

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ястребов Олег Александрович

Должность: Ректор

Дата подписания: 25.05.2026 12:25:52

Уникальный программный ключ:

ca953a01204891083f939673078ef1a989dae18a

**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**

### **ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ**

(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 «Параллельное и распределенное программирование» 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 5 семестре 3 курса. Дисциплину реализует Кафедра прикладного искусственного интеллекта. Дисциплина состоит из 4 разделов и 26 тем и направлена на изучение принципов и технологий параллельного и распределённого программирования: архитектур параллельных вычислительных систем (многоядерные процессоры, GPU, кластеры), моделей параллелизма (потoki, процессы, SIMD, SPMD), средств многопоточного программирования (threading, multiprocessing, concurrent.futures, asyncio), технологий распределённых вычислений (MPI, gRPC, очереди сообщений), архитектурных паттернов распределённых систем (клиент-сервер, MapReduce, микросервисы), а также применения параллельных и распределённых подходов для ускорения обучения и инференса моделей машинного обучения, включая data parallelism, pipeline parallelism и распределённую обработку данных.

Целью освоения дисциплины является формирование у студентов системных знаний и практических навыков проектирования и реализации параллельных и распределённых программных решений для задач обработки данных и систем искусственного интеллекта, включая владение средствами многопоточного и многопроцессного программирования, асинхронного ввода-вывода, распределённых вычислений, способность анализировать ограничения производительности и масштабируемости, обосновывать выбор архитектурных паттернов и проектировать инфраструктуру для параллельного обучения и развёртывания моделей ИИ.

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Параллельное и распределенное программирование» 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)
ОПК-2	Способен понимать принципы работы современных информационных технологий и применять компьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ОПК-2.1 Знает принципы работы современных информационных технологий, включая технологии машинного обучения, облачных вычислений, высокопроизводительных вычислений (HPC) и параллельного программирования; ОПК-2.3 Владеет навыками использования вычислительных методов, включая массово-параллельные вычисления на GPU, для обучения и развёртывания моделей ИИ;
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования,	ОПК-3.3 Владеет навыками создания, тестирования и отладки алгоритмических и программных решений для систем ИИ, включая разработку пайплайнов обработки данных и обучения моделей;

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
	пригодные для практического применения	
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	ПК-1.1 Анализирует возможности реализации функциональных и нефункциональных требований к ПО систем ИИ, выявляет противоречия и ограничения;
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;
FC-1	Способен проводить передовые исследования в области архитектур, алгоритмов МО, оптимизации и математики	FC-1.3 Развивает методы ускорения обучения;
PL-1	Способен применять язык программирования Python для решения задач в области ИИ	PL-1.3 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности;

### 3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Параллельное и распределенное программирование» 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 «Параллельное и распределенное программирование».

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

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
ОПК-2	Способен понимать принципы работы современных информационных технологий и применять компьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	История и теория программирования; Введение в искусственный интеллект; Программирование на языке Python;	Массово-параллельные вычисления в машинном обучении (GPU); Основы глубокого обучения; Методы машинного обучения; Нейронные сети;
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и	Технологическая (проектно-технологическая) практика (учебная); Дискретная математика; История и теория программирования; Программирование на языке C++;	Методы машинного обучения; Нейронные сети; Технологическая (проектно-технологическая) практика (производственная);

