизации и конкурентности C++17/20 (std::mutex/std::shared_mutex, std::condition_variable, std::atomic, std::future), а также базовые метрики и инструменты оценки качества API (p95/p99 latency, throughput; perf/VTune, ThreadSanitizer/AddressSanitizer).		

			ПК-16 (PL-3).2 Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем Уровень: Продвинутый Уровень освоения индикатора: Понимает методы оптимизации моделей (квантование, сжатие весов модели и пр.) и вычислений ИИ. Находит и использует библиотеки, соответствующие решаемой задаче		анализировать ограничения целевой платформы (CPU/RAM/энергопотребление, наличие GPU/NPU, ОС/рантайм) и под них выбирать стек инференса на C++ для ИИ-сервиса/модуля с API (например, ONNX Runtime, OpenVINO, TensorRT, TFLite), настраивать сборку и развертывание (в т.ч. кросс-компиляцию для embedded). Уметь применять базовые методы ускорения и “облегчения” моделей и вычислений (квантование int8/FP16, pruning/сжатие, оптимизация графа, батчинг, оптимизация препроцессинга/постпроцессинга, SIMD/NEON/AVX при необходимости) и проверять эффект по метрикам latency/throughput/память, профилируя и устраняя узкие места на выбранной платформе.	
			ПК-16 (PL-3).3 Разрабатывает и отлаживает решения на C++, использующие GPU и FPGA для массовой параллелизации вычислений в рамках общей системы ИИ, с применением как готовых решений, так и разработкой своих Уровень: Продвинутый Уровень освоения индикатора: Знает методы оптимизации моделей (квантование, сжатие весов модели и пр.) и вычислений ИИ.			практиками разработки и интеграции GPU/FPGA-ускорения в составе API/сервисов ИИ на C++ (выделение вычислительных ядер инференса/препроцессинга, организация массово параллельных вычислений, управление передачей данных CPU-GPU/FPGA и минимизация копирований). Владеть готовыми средствами

			Владеет готовыми инструментами для оптимизации моделей (TensorRT и пр.). Умеет использовать средства отладки и профилирования кода, находить участки кода, ограничивающие производительность системы			оптимизации и развертывания моделей на ускорителях (например, TensorRT для NVIDIA GPU, а также OpenVINO/ONNX Runtime с провайдерами ускорения — по применяемому стеку), включая настройку режимов точности (FP16/INT8), калибровку и проверку качества/скорости. Владеть инструментами отладки и профилирования производительности, позволяющими выявлять узкие места в AI-системе и API-контуре (NVIDIA Nsight Systems/Compute, профилировщики CPU, анализ контеншена и копирований, трассировка задержек), и методикой интерпретации результатов для принятия инженерных решений по ускорению.

Таблица 2а

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость	
	час. всего/*	В т.ч. по семестрам №3
Общая трудоёмкость дисциплины по учебному плану	108/4	108/4
1. Контактная работа:	54,25/4	54,25/4
Аудиторная работа	54,25/4	54,25/4
в том числе:		
лекции (Л)	18	18
практические занятия (ПЗ)	36/4	36/4
лабораторные работы (ЛР)	0	0
курсовая работа (проект) (КР/КП) (консультация, защита)	0	0
консультации перед экзаменом	0	0
контактная работа на промежуточном контроле (КРА)	0,25	0,25
2. Самостоятельная работа (СРС)	53,75	53,75
реферат/эссе (подготовка)	0	0
курсовая работа/проект (КР/КП) (подготовка)	0	0
расчётно-графическая работа (РГР) (подготовка)	0	0
контрольная работа	0	0
самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)	46,75	46,75
Подготовка к экзамену (контроль)	0	0
Подготовка к зачёту/ зачёту с оценкой (контроль)	0	0
Вид промежуточного контроля:	зачет	

* в том числе практическая подготовка.(см учебный план)

4.2 Содержание дисциплины

Таблица 3а

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа				Внеаудиторная работа СР
		Л	ПЗ/С всего/*	ЛР всего/*	ПК-Р	
Раздел 1 «Основы API в ИИ-системах»	20	4	8	-	-	8
Раздел 2 «Реализация API на C++ и интеграция инференса»	24	4	8	-	-	12
Раздел 3 «Высокопроизводительные API-сервисы для ИИ»	26	4	8/2	-	-	14
Раздел 4. «Оптимизация под платформы и ускорители»	24	4	8/2	-	-	12
Раздел 5. «Качество и эксплуатация API ИИ-сервисов»	13,75	2	4	-	-	7,75
Всего за 3 семестр	108	18	36/4	0	0,25	53,75
Итого по дисциплине	108	18	36/4	0	0,25	53,75

* в том числе практическая подготовка

Раздел 1. Основы API в ИИ-системах

Тема 1. Введение в API-технологии для ИИ: роль API, типовые архитектуры, требования (латентность, throughput, безопасность)

Роль API в жизненном цикле ИИ-систем (обучение/инференс/данные). Типовые архитектуры: монолит, микросервисы, API gateway, inference-service. Синхронные/асинхронные вызовы, очереди, event-driven. Ключевые требования: задержка, пропускная способность, стабильность, стоимость. Базовые риски и требования безопасности для API ИИ (доступ к модели/данным, лимиты). Обзор интерфейсов для инференса: REST/gRPC, streaming, batch.

Тема 2. Проектирование API-контрактов для ИИ-сервисов: REST/gRPC, модели данных, ошибки, версионирование.

Принципы проектирования ресурсов/методов и RPC-методов под ИИ-сценарии. Модели данных запросов/ответов (входы модели, параметры, метаданные, результаты). Схема и валидация: обязательные поля, ограничения, форматы. Обработка ошибок и коды ответов, классификация ошибок (клиент/сервер/модель). Версионирование API и моделей (API v1/v2, model_version). Идемпотентность, пагинация/фильтры для сервисов данных. Документация контракта (OpenAPI/Proto) и примеры.

Раздел 2. Реализация API на C++ и интеграция инференса

Тема 3. Реализация API на C++: HTTP (Boost.Asio/Beast) и/или gRPC C++, сериализация JSON/Protobuf.

Структура C++ API-сервиса: роутинг/handlers, middleware, конфигурация. Реализация HTTP endpoint'ов на Boost.Asio/Beast и/или RPC на gRPC C++. Сериализация/десериализация JSON и Protobuf, контроль типов и схем. Валидация входных данных и нормализация параметров. Единый формат ошибок API и обработка исключений. Логирование и корреляция запросов (request-id). Базовые меры защиты на уровне API (TLS, ключи/токены — в рамках используемого стека).

Тема 4. Интеграция AI/ML-компонента за API: пайплайн, препроцессинг/постпроцессинг, очереди, батчинг .

Организация пайплайна инференса: decode → preprocess → inference → postprocess. Форматирование входов/выходов модели и контроль размеров/типов данных. Управление контекстом модели и потокобезопасность при инференсе. Очереди задач, диспетчеризация и backpressure. Батчинг запросов и компромисс latency/throughput. Кэширование допустимых результатов и дедупликация запросов. Таймауты, отмена выполнения и деградация сервиса.

Раздел 3. Высокопроизводительные API-сервисы для ИИ

Тема 5. Многопоточность в API ИИ-сервисах: thread pool, синхронизация, атомики/блокировки, потокобезопасные структуры

Модели конкурентности API-сервера: per-request, пул потоков, producer–consumer. Проектирование thread pool и очередей задач под инференс. Потокобезопасность общих ресурсов API: кэш, метрики, пул соединений, лимитеры. Выбор atomic vs mutex/shared_mutex, типовые ошибки (race/deadlock) и

способы предотвращения. Снижение contention и практики low-lock/lock-free (на уровне применения). Проверка корректности под параллельной нагрузкой и воспроизведение проблем.

