

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ястребов Олег Александрович

Должность: Ректор

Дата подписания: 22.05.2026 14:55:10

Уникальный программный ключ:

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

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++

(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 «Программирование на языке C++» входит в программу бакалавриата «Искусственный интеллект: разработка и обучение интеллектуальных систем» по направлениям подготовки 02.03.02 Фундаментальная информатика и информационные технологии и 09.03.03 Прикладная информатика, и изучается в 4, 5 семестрах 2, 3 курсов. Дисциплину реализует Кафедра прикладного искусственного интеллекта. Дисциплина состоит из 8 разделов и 68 тем и направлена на изучение языка программирования C++ как средства разработки высокопроизводительных программных систем: типов данных и управляющих конструкций, указателей и управления памятью, объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм), обобщённого программирования (шаблоны), стандартной библиотеки (STL), средств современного C++ (C++17/C++20), паттернов проектирования, инструментов сборки, тестирования и профилирования, а также применения C++ для реализации высокопроизводительных компонентов систем ИИ (движки инференса, оптимизированные алгоритмы обработки данных).

Целью освоения дисциплины является формирование у студентов системных знаний и практических навыков разработки программных решений на языке C++ для задач обработки данных и систем искусственного интеллекта, включая владение объектно-ориентированным и обобщённым программированием, стандартной библиотекой, паттернами проектирования модульных и масштабируемых систем, навыками компиляции, отладки, тестирования и профилирования программ, а также способность разрабатывать технические спецификации на программные компоненты и проектировать архитектуру ИС с использованием типовых архитектурных шаблонов.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Программирование на языке C++» 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-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	ОПК-3.1 Знает основные алгоритмы и структуры данных, парадигмы программирования, принципы проектирования программных систем и баз данных; ОПК-3.2 Умеет разрабатывать программные решения на языках Python и C++ для задач обработки данных, машинного обучения и глубокого обучения, проектировать и реализовывать базы данных; ОПК-3.3 Владеет навыками создания, тестирования и отладки алгоритмических и программных решений для систем ИИ, включая разработку пайплайнов обработки данных и обучения моделей;
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	ПК-1.2 Разрабатывает технические спецификации на программные компоненты систем ИИ и описывает их взаимодействие;
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и	ПК-2.1 Проектирует архитектуру ИС с компонентами ИИ, выбирает архитектурные паттерны и технологический стек;

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
	базы данных таких систем	

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Программирование на языке C++» 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 «Программирование на языке C++».

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-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	Дискретная математика; История и теория программирования; Алгоритмы и структуры данных; Введение в базы данных; Программирование на языке Python; Технологическая (проектно-технологическая) практика (учебная);	Методы машинного обучения; Нейронные сети; Технологическая (проектно-технологическая) практика (производственная);
ПК-1	Способен анализировать требования к программному обеспечению систем ИИ, разрабатывать технические спецификации и техническое задание на систему	Технологическая (проектно-технологическая) практика (учебная); Правоведение; Введение в искусственный интеллект; Искусственный интеллект и когнитивная психология; Этика и безопасность использования искусственного интеллекта; История и теория программирования; Введение в базы данных;	Эксплуатационная практика (производственная); Преддипломная практика; Технологическая (проектно-технологическая) практика (производственная); 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

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
			естественного языка (NLP);
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и базы данных таких систем	Алгоритмы и структуры данных; Программирование на языке Python; Введение в базы данных; Технологическая (проектно-технологическая) практика (учебная);	Массово-параллельные вычисления в машинном обучении (GPU); MLOps и промышленная разработка систем искусственного интеллекта; Практическая подготовка на проектах отраслевых 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 (производственная);

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

** - элективные дисциплины /практики

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Программирование на языке С++» составляет «8» зачетных единиц.