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
	информационных моделей, баз данных, средств тестирования, пригодные для практического применения	Алгоритмы и структуры данных; Введение в базы данных; Программирование на языке Python; Методы разработки решений на основе искусственного интеллекта (Git, Docker);	
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	Эксплуатационная практика (учебная); 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; Программирование на языке C++; Методы разработки решений на основе искусственного интеллекта (Git, Docker); Введение в базы данных;	Эксплуатационная практика (производственная); Преддипломная практика; Технологическая (проектно-технологическая) практика (производственная); Методы машинного обучения; Массово-параллельные вычисления в машинном обучении (GPU); Оптимизация моделей машинного обучения; Основы глубокого обучения; Безопасность систем искусственного интеллекта; Практическая подготовка на проектах отраслевых промышленных партнеров; <i>Большие языковые модели</i> **; MLOps и промышленная разработка систем искусственного интеллекта; Нейронные сети; 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	Программирование на языке C++; Методы разработки решений на основе искусственного интеллекта (Git, Docker); Алгоритмы и структуры данных; Программирование на языке Python; Введение в базы данных; Технологическая (проектно-технологическая) практика (учебная); Эксплуатационная практика (учебная);	Массово-параллельные вычисления в машинном обучении (GPU); MLOps и промышленная разработка систем искусственного интеллекта; Практическая подготовка на проектах отраслевых промышленных партнеров; Проектирование и разработка систем компьютерного зрения; Практикум по обработке естественного языка (NLP); Основы глубокого обучения; <i>Вайб-коддинг</i> **; Эксплуатационная практика (производственная); Преддипломная практика; Технологическая (проектно-технологическая) практика

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
			(производственная);
PL-1	Способен применять язык программирования Python для решения задач в области ИИ	Технологическая (проектно-технологическая) практика (учебная); Программирование на языке Python; Алгоритмы и структуры данных; Статистические методы и первичный анализ данных;	Технологическая (проектно-технологическая) практика (производственная); Эксплуатационная практика (производственная); <i>Вайб-кодиг</i> **; Методы машинного обучения; Основы глубокого обучения;
FC-1	Способен проводить передовые исследования в области архитектур, алгоритмов МО, оптимизации и математики	Эксплуатационная практика (учебная); Линейная алгебра; Математический анализ; Теория вероятностей и математическая статистика; Введение в искусственный интеллект; Численная линейная алгебра;	Преддипломная практика; Технологическая (проектно-технологическая) практика (производственная); Методы машинного обучения; Оптимизация моделей машинного обучения; Нейронные сети; Практическая подготовка на проектах отраслевых промышленных партнеров; Основы глубокого обучения; Массово-параллельные вычисления в машинном обучении (GPU);

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

\*\* - элективные дисциплины /практики

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Параллельное и распределенное программирование» составляет «3» 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)
			5
<i>Контактная работа, ак.ч.</i>	51		51
Лекции (ЛК)	0		0
Лабораторные работы (ЛР)	0		0
Практически/семинарские занятия (СЗ)	51		51
<i>Самостоятельная работа обучающихся, ак.ч.</i>	48		48
<i>Контроль (экзамен/зачет с оценкой), ак.ч.</i>	9		9
<b>Общая трудоемкость дисциплины</b>	<b>ак.ч.</b>	<b>108</b>	<b>108</b>
	<b>зач.ед.</b>	<b>3</b>	<b>3</b>