Тема 6. Производительность и профилирование API: метрики p95/p99, поиск узких мест, оптимизация сериализации/копирований/памяти

Метрики и целевые показатели API (p95/p99 latency, throughput, error rate). Профилирование обработчиков запросов и пайплайна инференса (CPU/память/I/O). Выявление bottleneck'ов: блокировки, аллокации, копирования, сериализация, очереди. Оптимизация форматов и сериализации данных, уменьшение overhead. Управление памятью и временем жизни объектов, снижение лишних копирований. Нагрузочное тестирование и сравнение конфигураций (пул потоков, батчинг, кэш).

Раздел 4. Оптимизация под платформы и ускорители

Тема 7. Платформенная оптимизация и embedded: ограничения, выбор библиотек/рантаймов, кросс-сборка, измерения на целевой платформе.

Анализ ограничений платформы: CPU/RAM/энергопотребление, ОС и доступные рантаймы. Выбор C++ библиотек/движков инференса под платформу (подходы и примеры). Оптимизация модели и вычислений: квантование, сжатие/прунинг, упрощение графа. Кросс-сборка и развёртывание, конфигурация зависимостей. Измерения на целевой платформе: latency, память, стабильность. Оптимизация препроцессинга и I/O для embedded-режимов.

Тема 8. GPU/FPGA-ускорение в ИИ-системе: готовые инструменты оптимизации (например, TensorRT), профилирование, анализ эффекта.

Выбор участков для ускорения: препроцессинг, инференс, постпроцессинг, батчинг. Применение готовых оптимизаторов/рантаймов (например, TensorRT) и режимов FP16/INT8. Асинхронность, управление потоками и пайплайнинг при работе с ускорителем. Минимизация CPU↔GPU/FPGA копирований и управление буферами. Профилирование ускоренного контура и поиск узких мест. Оценка эффекта ускорения и компромиссы качества/задержки/ресурсов.

Раздел 5. Качество и эксплуатация API ИИ-сервисов

Тема 9. Тестирование, документирование и эксплуатация: unit/integration/contract, мониторинг/трассировка, итоговый мини-проект.

Тестирование API: модульное, интеграционное, контрактное, негативные сценарии. Проверка корректности инференса на эталонных наборах (на уровне подхода). Документирование API (OpenAPI/Proto), примеры запросов/ответов и сценарии использования. Мониторинг и логирование: latency, RPS, ошибки, ресурсные показатели. Трассировка запросов и диагностика деградаций, базовые алерты. Итоговый мини-проект: прототип API-сервиса для ИИ, спецификация, тесты, отчёт по профилированию и оптимизациям, защита результата.

4.3 Лекции/лабораторные/практические/ занятия

Таблица 4а

Содержание лекций/лабораторного практикума/практических занятий занятий и контрольные мероприятия

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия[1]	Кол-во Часов/ из них практическая подготовка[2]
1.	**Раздел 1.** Основы API в ИИ-системах				
	Тема 1. Введение в API-технологии для ИИ: роль API, типовые архитектуры, требования (латентность, throughput, безопасность)	Лекция №1 API в ИИ-системах: архитектуры, контуры инференса, требования к latency/throughput и безопасности	ПК-16 (PL-3).1. ПК-16 (PL-3).2.	-	2
		Практическая работа №1. Проектирование архитектуры AI inference-сервиса и потоков данных	ПК-16 (PL-3).1. ПК-16 (PL-3).2.	Защита работы	2
		Практическая работа №2. Метрики и SLO для API инференса: p95/p99 latency, throughput, error rate, лимитирование и backpressure	ПК-16 (PL-3).1.	Защита работы	2
	Тема 2. Проектирование API-контрактов для ИИ-сервисов: REST/gRPC, модели данных, ошибки, версионирование	Лекция №2 Контракты API для инференса: REST vs gRPC, схемы данных, коды ошибок, идемпотентность, версионирование API и моделей.	ПК-16 (PL-3).1.	-	2
		Практическая работа №3. Проектирование REST API для ИИ-сервиса и описание контракта в OpenAPI (эндпоинты, схемы, ошибки, версии).	ПК-16 (PL-3).1.	Защита работы	2
		Практическая работа №4. Проектирование gRPC интерфейса инференса: proto, статусы, streaming/batch, стратегия совместимости и версионирования.	ПК-16 (PL-3).1.	Защита работы	2

[1] Вид контрольного мероприятия (текущий контроль) для практических и лабораторных занятий: устный опрос, контрольная работа, защита лабораторных работ, тестирование, коллоквиум и т.д.

[2] Участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью и направленных на формирование, закрепление, развитие практических навыков и компетенций по профилю образовательной программы.

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия[1]	Кол-во Часов/ из них практическая подготовка[2]
2.	**Раздел 2.** Реализация API на C++ и интеграция инференса				
	Тема 3. Реализация API на C++: HTTP (Boost.Asio/ Beast) и/или gRPC C++, сериализация JSON/Protobuf	Лекция №3 Реализация API на C++: HTTP на Boost.Asio/Beast и/или gRPC C++; сериализация JSON/Protobuf; обработка ошибок и логирование.	ПК-16 (PL-3).1.	-	2
		Практическая работа №5. Реализация HTTP endpoint'ов на Boost.Beast: приём запроса, валидация, формирование ответов и ошибок.	ПК-16 (PL-3).1.	Защита работы	2
		Практическая работа №6. Реализация gRPC сервиса на C++ (или расширение HTTP): Protobuf/JSON, единый формат ошибок, таймауты, request-id.	ПК-16 (PL-3).1.	Защита работы	2
	Тема 4. Интеграция AI/ML-компонента за API: пайплайн, препроцессинг/постпроцессинг, очереди, батчинг	Лекция №4 Инференс за API: пайплайн (pre/infer/post), очереди задач, батчинг, таймауты и деградация сервиса.	ПК-16 (PL-3).1. ПК-16 (PL-3).2.	-	2
		Практическая работа №7. Подключение рантайма инференса к C++ API (ONNX Runtime/OpenVINO/др.): конфигурация, форматы входов/выходов, измерение базовой latency.	ПК-16 (PL-3).2.	Защита работы	2
		Практическая работа №8. Очередь задач и батчинг инференса в API: реализация, настройка, сравнение latency/throughput и потребления памяти.	ПК-16 (PL-3).1. ПК-16 (PL-3).2.	Защита работы	2
3.	**Раздел 3.** Высокопроизводительные API-сервисы для ИИ				
	Тема 5. Многопоточность в API ИИ-сервисах: thread pool, синхронизация,	Лекция №5 Многопоточность C++ API-сервисов ИИ: thread pool, atomics vs locks, типовые гонки в кэше/очередях/метриках, устранение deadlock	ПК-16 (PL-3).1.	-	2
		Практическая работа №9.	ПК-16 (PL-3).1.	Защита работы	2/1