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

Вид учебной работы	ВСЕГО, ак.ч.		Семестр(-ы)	
			4	5
<i>Контактная работа, ак.ч.</i>	136		68	68
Лекции (ЛК)	0		0	0
Лабораторные работы (ЛР)	0		0	0
Практически/семинарские занятия (СЗ)	136		68	68
<i>Самостоятельная работа обучающихся, ак.ч.</i>	125		76	49
<i>Контроль (экзамен/зачет с оценкой), ак.ч.</i>	27		0	27
Общая трудоемкость дисциплины	ак.ч.	288	144	144
	зач.ед.	8	4	4

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

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

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
Раздел 1	Основы языка C++: типы данных, переменные, управляющие конструкции	1.1	Введение в C++. Компиляция и среда разработки	Место C++ в экосистеме ИИ (движки инференса, ONNX Runtime, TensorRT, libtorch). Стандарты языка (C++11/14/17/20). Процесс компиляции: препроцессор, компилятор, компоновщик. Настройка среды разработки (GCC/Clang, IDE). Первая программа, ввод-вывод через iostream	СЗ	ОПК-3.1, ОПК-3.2
		1.2	Целочисленные типы и арифметические операции	Типы: int, short, long, long long, unsigned. Размеры типов и sizeof. Арифметические операции, приоритет операций. Переполнение целых чисел. Литералы и суффиксы. Связь с типами данных в тензорных вычислениях (int8, int32)	СЗ	ОПК-3.1, ОПК-3.2
		1.3	Типы с плавающей запятой, bool, символьные типы	float, double, long double: точность, диапазон, особенности сравнения. Тип bool и логические операции. Тип char, таблица ASCII, кодировки. Приведение типов: implicit, static cast. Автоматический вывод типа (auto)	СЗ	ОПК-3.1, ОПК-3.2
		1.4	Условные операторы и ветвление	Оператор if/else, вложенные условия. Тернарный оператор. Оператор switch/case: синтаксис, break, fallthrough. Область видимости переменных в блоках. Задачи на ветвление с примерами из обработки данных	СЗ	ОПК-3.1, ОПК-3.2
		1.5	Циклы и итерации	Циклы for, while, do-while. Операторы break и continue. Вложенные циклы. Range-based for (обзор). Задачи на циклы: обработка последовательностей, поиск элементов, вычисление статистик	СЗ	ОПК-3.1, ОПК-3.2
		1.6	Массивы в стиле C и строки	Статические массивы: объявление, инициализация, доступ по индексу. Многомерные массивы. C-строки (char[]): функции strlen, strcpy, strcmp. Буферные переполнения и безопасность. Связь массивов с указателями (обзор)	СЗ	ОПК-3.1, ОПК-3.2
		1.7	Контейнеры std::array и std::string	Контейнер std::array: преимущества над C-массивами, методы. Класс std::string: создание, конкатенация, поиск подстроки, методы size, substr, find, replace. Сравнение строк. Конвертация строк и чисел (stoi, to_string)	СЗ	ОПК-3.1, ОПК-3.2
		1.8	Консольный ввод-вывод и форматирование	Потоки cin и cout. Методы getline для чтения строк. Манипуляторы форматирования: setw, setprecision, fixed,	СЗ	ОПК-3.2, ОПК-3.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				hex. Обработка ошибок ввода (cin.fail, cin.clear, cin.ignore). Задачи на парсинг пользовательского ввода		
		1.9	Практикум: калькулятор статистик для числовой последовательности	Сквозная задача: чтение последовательности чисел с консоли, вычисление среднего, минимума, максимума, медианы. Обработка некорректного ввода. Форматированный вывод результатов. Ручная сортировка массива для нахождения медианы	СЗ	ОПК-3.2, ОПК-3.3
Раздел 2	Указатели, управление памятью, функции	2.1	Указатели: определение и адресная арифметика	Понятие адреса и указателя. Оператор взятия адреса (&) и разыменования (*). Арифметика указателей: инкремент, декремент, разность. Связь указателей и массивов. Нулевой указатель (nullptr). Указатель на константу и константный указатель	СЗ	ОПК-3.1, ОПК-3.2
		2.2	Ссылки и передача параметров	Определение ссылки, отличия от указателей. Передача параметров в функции: по значению, по ссылке, по указателю. Const-ссылки для защиты от изменения. Возврат значений из функций: по значению, по ссылке. Опасности возврата ссылки на локальную переменную	СЗ	ОПК-3.1, ОПК-3.2
		2.3	Динамическое управление памятью	Стек и куча: различия, время жизни объектов. Операторы new/delete, new[]/delete[]. Проблемы ручного управления: утечки памяти, висячие указатели, двойное освобождение. Диагностика утечек (Valgrind, обзор)	СЗ	ОПК-3.1, ОПК-3.2, ОПК-3.3
		2.4	Умные указатели и принцип RAII	Принцип RAII: захват ресурса при инициализации. std::unique_ptr: исключительное владение, перемещение. std::shared_ptr: совместное владение, счётчик ссылок. std::weak_ptr: неовладеющий наблюдатель. Связь с управлением памятью GPU-тензоров в фреймворках ИИ	СЗ	ОПК-3.1, ОПК-3.2, ОПК-3.3
		2.5	Функции: объявление, определение, перегрузка	Объявление и определение функций. Прототипы. Перегрузка функций: правила разрешения. Аргументы по умолчанию. Inline-функции. Рекурсия: базовые примеры, стек вызовов. Указатели на функции (обзор)	СЗ	ОПК-3.1, ОПК-3.2
		2.6	Модульность: заголовочные файлы и отдельная компиляция	Разделение объявления (.h) и реализации (.cpp). Заголовочные файлы и include-guards (#pragma once). Раздельная компиляция. Пространства имён (namespace). Организация кода в несколько файлов. Структура проекта на C++	СЗ	ОПК-3.1, ПК-1.2