## 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	Основы параллельных вычислений и многопоточное программирование	1.1	Введение в параллельные вычисления. Архитектуры	Мотивация: закон Мура и стена мощности, многоядерная эра. Классификация Флинна: SISD, SIMD, MISD, MIMD. Архитектуры: многоядерные CPU, GPU (SIMT), кластеры, облачные вычисления. Модели памяти: общая (shared memory), распределённая (distributed memory), гибридная. Закон Амдала и закон Густафсона: предел ускорения. Связь с обучением моделей ИИ: почему нужен параллелизм	СЗ	ОПК-2.1, ПК-1.1
		1.2	Потоки и процессы в Python. GIL	Процесс vs. поток: адресное пространство, overhead, коммуникация. Global Interpreter Lock (GIL) в CPython: что блокирует, почему существует, как обойти. Модуль threading: создание потоков, Thread, join, daemon. Модуль multiprocessing: Process, start, join. Сравнение: threading vs. multiprocessing для CPU-bound и IO-bound задач. Практика: замеры времени для обоих подходов	СЗ	ОПК-2.3, ОПК-3.3
		1.3	Синхронизация потоков и процессов	Проблемы параллелизма: гонки данных (race conditions), дедлоки (deadlocks), livelock, starvation. Примитивы синхронизации: Lock, RLock, Semaphore, Event, Condition, Barrier. Атомарные операции. Паттерн producer-consumer с Queue. Практика: демонстрация гонки данных, исправление с помощью Lock, реализация producer-consumer	СЗ	ОПК-2.3, ОПК-3.3
		1.4	Пулы потоков и процессов. concurrent.futures	Высокоуровневый API: concurrent.futures — ThreadPoolExecutor, ProcessPoolExecutor. Методы: submit, map, as_completed, result. Обработка исключений в пулах. Выбор числа воркеров. Практика: параллельная обработка набора данных (загрузка файлов, применение трансформаций к изображениям) с замером ускорения	СЗ	ОПК-2.3, ОПК-3.3, FC-1.3
		1.5	Многопроцессная обработка данных	multiprocessing: Pool, map, starmap, apply_async. Обмен данными между процессами: Pipe, Queue, Value, Array, Manager. Shared memory (multiprocessing.shared_memory). Практика: параллельная предобработка датасета (нормализация, аугментация) с multiprocessing.Pool. Сравнение с последовательной обработкой	СЗ	ОПК-2.3, FC-1.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
		1.6	Параллельная обработка данных с помощью NumPy и Pandas	Векторизация NumPy как неявный параллелизм (BLAS, LAPACK, MKL). Настройка числа потоков: OMP_NUM_THREADS, MKL_NUM_THREADS. Параллельные операции в Pandas: Dask DataFrame (обзор), modin (обзор), pandarallel. Практика: сравнение скорости цикла, vectorized NumPy и параллельного Pandas на задаче обработки большого датасета	СЗ	FC-1.3, ОПК-2.3
		1.7	Практикум: параллельный пайплайн предобработки данных для ML	Сквозная задача: загрузка большого набора изображений → параллельное изменение размера и аугментация (ProcessPoolExecutor) → параллельное вычисление статистик (multiprocessing.Pool) → сохранение результатов. Замер ускорения. Анализ: закон Амдала на практике, узкие места (I/O vs. CPU). Документирование требований к производительности	СЗ	FC-1.3, ОПК-3.3, ПК-1.1
Раздел 2	Асинхронное программирование и сетевое взаимодействие	2.1	Асинхронное программирование: asyncio	Модель событийного цикла (event loop). Корутины: async def, await. Отличие от потоков: кооперативная многозадачность. asyncio.run, asyncio.gather, asyncio.create_task. Асинхронные генераторы и контекстные менеджеры. Когда использовать asyncio: IO-bound задачи, сетевые запросы, веб-серверы	СЗ	ОПК-2.3, ОПК-3.3
		2.2	Асинхронный HTTP и работа с API	Библиотека aiohttp: асинхронные HTTP-запросы (ClientSession, get, post). Семафоры для ограничения числа одновременных запросов. Обработка ошибок и таймауты. Практика: асинхронный сбор данных из REST API (параллельная загрузка данных из нескольких источников). Сравнение с синхронным requests	СЗ	ОПК-2.3, ОПК-3.3
		2.3	Архитектуры сетевого взаимодействия: REST, gRPC, очереди	REST API: принципы, HTTP-методы, JSON, FastAPI (обзор). gRPC: Protocol Buffers, определение сервисов, генерация кода, потоковая передача. Очереди сообщений: RabbitMQ, Redis Pub/Sub (обзор). Паттерны: запрос-ответ, публикация-подписка, очередь задач. Критерии выбора: латентность, пропускная способность, надёжность	СЗ	ОПК-2.1, ПК-2.1
		2.4	Микросервисная архитектура для ИИ-систем	Монолит vs. микросервисы: преимущества и недостатки. Микросервисы для ML: сервис данных, сервис обучения, сервис инференса, сервис мониторинга. Контейнеризация	СЗ	ПК-2.1, ОПК-2.1