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия[1]	Кол-во Часов/ из них практическая подготовка[2]
	ато-мики/блоки-ровки, пото-кобезопас-ные струк-туры	Реализация пула потоков и потокобезопасных компонентов API (очередь, кэш, счётчики метрик/rate limit).			
		Практическая работа №10. Отладка конкурентных ошибок в API под нагрузкой: ThreadSanitizer, исправление гонок и дедлоков, повторная проверка.	ПК-16 (PL-3).1.	Защита работы	2
	Тема 6. Производительность и профилирование API: метрики p95/p99, поиск узких мест, оптимизация сериализации/копирований/памяти	Лекция №6 Профилирование и оптимизация API инференса: p95/p99, contention, аллокации, копирования, сериализация, настройка пула и батчинга.	ПК-16 (PL-3).1.	-	2
		Практическая работа №11. Нагрузочное тестирование API (wrk/hey/k6): сбор latency/throughput/error rate, построение профиля нагрузки.	ПК-16 (PL-3).1.	Защита работы	2/1
		Практическая работа №12. Профилирование (perf/VTune) и устранение bottleneck'ов: блокировки, аллокации, сериализация, копирования; повторные замеры.	ПК-16 (PL-3).1.	Защита работы	2
4.	**Раздел 4.** Оптимизация под платформы и ускорители				
	Тема 7. Платформенная оптимизация и embedded: ограничения, выбор библиотек/рантаймов, кросс-сборка, измерения на целевой платформе	Лекция №7 Edge/embedded ИИ-сервисы с API: ограничения платформ, выбор рантаймов, квантование/сжатие, кросс-сборка и измерения.	ПК-16 (PL-3).2.	-	2
		Практическая работа №13. Сборка/развертывание C++ API + инференс под целевую платформу (кросс-сборка или имитация среды), настройка зависимостей рантайма.	ПК-16 (PL-3).2.	Защита работы	2/1
		Практическая работа №14. Оптимизация модели для ограниченной платформы: квантование/упрощение	ПК-16 (PL-3).2.	Защита работы	2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия[1]	Кол-во Часов/ из них практическая подготовка[2]
		графа, сравнение latency/памяти/качества.			
	Тема 8. GPU/FPGA-ускорение в ИИ-системе: готовые инструменты оптимизации (например, TensorRT), профилирование, анализ эффекта	Лекция №8 GPU/FPGA ускорение в ИИ-системе с API: где ускорять, TensorRT/провайдеры ONNX Runtime, FP16/INT8, пайплайнинг и профилирование.	ПК-16 (PL-3).3.	-	2
		Практическая работа №15. Подготовка и развертывание оптимизированной модели на GPU (TensorRT или аналог): FP16/INT8, калибровка, проверка скорости/качества.	ПК-16 (PL-3).3.	Защита работы	2/1
		Практическая работа №16. Профилирование ускоренного инференса и API-контура (Nsight/CPU-профилировщик): поиск ограничителей производительности и оптимизация	ПК-16 (PL-3).3.	Защита работы	2
5	**Раздел 5.** Качество и эксплуатация API ИИ-сервисов				
	Тема 9. Тестирование, документирование и эксплуатация: unit/integration/contract, мониторинг/трассировка, итоговый мини-проект	Лекция №9 Качество и эксплуатация AI API-сервиса: тестирование (unit/integration/contract), документация, мониторинг/трассировка, регресс производительности.	ПК-16 (PL-3).1. ПК-16 (PL-3).2.	-	2
		Практическая работа №17. Контрактные и интеграционные тесты API + актуализация спецификации (OpenAPI/Proto), негативные сценарии и таймауты.	ПК-16 (PL-3).1.	Защита работы	2
		Практическая работа №18. Итоговый мини-проект: сборка сервиса, прогон нагрузочных тестов, профилирование, отчёт по оптимизациям и защита решения.	ПК-16 (PL-3).1. ПК-16 (PL-3).2. ПК-16 (PL-3).3.	Защита работы	2

Таблица 5а[3]

Перечень вопросов для самостоятельного изучения дисциплины

[3] Таблица 5а заполняется для очной формы обучения

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
Раздел 1		
1.	Тема 1 Введение в API-технологии для ИИ: роль API, типовые архитектуры, требования	Роль API в контурах ИИ (данные → препроцессинг → инференс → постпроцессинг → ответ). Типовые архитектуры API для ИИ: монолит/микросервисы, API gateway, inference-service, сервисы данных. Синхронные и асинхронные сценарии (очереди, события, callbacks). Показатели качества API для ИИ: latency, p95/p99, throughput, error rate, доступность, стоимость. Базовые риски безопасности API ИИ-сервиса (доступ к модели/данным, вредоносные входы, злоупотребление ресурсами). Компетенции: ПК-16 (PL-3).1. ПК-16 (PL-3).2.
2.	Тема 2. Проектирование API-контрактов для ИИ-сервисов: REST/gRPC, модели данных, ошибки, версионирование	REST vs gRPC для инференса: когда какой подход оправдан. Контракт запрос/ответ для модели: входные поля, параметры, метаданные, формат результата. Схемы данных и правила валидации (ограничения, типы, размеры, допустимые значения). Единая модель ошибок для AI API (ошибки клиента, сервера, модели, таймауты, перегрузка), коды/статусы и сообщения. Версионирование API и модели (api_version/model_version), совместимость и эволюция схем. Идемпотентность, дедупликация запросов, форматирование ответов и диагностика. Компетенции: ПК-16 (PL-3).1.
Раздел 2		
3	Тема 3. Реализация API на C++: HTTP (Boost.Asio/Beast) и/или gRPC C++, сериализация JSON/Protobuf	Архитектура C++ API-сервиса: обработчики, маршрутизация, middleware, конфигурация. Сборка проекта и управление зависимостями (CMake, структуры модулей). Сериализация и её влияние на производительность: JSON vs Protobuf, контроль схем и версий сообщений. Валидация входных данных и формирование единых ошибок API. Логирование и корреляция запросов (request-id), уровни логов и формат. Настройка таймаутов, keep-alive/соединений, базовые принципы TLS (на уровне понимания). Компетенции: ПК-16 (PL-3).1.
4	Тема 4. Интеграция AI/ML-компонента за API: пайплайн, препроцессинг/постпроцессинг, очереди, батчинг	Организация инференс-пайплайна: препроцессинг, инференс, постпроцессинг; где возникают задержки. Выбор и подключение рантайма инференса на C++ (ONNX Runtime/OpenVINO/TensorRT и др. — по стеку). Форматы входов/выходов и контроль размеров/типов тензоров, работа с памятью. Очереди задач, backpressure, стратегии отказа при перегрузке. Батчинг: правила формирования батча, компромисс latency vs throughput. Таймауты, отмена, “теплый старт” (warmup), кэширование допустимых результатов. Компетенции: ПК-16 (PL-3).1. ПК-16 (PL-3).2.
Раздел 3		
5	Тема 5. Многопоточность в API ИИ-сервисах: thread pool, синхронизация, атомики/блокировки, потокобезопасные структуры	Типовые конкурентные ошибки в API ИИ-сервиса: гонки на кэше, очередях, пулах соединений, метриках, контекстах модели. Thread pool и модели обработки запросов (producer–consumer), выбор размера пула. Когда использовать атомарные операции, а когда мьютексы/разделяемые блокировки; примеры для счётчиков и общих структур. Причины deadlock/livelock и базовые способы предотвращения (порядок захвата, scoped_lock, минимизация критических секций). Потокобезопасные структуры для API-контура: очередь задач, кэш, rate limiter. Компетенции: ПК-16 (PL-3).1.