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
		2.7	Препроцессор и система сборки CMake	Директивы препроцессора: #include, #define, #ifdef/#ifndef. Макросы и их опасности. Условная компиляция. Этапы сборки. Введение в CMake: CMakeLists.txt, add_executable, target_link_libraries. Статические и динамические библиотеки	СЗ	ОПК-3.1, ОПК-3.3, ПК-1.2
		2.8	Практикум: динамический массив с ручным управлением памятью	Реализация класса DynamicArray: выделение памяти через new[], изменение размера, копирование, освобождение. Демонстрация утечек при отсутствии delete[]. Рефакторинг на std::unique_ptr. Сравнение с std::vector	СЗ	ОПК-3.2, ОПК-3.3
		2.9	Практикум: модульная утилита обработки CSV-файла	Сквозная задача: разработка консольной утилиты для чтения CSV-файла, хранения данных, вычисления статистик. Организация кода в модули (data_loader.h/.cpp, stats.h/.cpp, main.cpp). Сборка через CMake. Обработка ошибок ввода	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
Раздел 3	Объектно-ориентированное программирование (основы)	3.1	Классы и объекты: инкапсуляция	Определение класса: поля, методы, спецификаторы доступа (public, private, protected). Конструкторы: по умолчанию, с параметрами, делегирующие. Деструкторы. Списки инициализации. Ключевое слово this. Const-методы	СЗ	ОПК-3.1, ОПК-3.2, ПК-1.2
		3.2	Перегрузка операторов	Перегрузка арифметических операторов (+, -, *). Перегрузка операторов сравнения (==, <). Перегрузка потокового ввода-вывода (<<, >>). Перегрузка оператора индексирования []. Перегрузка как метод класса и как свободная функция (friend)	СЗ	ОПК-3.1, ОПК-3.2
		3.3	Копирование и перемещение объектов	Правило трёх: копирующий конструктор, оператор присваивания, деструктор. Глубокое vs. поверхностное копирование. Семантика перемещения: rvalue-ссылки (&&), конструктор перемещения, оператор перемещения. Правило пяти. std::move	СЗ	ОПК-3.1, ОПК-3.2
		3.4	Наследование	Виды наследования (public, protected, private). Доступ к членам базового класса. Конструкторы и деструкторы при наследовании: порядок вызова. Множественное наследование: проблема ромба, виртуальное наследование. Композиция vs. наследование	СЗ	ОПК-3.1, ОПК-3.2, ПК-2.1
		3.5	Полиморфизм и виртуальные функции	Статический полиморфизм (перегрузка) vs. динамический (виртуальные функции). Ключевое слово virtual, override,	СЗ	ОПК-3.1, ОПК-3.2,