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				микросервисов (Docker). Оркестрация (Docker Compose, Kubernetes — обзор). API Gateway. Service discovery. Связь с паттернами промышленных ИИ-систем		
		2.5	Практикум: асинхронный сервис инференса	Реализация асинхронного веб-сервиса предсказаний на FastAPI: загрузка модели, endpoint predict (async), batch predict. Нагрузочное тестирование (locust / wrk). Сравнение производительности: синхронный vs. асинхронный обработчик. Ограничение параллелизма. Обработка ошибок и таймауты	СЗ	PL-1.3, ОПК-3.3, ПК-2.1
		2.6	Практикум: gRPC-сервис для модели ML	Определение сервиса в .proto: PredictRequest, PredictResponse. Генерация Python-кода (grpc_tools). Реализация сервера и клиента. Поточковая передача (server streaming). Замер латентности. Сравнение gRPC и REST по скорости и объёму передаваемых данных. Практика: развёртывание в Docker-контейнере	СЗ	PL-1.3, ОПК-2.3, ПК-2.1
		2.7	Практикум: очередь задач для ML-пайплайна	Celery + Redis: определение задач, запуск воркеров, отправка задач, получение результатов. Практика: асинхронная обработка запросов на обучение модели (задача ставится в очередь, воркер выполняет обучение, результат возвращается). Мониторинг задач (Flower, обзор). Обсуждение: когда использовать очереди vs. синхронный вызов	СЗ	PL-1.3, ПК-2.1, ОПК-2.3
Раздел 3	Распределённые вычисления и обработка данных	3.1	Распределённые вычисления: модели и паттерны	Распределённые системы: определение, свойства (прозрачность, отказоустойчивость, масштабируемость). Теорема CAP. Консенсус (Raft, Paxos — обзор концепции). Паттерн MapReduce: принцип, этапы (Map, Shuffle, Reduce), примеры. Связь с распределённым обучением моделей ИИ	СЗ	ОПК-2.1, ПК-2.1
		3.2	MPI: обмен сообщениями между процессами	Модель передачи сообщений (Message Passing Interface). mpi4py: инициализация (MPI.COMM_WORLD), rank, size. Точечные операции: Send, Recv, send, recv. Коллективные операции: Bcast, Scatter, Gather, Reduce, Allreduce. Практика: параллельное вычисление числа $\pi$ методом Монте-Карло с MPI	СЗ	ОПК-2.3, ОПК-2.1
		3.3	MPI для параллельной обработки данных	Распределение данных между процессами: Scatter для разбиения массива. Параллельная обработка (каждый	СЗ	ОПК-2.3, FC-1.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				процесс обрабатывает свою часть). Сбор результатов: Gather, Reduce. Практика: параллельное вычисление статистик (среднее, дисперсия) по частям большого датасета с MPI. Замер ускорения и эффективности		
		3.4	Распределённая обработка данных: Dask	Dask: архитектура (scheduler, workers, client). Dask Array: параллельный аналог NumPy. Dask DataFrame: параллельный аналог Pandas. Dask Delayed: ленивые вычисления, граф задач. Практика: обработка датасета, не помещающегося в память одного узла. Визуализация графа задач. Сравнение с Pandas	СЗ	FC-1.3, ОПК-2.3
		3.5	Распределённая обработка данных: Ray (обзор)	Ray: архитектура (driver, workers, object store). Ray Core: remote functions (@ray.remote), remote actors. Ray Data: распределённые датасеты. Ray Train и Ray Serve (обзор). Сравнение Dask и Ray: модели выполнения, API, экосистемы. Практика: распределённая предобработка данных с Ray	СЗ	FC-1.3, ОПК-2.3, PL-1.3
		3.6	Практикум: распределённый ETL-пайплайн	Сквозная задача: загрузка данных из нескольких источников (Dask/Ray) → параллельная очистка и трансформация → агрегация → сохранение результатов. Работа с данными, превышающими объём оперативной памяти. Замеры производительности. Анализ узких мест. Документирование архитектуры пайплайна и требований к ресурсам	СЗ	FC-1.3, ПК-1.1, ПК-2.1
Раздел 4	Параллельное и распределённое обучение моделей ИИ	4.1	Параллельное обучение: data parallelism	Стратегии параллельного обучения: data parallelism, model parallelism, pipeline parallelism. Data parallelism: каждый GPU обрабатывает свой батч, градиенты усредняются (AllReduce). PyTorch DataParallel (dp) и DistributedDataParallel (DDP). Синхронный и асинхронный SGD. Масштабирование learning rate. Практика: обучение CNN с DDP на нескольких процессах	СЗ	FC-1.3, ОПК-2.3
		4.2	PyTorch DistributedDataParallel: практика	Настройка DDP: init_process_group, DistributedSampler, DistributedDataParallel. Запуск: torchrun, torch.distributed.launch. Обучение на нескольких GPU (или имитация на CPU). Сохранение и загрузка чекпоинтов в DDP. Замеры: throughput (samples/sec), scaling efficiency.	СЗ	FC-1.3, ОПК-3.3