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
6	Тема 6. Производительность и профилирование API: метрики p95/p99, поиск узких мест, оптимизация сериализации/копирований/памяти	Как корректно измерять latency/throughput для AI API (прогрев, стабильные входы, повторяемость). Интерпретация p95/p99 и типовые причины “длинного хвоста” задержек. Основы нагрузочного тестирования API и сценарии нагрузок (ступеньки, всплески, долгий прогон). Профилирование CPU/памяти и поиск bottleneck’ов: блокировки, аллокации, копирования, сериализация, I/O. Подходы к оптимизации: уменьшение копирований, эффективные буферы, выбор формата данных, настройка пула потоков и батчинга. Компетенции: ПК-16 (PL-3).1.
Раздел 4		
7	Тема 7. Платформенная оптимизация и embedded: ограничения, выбор библиотек/рантаймов, кросс-сборка, измерения	Анализ ограничений целевой платформы: CPU/RAM, энергопотребление, ОС/драйверы, доступные ускорители. Выбор рантайма инференса и зависимостей под платформу, критерии выбора (скорость, память, поддержка ops, переносимость). Методы облегчения модели и вычислений: квантование (INT8/FP16), pruning/сжатие, оптимизация графа; оценка влияния на качество и скорость. Кросс-компиляция и развёртывание (ARM/x86), особенности зависимостей и сборки. Измерения на целевой платформе: latency/throughput/память, выявление узких мест и их устранение. Компетенции: ПК-16 (PL-3).2.
8	Тема 8. GPU/FPGA-ускорение в ИИ-системе: инструменты оптимизации, профилирование, анализ эффекта	Где в AI API-сервисе появляется ускорение: инференс, препроцессинг, постпроцессинг; влияние передачи данных CPU↔GPU/FPGA. Применение готовых инструментов оптимизации и развертывания (например, TensorRT; провайдеры ускорения в ONNX Runtime/OpenVINO — по стеку), режимы FP16/INT8 и калибровка. Организация асинхронности и пайплайнинга, выбор размеров батча. Профилирование ускоренного контура и поиск ограничителей производительности (GPU/CPU, копирования, синхронизация, очереди). Критерии оценки эффекта ускорения и корректное сравнение конфигураций. Компетенции: ПК-16 (PL-3).3.
Раздел 5		
9	Тема 9. Тестирование, документирование и эксплуатация: unit/integration/contract, мониторинг/трассировка, мини-проект	Виды тестов для AI API: unit, integration, contract; негативные сценарии (таймауты, перегрузка, невалидные входы). Подходы к тестированию инференса: эталонные примеры, стабильность результатов, регресс по качеству и скорости. Поддержка документации API (OpenAPI/Proto), примеры запросов/ответов, политика версий. Мониторинг и диагностика: метрики latency/RPS/errors, логи с request-id, трассировка запросов и поиск “узких мест” в цепочке. Регресс производительности после изменений, базовые практики CI для сборки/тестов/проверок. Компетенции: ПК-16 (PL-3).1. ПК-16 (PL-3).2. ПК-16 (PL-3).3.

5. Образовательные технологии

Таблица 6

Применение активных и интерактивных образовательных технологий

№ п/п			Наименование используемых активных и интерактивных образовательных технологий
7	Тема 7. Производительность и профилирование API	Л	Практико-ориентированная лекция с разбором метрик p95/p99 и “длинного хвоста”.
		ПЗ	нагрузочное тестирование и профилирование (perf/VTune), мини-исследование и защита выводов. ИКТ: инструменты мониторинга/профилирования и сбор отчётов.
1.	Тема 1. Введение в API-технологии для ИИ	ЛЗ	Проблемное обучение: разбор типовых ошибок/дедлоков на примерах API-кэша/очереди.
		ПЗ	Практикум: командное решение задач по синхронизации (atomics/locks), code review решений. ИКТ: использование санитайзеров и средств статического анализа (по возможностям).
8	Тема 8. GPU/FPGA-ускоренный инференс	Л	Практико-ориентированная лекция с разбором метрик p95/p99 и “длинного хвоста”.
2	Тема 2. Проектирование API-контрактов для ИИ-сервисов	ЛЗ	нагрузочное тестирование и профилирование (perf/VTune), мини-исследование и защита выводов. ИКТ: инструменты мониторинга/профилирования и сбор отчётов.
		ПЗ	Проблемное обучение: разбор типовых ошибок/дедлоков на примерах API-кэша/очереди.
9	Тема 9. Тестирование	Л	Практикум: командное решение задач по синхронизации (atomics/locks), code review решений. ИКТ: использование санитайзеров и средств статического анализа (по возможностям).
3	Тема 3. Реализация API на C++	Л	Практико-ориентированная лекция с разбором метрик p95/p99 и “длинного хвоста”.
		ПЗ	нагрузочное тестирование и профилирование (perf/VTune), мини-исследование и защита выводов. ИКТ: инструменты мониторинга/профилирования и сбор отчётов.
		ПЗ	Проблемное обучение: разбор типовых ошибок/дедлоков на примерах API-кэша/очереди.
			Практикум: командное решение задач по синхронизации (atomics/locks), code review решений. ИКТ: использование санитайзеров и средств статического анализа (по возможностям).
4	Тема 4. Интеграция AI/ML-компонента за API	Л	Практико-ориентированная лекция с разбором метрик p95/p99 и “длинного хвоста”.
		ПЗ	нагрузочное тестирование и профилирование (perf/VTune), мини-исследование и защита выводов. ИКТ: инструменты мониторинга/профилирования и сбор отчётов.
5	Тема 5. Многопоточность в API ИИ-сервисах	Л	Проблемное обучение: разбор типовых ошибок/дедлоков на примерах API-кэша/очереди.
		ПЗ	Практикум: командное решение задач по синхронизации (atomics/locks), code review решений. ИКТ: использование санитайзеров и средств статического анализа (по возможностям).
6	Тема 6. Производительность и профилирование API	Л	Практико-ориентированная лекция с разбором метрик p95/p99 и “длинного хвоста”.
		ПЗ	нагрузочное тестирование и профилирование (perf/VTune), мини-исследование и защита выводов. ИКТ: инструменты мониторинга/профилирования и сбор отчётов.

6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины

6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

1) Примеры заданий практических работ

Практическая работа №1. Проектирование архитектуры API-сервиса для ИИ-инференса

Задание. Разработать архитектуру API-ориентированной ИИ-системы для выбранной задачи инференса, определить состав компонентов и взаимодействий, сформулировать требования к качественным характеристикам.

Порядок выполнения работы.

1) Выбрать вариант ИИ-задачи (по заданию преподавателя) и определить входные данные и ожидаемый результат.

2) Сформулировать ограничения на время ответа и объём входных данных.

3) Определить состав компонентов (API-сервис, модуль инференса, кэш, хранилище данных, очередь сообщений при необходимости).
4) Оформить архитектурную схему системы.
5) Описать последовательность обработки запроса для двух сценариев: корректный запрос; отказ (ошибка валидации, таймаут, перегрузка).
6) Сформулировать не менее пяти требований к качеству: задержка ответа, пропускная способность, устойчивость, ограничения на входные данные, базовые требования безопасности.
Отчет. Архитектурная схема. Описание последовательности обработки запроса для двух сценариев. Перечень требований к качеству.