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				final. Таблица виртуальных функций (vtable). Абстрактные классы и чисто виртуальные функции. Интерфейсы в C++. Виртуальные деструкторы		ПК-2.1
		3.6	Обработка исключений	Механизм try/catch/throw. Стандартные исключения (std::exception, std::runtime_error, std::invalid_argument). Создание пользовательских исключений. Гарантии безопасности: базовая, строгая, nothrow. Принцип RAII для безопасного управления ресурсами при исключениях	СЗ	ОПК-3.1, ОПК-3.3
		3.7	Практикум: реализация матричного класса	Класс Matrix: двумерное хранение, конструкторы, перегрузка операторов (сложение, вычитание, умножение, транспонирование). Проверка размерностей с выбросом исключений. Copy и move семантика. Связь с тензорными операциями в фреймворках ИИ	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
		3.8	Практикум: иерархия классов для ML-пайплайна	Проектирование иерархии: абстрактный класс Model (методы fit, predict), наследники LinearRegression, KNNClassifier. Класс Dataset с загрузкой данных. Класс Metrics с методами оценки. Демонстрация полиморфизма: единый интерфейс для разных моделей	СЗ	ОПК-3.2, ОПК-3.3, ПК-2.1
Раздел 4	Объектно-ориентированное программирование (продвинутое уровень)	4.1	Принципы проектирования SOLID	Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion — определение, примеры нарушений и корректного применения на C++. Связь с качеством архитектуры ПО. Рефакторинг кода с применением SOLID	СЗ	ОПК-3.1, ПК-2.1, ПК-1.2
		4.2	Статические члены класса и перечисления	Статические поля и методы: определение, инициализация, область применения. Паттерн «Синглтон» (обзор и критика). Перечисления enum и enum class. Scoped enumerations. Использование enum class для типобезопасности	СЗ	ОПК-3.1, ОПК-3.2
		4.3	Агрегация, композиция и делегирование	Отношения между объектами: «имеет» vs. «является». Агрегация: объект использует другой объект. Композиция: объект владеет другим. Делегирование: перенаправление вызовов. Примеры: DataLoader владеет Dataset, Model использует Optimizer	СЗ	ОПК-3.1, ПК-2.1
		4.4	Практикум: реализация линейной регрессии на C++	Класс LinearRegression: хранение весов, метод fit (градиентный спуск), метод predict. Класс	СЗ	ОПК-3.2, ОПК-3.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				GradientDescentOptimizer как отдельный компонент. Перегрузка оператора вывода для параметров модели. Обработка исключений. Тестирование на синтетических данных		
		4.5	Практикум: реализация k-NN классификатора на C++	Класс KNNClassifier: хранение обучающей выборки, метод predict (поиск k ближайших соседей), функция расстояния как параметр. Наследование от абстрактного класса Model. Тестирование на наборе Iris (CSV-загрузка)	СЗ	ОПК-3.2, ОПК-3.3, ПК-2.1
		4.6	Практикум: класс DataLoader и Pipeline	Класс DataLoader: чтение CSV, разбиение на train/test, итератор по мини-батчам. Класс Pipeline: последовательность шагов (загрузка → предобработка → обучение → оценка). Композиция объектов. Документирование интерфейсов	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
		4.7	Практикум: спецификация интерфейсов и документация	Описание интерфейсов классов: входы/выходы, предусловия, постусловия. Документирование в стиле Doxygen. Генерация документации. Связь спецификации интерфейсов с разработкой технической документации на программные компоненты	СЗ	ПК-1.2, ОПК-3.3
		4.8	Практикум: итоговый проект 4 семестра — ML-библиотека (часть 1)	Сквозная задача: объединение компонентов (Matrix, DataLoader, LinearRegression, KNN, Metrics, Pipeline) в единую библиотеку. Организация проекта: CMake, структура директорий (include/, src/, tests/). Подготовка к расширению во 2 семестре	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2, ПК-2.1
Раздел 5	Шаблоны и основы обобщённого программирования	5.1	Шаблоны функций	Определение шаблона функции. Инстанцирование: явное и неявное. Вывод типов (type deduction). Специализация шаблонов функций. Шаблоны с несколькими параметрами. SFINAE (обзор). Практические примеры: шаблонные функции сортировки, поиска	СЗ	ОПК-3.1, ОПК-3.2
		5.2	Шаблоны классов	Определение шаблона класса. Полная и частичная специализация. Шаблоны с нетиповыми параметрами (template<int N>). Шаблоны и наследование. Зависимые имена (typename). Variadic templates (обзор)	СЗ	ОПК-3.1, ОПК-3.2
		5.3	Итераторы: концепция и категории	Обобщённая концепция итератора. Категории: input, output, forward, bidirectional, random access. Связь итераторов с указателями. Разыменование, инкремент, сравнение.	СЗ	ОПК-3.1, ОПК-3.2