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы *	Формируемые индикаторы
			Типичные ошибки и отладка		
		4.3 Model parallelism и pipeline parallelism (обзор)	Model parallelism: размещение разных слоёв на разных GPU. Проблема: последовательность слоёв → простой GPU. Pipeline parallelism: разделение модели на стадии, микробатчи. GPipe, PipeDream (обзор). Tensor parallelism: разделение одного слоя (Megatron-LM, обзор). ZeRO (DeepSpeed, обзор): распределение состояния оптимизатора. Когда какой подход использовать	СЗ	FC-1.3, ОПК-2.1
		4.4 Параллельный DataLoader и ускорение подготовки данных	PyTorch DataLoader: num_workers, pin_memory, prefetch_factor. Узкое горлышко: CPU-bound предобработка при GPU-обучении. DALI (NVIDIA Data Loading Library, обзор). WebDataset для потоковой загрузки больших датасетов. Практика: профилирование DataLoader (torch.utils.bottleneck), оптимизация num_workers, замер влияния на общее время обучения	СЗ	FC-1.3, ОПК-3.3
		4.5 Развёртывание моделей с параллельной обработкой запросов	Инференс-сервер: параллельная обработка запросов. Батчирование запросов (dynamic batching). Triton Inference Server (обзор): поддержка множества фреймворков, модельный ансамбль, масштабирование. TorchServe (обзор). Практика: FastAPI + asyncio + ProcessPoolExecutor для параллельного инференса. Нагрузочное тестирование, анализ пропускной способности	СЗ	PL-1.3, ПК-2.1
		4.6 Практикум: итоговый проект — распределённый ML-пайплайн	Финальная интеграция: распределённая предобработка данных (Dask/Ray) → параллельное обучение модели (DDP) → асинхронный сервис инференса (FastAPI) → очередь задач (Celery). Docker Compose для всех компонентов. Замеры производительности на каждом этапе. Документирование архитектуры, обоснование выбора технологий, анализ ограничений масштабируемости. Презентация	СЗ	FC-1.3, PL-1.3, ПК-2.1, ПК-1.1

\* - 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)
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. Малявко, А. А. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi: учебник для вузов / А. А. Малявко. — 3-е изд., испр. и доп. — Москва: Издательство Юрайт, 2025. — 135 с. — (Высшее образование). — ISBN 978-5-534-14116-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562821>

### Дополнительная литература:

1. Лупин, С. А. Технологии параллельного программирования : учебное пособие / С.А. Лупин, М.А. Посыпкин. — Москва : ФОРУМ : ИНФРА-М, 2026. — 206 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0853-2. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2222136>

### 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](http://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\*:*

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