Практическая работа №2. Метрики качества и целевые показатели API ИИ-сервиса
Задание. Определить метрики функционирования API ИИ-сервиса и установить целевые значения показателей качества (SLO), подготовить программу измерений.
Порядок выполнения работы.
1) Определить перечень метрик: среднее время ответа и процентильные значения (p95, p99), пропускная способность (запросов в секунду), доля ошибок, как минимум одна ресурсная метрика (CPU или память).
2) Задать целевые значения для ключевых метрик и описать критерии деградации качества.
3) Сформировать профиль нагрузки, включающий базовый режим, режим ступенчатого роста и режим всплеска нагрузки; указать длительность каждого режима.
4) Описать методику измерений: прогрев, количество повторов, фиксированный набор входных данных.
Отчет. Таблица метрик и целевых значений. Описание критериев деградации. Программа измерений с профилем нагрузки.

Практическая работа №3. Проектирование REST API и подготовка спецификации OpenAPI
Задание. Разработать контракт REST API сервиса инференса и оформить его в виде спецификации OpenAPI с примерами запросов и ответов.
Порядок выполнения работы.
1) Определить состав методов API: метод проверки работоспособности и метод инференса.
2) Описать структуру запроса инференса (входные данные, параметры, версия модели) и структуру ответа (результат, дополнительные показатели, метаданные).
3) Определить правила валидации входных данных (обязательные поля, допустимые диапазоны, ограничения на размер).
4) Разработать единый формат ошибок API и перечень типовых ошибок (валидация, таймаут, перегрузка).
5) Определить стратегию версионирования API и совместимости изменений. 6) Сформировать спецификацию OpenAPI и подготовить примеры трёх сценариев: корректный запрос; запрос с ошибкой валидации; отказ по перегрузке или таймауту.
Отчет. Файл спецификации OpenAPI. Примеры запросов и ответов для трёх сценариев. Краткое описание стратегии версионирования.

Практическая работа №4. Проектирование gRPC интерфейса и подготовка proto-описания
Задание. Разработать gRPC интерфейс сервиса инференса и оформить его в виде proto-описания с правилами совместимости.
Порядок выполнения работы.
1) Определить перечень rpc-методов (не менее одного метода инференса).
2) Описать сообщения запроса и ответа: входные данные, параметры, версия модели, идентификатор запроса, результат.
3) Определить обработку ошибок: типовые причины отказа и соответствующие статусы.
4) Сформулировать правила развития интерфейса без нарушения совместимости (добавление полей, запрет переиспользования номеров, reserved).
5) Подготовить файл proto-описания и примеры сообщения запроса и ответа.

Отчет. Файл proto. Правила совместимости (1–2 абзаца). Примеры запроса и ответа.

Практическая работа №5. Реализация прототипа HTTP API на C++
Задание. Реализовать прототип HTTP API-сервиса на C++ с методами проверки работоспособности и инференса, обеспечив валидацию и единый формат ошибок.
Порядок выполнения работы.
1) Развернуть проект C++ (структура, сборка).
2) Реализовать HTTP-сервер и метод проверки работоспособности.
3) Реализовать метод инференса, включая разбор входных данных в формате JSON и проверку обязательных полей.
4) Реализовать формирование ответа и единый формат ошибок для типовых ситуаций.
5) Реализовать журналирование запросов и ошибок, а также идентификатор запроса (при отсутствии у клиента генерировать на стороне сервера и возвращать в ответе).
6) Настроить ограничение на размер запроса и ограничение по времени обработки запроса.
7) Выполнить проверку работоспособности на наборе тестовых запросов.
Отчет. Исходный код. Инструкция сборки и запуска. Примеры запросов и ответов (включая ошибочный).

Практическая работа №6. Реализация API по разработанному контракту (gRPC или расширение HTTP)
Задание. Реализовать серверную часть API в соответствии с ранее разработанным контрактом и подготовить средства проверки корректности работы.
Порядок выполнения работы.
1) Выбрать реализацию: gRPC сервер на C++ по proto-описанию либо расширение HTTP API по спецификации OpenAPI.
2) Реализовать обработку запроса и формирование ответа в соответствии с контрактом, включая все обязательные поля.
3) Реализовать обработку ошибок: неверный формат данных, превышение допустимого размера, превышение таймаута.
4) Подготовить тестовый клиент или набор автоматизированных тестов, включающий не менее пяти сценариев (минимум два сценария должны быть ошибочными).
5) Провести проверку и зафиксировать результаты.
Отчет. Исходный код сервера. Тестовый клиент или набор тестов. Протокол проверки (сценарии и результаты).

Практическая работа №7. Интеграция рантайма инференса в API-сервис (режим CPU)
Задание. Подключить рантайм инференса на C++ к API-сервису, обеспечить выполнение инференса и провести измерения времени обработки запросов.
Порядок выполнения работы.
1) Подключить выбранный рантайм (например, ONNX Runtime или OpenVINO) и настроить зависимости проекта.
2) Реализовать загрузку модели при запуске сервиса.
3) Реализовать подготовку входных данных для модели и преобразование выхода модели в формат ответа API.
4) Выполнить серию измерений времени обработки для холодного режима (первые запросы после старта) и прогретого режима (после прогрева).
5) Рассчитать среднее и p95 для каждой серии и сравнить результаты.
Отчет. Код интеграции инференса. Таблица измерений. Выводы по сравнению холодного и прогретого режимов.

Практическая работа №8. Очередь задач и батчинг запросов инференса
Задание. Реализовать очередь задач и механизм батчинга запросов инференса, обеспечить защиту от перегрузки и выполнить сравнительные измерения.

Порядок выполнения работы.
1) Реализовать постановку запросов в очередь задач между обработчиком API и модулем инференса.
2) Реализовать механизм формирования батча по двум ограничениям: максимальный размер батча и максимальное время ожидания.
3) Реализовать ограничение на размер очереди и корректный отказ в обслуживании при перегрузке.
4) Выполнить измерения производительности для трёх разных настроек батча и сравнить значения p95 времени ответа и пропускной способности.
5) Сделать вывод о наиболее рациональной настройке.
Отчет. Реализация очереди и батчинга. Таблица сравнительных измерений. Обоснование выбранной конфигурации.

Практическая работа №9. Многопоточная обработка запросов с использованием пула потоков
Задание. Реализовать обработку запросов в многопоточном режиме с применением пула потоков и оценить влияние параметров пула на производительность.
Порядок выполнения работы.
1) Реализовать пул потоков и определить, какие этапы обработки запроса выполняются в пуле.
2) Реализовать настройку числа потоков через параметры конфигурации.
3) Провести нагрузочные измерения при нескольких значениях числа потоков (например, 1, 2, 4, 8). 4) Сравнить пропускную способность и p95 времени ответа, сделать вывод о влиянии числа потоков.
Отчет. Реализация пула потоков. Таблица измерений для разных значений. Выводы.

Практическая работа №10. Потокобезопасность общих структур данных API-сервиса
Задание. Обеспечить потокобезопасный доступ к общим структурам данных API-сервиса (кэш и метрики) и подтвердить корректность под параллельной нагрузкой.
Порядок выполнения работы.
1) Реализовать кэш результатов инференса (с ограничением размера и временем жизни записей).
2) Реализовать набор метрик (не менее трёх), фиксируемых в процессе работы сервиса.
3) Обеспечить потокобезопасность: для счётчиков использовать атомарные операции; для кэша применить блокировки и обосновать выбор.
4) Провести проверку под параллельной нагрузкой и убедиться в корректности работы (нет аварийных завершений, метрики изменяются согласованно).
Отчет. Исходный код. Описание применённых механизмов синхронизации. Результаты проверки под нагрузкой.

6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Оценочные средства текущего контроля успеваемости и сформированности компетенций основана на подсчете баллов, «заработанных» студентом в течение семестра.

Успеваемость студента по дисциплине оценивается в баллах от 0 до 100.