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы *	Формируемые индикаторы
			Итераторы begin/end. Константные итераторы (cbegin, cend)		
		5.4 Контейнеры STL: std::vector	std::vector: внутреннее устройство (динамический массив), амортизированная сложность push_back. Методы: size, capacity, reserve, resize, shrink_to_fit, emplace_back. Инвалидация итераторов при перераллокации. Профилирование: reserve vs. без reserve	СЗ	ОПК-3.1, ОПК-3.2
		5.5 Контейнеры STL: std::list, std::deque, std::forward_list	std::list: двусвязный список, O(1) вставка/удаление, O(n) доступ. std::deque: двусторонняя очередь, доступ O(1). std::forward_list: односвязный список. Сравнение контейнеров по операциям. Критерии выбора контейнера	СЗ	ОПК-3.1, ОПК-3.2
		5.6 Контейнеры STL: ассоциативные (std::set, std::map)	std::set, std::map: красно-чёрное дерево, сложность O(log n). std::multiset, std::multimap. Пользовательские компараторы. Structured bindings для итерации по map (C++17). Примеры: подсчёт частот слов, группировка данных	СЗ	ОПК-3.1, ОПК-3.2
		5.7 Контейнеры STL: неупорядоченные (std::unordered_map, std::unordered_set)	Хеш-таблицы: принцип работы, коллизии, load factor. std::unordered_set, std::unordered_map: сложность O(1) среднее, O(n) худшее. Пользовательские хеш-функции. Когда предпочесть unordered контейнеры упорядоченным	СЗ	ОПК-3.1, ОПК-3.2
		5.8 Практикум: шаблонный класс Vector<T>	Реализация шаблонного класса Vector<T>: хранение данных, push_back с реаллокацией, оператор [], итератор (begin/end). Move-семантика для эффективности. Поддержка произвольных числовых типов (float, double, int). Юнит-тесты	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
		5.9 Практикум: шаблонный класс Matrix<T>	Рефакторинг матричного класса из 4 семестра в шаблонный Matrix<T>. Поддержка float, double, int. Специализация вывода для разных типов. Операции: сложение, умножение, транспонирование. Сравнение производительности Matrix<float> vs. Matrix<double>	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
Раздел 6	Алгоритмы STL, Лямбда-выражения и современный C++	6.1 Алгоритмы STL: немодифицирующие	std::find, std::find_if, std::count, std::count_if, std::accumulate, std::all_of, std::any_of, std::none_of, std::min_element, std::max_element. Использование с предикатами. Примеры: анализ набора данных	СЗ	ОПК-3.1, ОПК-3.2
		6.2 Алгоритмы STL: модифицирующие и сортировки	std::sort, std::stable_sort, std::partial_sort, std::nth_element. std::transform, std::copy, std::copy_if, std::remove_if, std::unique. std::for_each. Идиома erase-remove. Примеры:	СЗ	ОПК-3.1, ОПК-3.2