Оценка знаний проводится по следующим критериям:

- посещение занятий – 10 баллов;
- выполнение практических заданий – 10 баллов;
- выполнение контрольной работы - 10 баллов;
- качество коллоквиума – 10 баллов;

- качество курсового проекта - 20 баллов;
- промежуточный контроль (зачет) – 20 баллов;
- промежуточный контроль (экзамен) – 20 баллов.

Соответствие балльной оценки общепринятой 4-х балльной шкале оценок приведено в таблице 7.

Таблица 7

Соответствие балльных оценок по 4-х балльной шкале

Балльная оценка	Оценка по 4хбалльной шкале	Оценка по шкале «Зачтено» / «Не зачтено»
0-59	Неудовлетворительно - 2	Не зачтено
60-69	Удовлетворительно - 3	Зачтено
70-89	Хорошо – 4	Зачтено
90-100	Отлично - 5	Зачтено

Критерии оценивания результатов обучения показаны в таблицах 8,9.

Таблица 8

Критерии оценивания по шкале «Зачтено» / «Не зачтено»

Оценка «Зачтено/Не зачтено»	Критерии оценивания
Зачтено	Оценка «**зачтено**» ставится, если студент показал глубокие систематизированные знания в объеме, необходимом для дальнейшей учебы и в предстоящей работе по профессии, владеет приемами рассуждения и сопоставления материала из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов; выполнил все практические задания, предоставив правильные и аргументированные выводы в соответствии с предъявленными требованиями.
Незачтено	Оценка «**не зачтено**» ставится, если студент в ответах не раскрыл основное содержание вопросов, носящих несистематизированный, отрывочный, поверхностный характер; студент не понимает существа излагаемых им вопросов, что свидетельствует о том, что студент не может дальше продолжать обучение или приступать к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине; не выполнил практические задания в соответствии с предъявленными требованиями.

Таблица 9

Критерии оценивания результатов обучения (зачет)

Оценка	**Критерии оценивания**
Высокий уровень «5» (отлично)	оценку **«отлично»** заслуживает студент, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы. **Компетенции**, закреплённые за дисциплиной, **сформированы на уровне – высокий**.
Средний уровень «4»	оценку **«хорошо»** заслуживает студент, практически полностью освоивший знания, умения, компетенции и теоретический

(хорошо)	материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки. **Компетенции**, закреплённые за дисциплиной, **сформированы на уровне – хороший (средний)**.
Пороговый уровень «3» (удовлетворительно)	оценку **«удовлетворительно»** заслуживает студент, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы. **Компетенции**, закреплённые за дисциплиной, **сформированы на уровне – достаточный**.
Минимальный уровень «2» (неудовлетворительно)	оценку «**неудовлетворительно»** заслуживает студент, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы. **Компетенции**, закреплённые за дисциплиной, **не сформированы**.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

1. Jiayi Wang, Guoliang Li AOP: Automated and Interactive LLM Pipeline Orchestration for Answering Complex Queries URL: https://vldb.org/cidrdb/papers/2025/p32-wang.pdf (дата доступа 28 августа 2025 г.)
2. Kim, S., Yu, Y. & Seo, H. Artificial intelligence orchestration for text-based ultrasonic simulation via self-review by multi-large language model agents. Sci Rep 15, 12474 (2025).URL: https://doi.org/10.1038/s41598-025-97498-y (дата доступа 28 августа 2025 г.)
3. Промышленные API для распознавания изображений, речи, рекомендаций, используемые как backend для LLM-агентов (описание входных/выходных форматов, SLA). URL: https://orq.ai/blog/llm-orchestration

7.2. Дополнительная литература

1. Петрова Е.С. Модели tool calling в LLM через REST API: безопасность и масштабируемость // Научный результат. Информационные технологии. 2025. Т. 10. № 2.
2. Смирнов Д.Ю. Защита API от prompt-инъекций в сервисах генеративного ИИ // Искусственный интеллект и принятие решений. 2025. № 1. С. 112–130.
3. Лебедев М.А. gRPC vs REST для model-serving в ИИ-приложениях // Вестник компьютерных и информационных технологий. 2025. № 1. (2 уровень).
4. Федоров И.П. Аутентификация и rate limiting в multi-tenant AI API // Программные продукты и системы. 2024. № 3. (2 уровень, ВАК по ИС).

5. Григорьева О.Н. API-интерфейсы для vision-language моделей: протоколы и производительность // Искусственный интеллект и принятие решений. 2024. № 3. С. 78–95. (1 уровень).

7.3. Нормативные правовые акты

1. ГОСТ Р 59277 2020. Системы искусственного интеллекта. Классификация, термины и общие положения. – М.: Стандартинформ.
2. ГОСТ Р 59898 2021. Оценка качества систем искусственного интеллекта. Общие положения. – М.: Стандартинформ.
3. ГОСТ Р 71476 2024. Искусственный интеллект. Концепции и терминология искусственного интеллекта. – М.: Стандартинформ.
4. ГОСТ Р ИСО/МЭК 25010 2015. Системы и программная продукция. Модели качества. – М.: Стандартинформ.
5. ГОСТ Р ИСО/МЭК 12207 2010. Информационная технология. Процессы жизненного цикла программных средств. – М.: Стандартинформ.
6. ГОСТ 19.201 78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. – М.: Изд во стандартов.
7. ГОСТ 19.502 78. Единая система программной документации. Описание применения. Требования к содержанию и оформлению. – М.: Изд во стандартов.
8. ГОСТ 34.602 89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – М.: Изд во стандартов.
9. ГОСТ Р 57580 2017. Защита информации финансовых организаций. Общие положения. – М.: Стандартинформ.
10. ГОСТ Р 56939 2016. Защита информации. Обеспечение безопасности персональных данных при их обработке в информационных системах. – М.: Стандартинформ.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Hugging Face – платформа открытых моделей и датасетов, документация по Model Hub и Inference API: https://huggingface.co
2. GitHub – репозитории библиотек и примеров сервисов с ИИ-API (в т. ч. организации Hugging Face, OpenAI и др.): https://github.com
3. OpenAI – документация по API для работы с языковыми и мультимодальными моделями: https://platform.openai.com/docs
4. Google Cloud AI / Vertex AI – документация по API для NLP, Vision, Speech, табличных данных и генеративного ИИ: https://cloud.google.com/ai
5. Microsoft Azure AI (Azure AI Studio, Cognitive Services) – API-сервисы анализа текста, изображений, речи и генеративного ИИ: https://azure.microsoft.com/en-us/products/ai-services

6. Amazon Web Services (Amazon Bedrock и другие AI-сервисы) – API для генеративных и классических ML-моделей: https://aws.amazon.com/machine-learning
7. Платформы российских провайдеров генеративного ИИ и AI-API (GigaChat, YaGPT, др.) – официальные порталы с документацией по REST/SDK-интерфейсам.
8. Документация библиотеки Transformers (работа с моделями Hugging Face, примеры API-интеграции): https://huggingface.co/docs/transformers
9. Документация Hugging Face Hub / Inference Endpoints (управление репозиториями и деплой моделей как сервисов): https://huggingface.co/docs/hub
10. Документация по фреймворкам оркестрации LLM и агентным системам (chain-/agent-подходы, tool calling, вызов внешних API).
11. Документация web-фреймворков для создания REST/gRPC-сервисов (FastAPI, Django REST Framework, Flask, gRPC и др.).
12. Облачные и локальные среды для практики (Google Colab, Kaggle Notebooks, JupyterHub в вузе и др.).

9. Перечень программного обеспечения и информационных справочных систем

1. Базы данных Министерства сельского хозяйства Российской Федерации: www.mcx.ru.

2. Базы данных Федеральной службы государственной статистики: www.gks.ru.

3. Python 3.12+ – интерпретатор языка программирования для разработки API-сервисов и интеграции моделей ИИ (библиотеки requests, FastAPI, Flask).

4. FastAPI – фреймворк для создания высокопроизводительных REST/gRPC API с автоматической документацией (Swagger/OpenAPI).

5. Hugging Face Transformers – библиотека для загрузки, дообучения и инференса моделей ИИ (LLM, CV, speech) с примерами API-интеграции.

6. LangChain / LlamaIndex – фреймворки оркестрации LLM-агентов, tool calling и цепочек вызовов внешних API.

7. GitHub Copilot / Cursor AI – ИИ-ассистенты для автодополнения кода в IDE (VS Code, JetBrains), генерации API-эндпоинтов и тестов.

8. Docker / Podman – контейнеризация для деплоя ИИ-моделей как микросервисов с API (многоуровневая архитектура).

9. Google Colab / Kaggle Notebooks – облачные Jupyter-среды для прототипирования API-сервисов с моделями ИИ.

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины (модуля)	Наименование программы	Тип программы	Автор / организация	Год разработки / актуальной версии

1	API-технологии в ИИ. Основы разработки сервисов	Python 3.x	Язык программирования, среда выполнения	Python Software Foundation	2008 / актуальная версия
2	API-технологии в ИИ. Веб-сервисы и REST	FastAPI	Фреймворк для разработки web- и REST-API	Sebastián Ramírez и соавт.	2018 / актуальная версия
3	Модели ИИ и интеграция по API	Hugging Face Transformers	Библиотека для работы с моделями ИИ	Hugging Face Inc.	2018 / актуальная версия
4	Оркестрация LLM и вызов внешних API	LangChain	Фреймворк для построения LLM-агентов и цепочек	LangChain Inc.	2022 / актуальная версия
5	Контейнеризация ИИ-сервисов с API	Docker Desktop / Docker Engine	Платформа контейнеризации	Docker Inc.	2013 / актуальная версия
6	Управление исходным кодом и проектами	Git / GitHub	Система контроля версий и хостинг репозиториев	Linus Torvalds, GitHub Inc.	2005 / актуальная версия
7	Проектирование и тестирование API	Postman	Среда тестирования и документации API	Postman Inc.	2014 / актуальная версия
8	Демонстрация и лабораторные работы по ИИ	Jupyter Notebook / JupyterLab	Интерактивная среда для выполнения кода	Project Jupyter	2015 / актуальная версия

Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных* помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	**Оснащенность специальных помещений и помещений для самостоятельной работы****
1	2
Корпус 1, Аудитория 201 Количество рабочих мест: 24	Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE.
Корпус 1, Аудитория 203 Количество рабочих мест: 18	Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE.Структурное подразделение: Кафедра Цифровая кафедра
Корпус 1, Аудитория 206 Количество рабочих мест: 24	Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi

	AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE.
Центральная научная библиотека имени Н.И. Железнова	Читальные залы библиотеки
Студенческое общежитие	Комната для самоподготовки

11. Методические рекомендации обучающимся по освоению дисциплины

Освоение дисциплины «API-технологии» требует активной вовлечённости обучающихся в процесс решения прикладных задач, связанных с проектированием, разработкой и сопровождением API-сервисов для компонентов искусственного интеллекта, включая интеграцию инференса, обеспечение устойчивости и производительности, а также тестирование и оценку качества полученных решений. Дисциплина ориентирована на развитие проектного и исследовательского подхода с использованием кейсов, в том числе приближённых к индустриальным сценариям (например, построение API для сервиса инференса, интеграция с сервисом данных, обеспечение требований по задержке и пропускной способности).

Лекционные занятия направлены на формирование системного понимания принципов построения API в составе ИИ-систем, проектирования контрактов (REST/gRPC), организации обмена данными и обработки ошибок, а также разработки высокопроизводительных решений на C/C++ с применением многопоточности и оптимизации под различные аппаратно-программные платформы, включая встроенные системы и ускорители. Лекции сопровождаются презентациями, схемами архитектур и потоков данных, примерами контрактов и фрагментами кода, проходят с использованием мультимедийного оборудования и интерактивных платформ и электронных курсов (например, LMS, системы опроса и тестирования, репозитории с учебными примерами, средства совместной работы).

Студенты обязаны вести тематический конспект, дополняя его материалами из рекомендованной и дополнительной литературы, в том числе отраслевыми источниками и официальной документацией стандартов и библиотек (C++17/20, gRPC/Protocol Buffers, Boost.Asio/Beast, документация рантаймов инференса и средств оптимизации моделей). Перед каждой новой темой рекомендуется повторение ключевых концепций и самостоятельное выполнение минизаданий на закрепление.

Практические занятия проводятся в компьютерных классах, оборудованных современным программным обеспечением, и строятся по принципу: постановка инженерной задачи – разбор эталонного решения/шаблона – выполнение индивидуального варианта – анализ и обсуждение результатов. Каждое практическое задание направлено на развитие конкретных навыков, в том числе:

• проектирование и документирование API-контрактов для ИИ-сервисов (OpenAPI/Proto), включая версионирование и обработку ошибок;

• реализация API-сервисов на C++ (HTTP и/или gRPC), интеграция с модулем инференса и сервисами данных;

• обеспечение конкурентной обработки запросов (пул потоков, синхронизация, атомарные операции, потокобезопасные структуры);

• измерение и анализ метрик качества API (p95/p99 latency, throughput, error rate), нагрузочное тестирование и профилирование кода;

• оптимизация решений под целевые платформы (в том числе embedded), применение оптимизаций моделей и вычислений (квантование, сжатие, оптимизация графа, батчинг);

• использование ускорителей (GPU и при наличии FPGA) и готовых инструментов оптимизации (например, TensorRT и/или провайдеры ускорения), профилирование ускоренного контура.

Результаты работы оформляются в виде исходного кода и отчётных материалов (контракт API, инструкция сборки и запуска, результаты тестирования и измерений производительности) и размещаются в системе контроля версий и/или в LMS (например, Git, GitLab/GitHub, корпоративный репозиторий, электронный курс).

Наиболее трудоёмкие темы дисциплины:

• Тема 5. Многопоточность и потокобезопасность в API ИИ-сервисах на C++;

• Тема 6. Нагрузочное тестирование, профилирование и оптимизация производительности API;

• Тема 7–8. Платформенная оптимизация (embedded) и ускорение на GPU/FPGA (при наличии условий).

Самостоятельная работа включает:

• изучение справочной и профессиональной документации по C++17/20, gRPC/Protocol Buffers, HTTP-стекам, рантаймам инференса и инструментам оптимизации;

• выполнение заданий по проектированию контрактов API, разработке и тестированию API-сервиса, анализу метрик и профилированию;

• подготовку мини-проектов, сравнительных измерений и отчётов по принятым архитектурным и оптимизационным решениям.

Для полноценного освоения дисциплины студенту необходимо:

• посещать все аудиторные формы занятий (лекции и практики);

• поддерживать личную систему организации разработки (структура проекта, сборка, конфигурации, контроль версий);

• использовать электронные ресурсы для хранения прогресса и материалов (Git, облачные хранилища, LMS);

• принимать участие в консультациях, включая онлайн-формат (через LMS, мессенджеры, e-mail);

• активно участвовать в коллективной обратной связи при разборе решений и защите практических работ.

Промежуточная и итоговая аттестация проводится на основе совокупности оценок за выполненные практические работы, участие в защите и анализ выбранного кейса. Итоговая форма зачёта — защита индивидуального (или парного) проекта, включающего разработку API-сервиса для ИИ-задачи на C++ с демон-

страцией работы, представлением контракта, результатами тестирования и профилирования, а также обоснованием применённых оптимизаций под целевую платформу и (при наличии) ускорители.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятия обязан отработать:

Пропущенные лекции – предоставив преподавателю конспект лекции, ответив на вопросы устно, пройдя собеседование по пропущенной теме, пройти тестирование.

Пропущенные практические занятия – в форме выполненных заданий, устного опроса, посещения дополнительных занятий.

Защита индивидуальных заданий проводятся в часы в дни и часы, устанавливаемые преподавателем.

Пропуск занятия по документально подтвержденной дирекцией уважительной причине не является основанием для снижения оценки выполненной практической работы.

Методические рекомендации преподавателям по организации обучения по дисциплине

Преподавание курса «API-технологии» должно носить контекстный характер и обеспечивать формирование у обучающихся профессионально значимых компетенций, связанных с проектированием, разработкой и сопровождением API-сервисов в составе систем искусственного интеллекта. В процессе обучения должна отчётливо прослеживаться целевая установка на развитие личности и инженерного мышления; интеграционное единство форм, методов и средств обучения; взаимодействие обучающихся и преподавателя; учет индивидуального стиля учебной деятельности и педагогической деятельности.

Реализация технологий контекстного обучения в профессионально-образовательном процессе обеспечивается соблюдением следующих условий: мотивационное обеспечение обучающихся на основе включения в профессионально ориентированные задачи (проектирование контракта инференса, обеспечение требований по задержке и пропускной способности, выбор платформы развертывания); наличие диагностически заданной цели обучения, то есть измеримого представления об ожидаемом результате (достижение индикаторов ПК-16 (PL-3).1–ПК-16 (PL-3).3, подтверждаемое результатами тестирования, профилирования и сравнительных измерений); представление учебного материала в виде системы познавательных и практических задач, ситуаций, заданий, проектов и упражнений (контракты REST/gRPC, реализация API на C++, интеграция рантайма инференса, многопоточность, оптимизация под платформы и ускорители); описание способов взаимодействия субъектов образовательного процесса (обсуждение архитектурных решений, совместный разбор типовых ошибок, взаимная экспертиза контрактов и отчётов); обозначение границ правилосообразной (алгоритмической) и творческой деятельности (обязательные требования к контракту и корректности работы сервиса, при этом допускается выбор стеков и ва-

риантов оптимизации при обосновании); обеспечение открытости обучения профессиональному будущему, ориентированность на практики разработки AI API-сервисов (наблюдаемость, тестируемость, безопасность, воспроизводимость измерений).

В результате изучения дисциплины студенты получают знания и навыки, необходимые для разработки высокопроизводительных API-сервисов на C++ в составе ИИ-систем, включая конкурентную обработку запросов, обеспечение потокобезопасности, профилирование и оптимизацию, а также выбор и применение библиотек и инструментов для целевых платформ, включая встроенные устройства и аппаратные ускорители (GPU и при наличии FPGA). Существенное внимание уделяется анализу качества решений по метрикам (p95/p99 задержки, пропускная способность, доля ошибок), корректной постановке экспериментов и интерпретации результатов.

Методика преподавания дисциплины строится на сочетании лекций с практическими занятиями; групповыми и индивидуальными консультациями по отдельным разделам программы; внеаудиторной самостоятельной работой обучающихся (работа с учебниками и учебными пособиями, методическими указаниями и заданиями, изучение специализированной литературы, поиск необходимой информации в сети Интернет, работа с официальной документацией библиотек и SDK). Самостоятельная работа ориентирована на освоение теоретических положений, подготовку к практическим занятиям, а также оформление отчётных материалов по измерениям производительности и принятым инженерным решениям.

Лекционный курс должен быть логичным и последовательным. Каждая лекция начинается с актуализации знаний и постановки цели занятия и задач, которые должны быть достигнуты в ходе изучения материала. Проведение лекций рекомендуется осуществлять на основе проблемного метода обучения, стимулирующего самостоятельный поиск решений и аргументацию выбора технологий и архитектурных подходов. Для повышения интереса и обеспечения наглядности используются мультимедийные средства (презентации, схемы архитектур и потоков данных, примеры контрактов и фрагменты кода), а также элементы интерактивного обучения (обсуждение вариантов контрактов, сравнительный анализ архитектур, разбор инженерных компромиссов latency/throughput/стоимость). В дополнение к традиционной лекции целесообразно применять проблемные лекции, лекции-визуализации, бинарные лекции (например, совмещение вопросов API и вопросов оптимизации инференса), дискуссии по выбору стека и методов оптимизации.

Важная роль на лекциях отводится дискуссии: обучающиеся рассматриваются как участники профессионального диалога, способные предлагать решения, аргументировать выбор и критически анализировать альтернативы. Каждая лекция завершается подведением итогов и формулировкой выводов, а также указанием материалов для самостоятельного изучения и подготовки к практическим занятиям.

Практические занятия строятся по аналогичной структуре: актуализация знаний, постановка цели и задач, выполнение работы, анализ полученных ре-

зультатов и формулирование выводов. Практические работы должны соответствовать принципам контекстного подхода и включать исследовательские задачи профессиональной направленности: проектирование и документирование REST/gRPC контрактов, реализацию API-сервисов на C++ с интеграцией инференса, обеспечение многопоточности и потокобезопасности, проведение нагрузочного тестирования и профилирования, оптимизацию под ограничения платформы и использование ускорителей. На практических занятиях рекомендуется применять технологии дифференцированного обучения, включая поддержку обучающихся, испытывающих затруднения, и расширенные задания для обучающихся с более высоким уровнем подготовки.

Практические занятия проводятся под руководством преподавателя и предусматривают анализ типовых ошибок, допущенных при выполнении заданий, и разбор наиболее удачных решений. Обучающиеся привлекаются к сравнительному анализу предложенных вариантов, обсуждают достоинства и недостатки, приобретают навыки ведения дискуссии и обоснования инженерных решений. Успех закрепления знаний и умений обеспечивается системой текущего контроля, включающей проверку результатов практических работ, защиту выполненных заданий и оценку отчётных материалов (контракты API, результаты нагрузочных измерений, выводы профилирования).

В процессе самостоятельной работы обучающиеся закрепляют теоретические положения, изучают примеры, рассмотренные на практических занятиях, и выполняют индивидуальные задания, которые при возможности соотносятся с научными интересами обучающегося или тематикой выпускной квалификационной работы. Существенное значение имеет работа с литературой и официальными источниками (документация C++17/20, gRPC/Protocol Buffers, Boost.Asio/Beast, рантаймы инференса и инструменты оптимизации, руководства по профилированию и нагрузочному тестированию), а также анализ актуальных отраслевых практик разработки AI API-сервисов.

Особенности методики преподавания данной дисциплины состоят в интенсификации теоретической, практической и самостоятельной работы обучающихся и широком применении активных и интерактивных форм и методов обучения, ориентированных на решение профессионально значимых задач разработки и оптимизации API-сервисов в составе систем искусственного интеллекта.

Программу разработал:

Лапшин М.С., ассистент _____
(подпись)