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы *	Формируемые индикаторы
			фильтрация и трансформация данных		
		6.3	Лямбда-выражения и функциональные объекты	СЗ	ОПК-3.1, ОПК-3.2
		6.4	Файловый ввод-вывод	СЗ	ОПК-3.2, ОПК-3.3
		6.5	Средства современного C++ (C++17)	СЗ	ОПК-3.1, ОПК-3.2
		6.6	Средства современного C++ (C++20, обзор)	СЗ	ОПК-3.1, ОПК-3.2
		6.7	Практикум: обработка датасета с использованием STL и лямбд	СЗ	ОПК-3.2, ОПК-3.3
		6.8	Практикум: реализация пайплайна предобработки данных на STL	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2
		6.9	Практикум: сериализация и десериализация моделей	СЗ	ОПК-3.2, ОПК-3.3
Раздел 7	Паттерны проектирования,	7.1	Паттерны проектирования: порождающие	СЗ	ОПК-3.1, ПК-2.1,

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы *	Формируемые индикаторы	
	тестирование и отладка		фабрика». Связь с созданием ML-моделей: ModelFactory, создающая LinearRegression или KNN по строковому имени		ПК-1.2	
		7.2	Паттерны проектирования: структурные и поведенческие	Паттерн «Стратегия»: семейство взаимозаменяемых алгоритмов (стратегии оптимизации: SGD, Adam). Паттерн «Наблюдатель»: логирование метрик обучения. Паттерн «Шаблонный метод»: скелет алгоритма обучения. Паттерн «Декоратор»: обёртки над моделями	СЗ	ОПК-3.1, ПК-2.1
		7.3	Архитектурные подходы к проектированию ИС	Модульная архитектура. Разделение на слои: данные, логика, представление. Принцип инверсии зависимостей на практике. Проектирование API: согласованность, минимальность, документирование. Связь с проектированием ML-пайплайнов	СЗ	ОПК-3.1, ПК-2.1, ПК-1.2
		7.4	Юнит-тестирование: Google Test	Принципы юнит-тестирования: изоляция, повторяемость, автоматизация. Фреймворк Google Test: TEST, ASSERT_EQ, ASSERT_NEAR, EXPECT, тестовые фикстуры (TEST_F). Организация тестов. Интеграция с CMake (add_test, ctest). Тестирование граничных случаев и исключений	СЗ	ОПК-3.3, ПК-1.2
		7.5	Тестирование ML-компонентов	Специфика тестирования ML-кода: детерминизм (фиксация seed), тестирование с допусками (ASSERT_NEAR). Тестирование инвариантов: размерности выхода, непустота предсказаний. Регрессионные тесты: сохранение предсказаний эталонной модели. Мутационное тестирование (обзор)	СЗ	ОПК-3.3, ПК-1.2
		7.6	Отладка: GDB и санитайзеры	Отладчик GDB/LLDB: точки останова, пошаговое выполнение, просмотр переменных и стека вызовов. Условные точки останова. Санитайзеры: AddressSanitizer (утечки, переполнения буфера), UndefinedBehaviorSanitizer. Компиляция с флагами -fsanitize	СЗ	ОПК-3.3
		7.7	Статический анализ и Valgrind	Статический анализ: clang-tidy (правила, автоисправления), cppcheck. Valgrind (memcheck): обнаружение утечек, чтение неинициализированной памяти, ошибки доступа. Практика: анализ учебного проекта инструментами статического и динамического анализа	СЗ	ОПК-3.3
		7.8	Практикум: полное покрытие ML-	Написание юнит-тестов для всех компонентов ML-	СЗ	ОПК-3.3,

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
			библиотеки тестами	библиотеки (Matrix, DataLoader, LinearRegression, KNN, Pipeline). Интеграция в CMake. Запуск через ctest. Проверка санитайзерами. Оценка покрытия кода (gcov/lcov, обзор)		ПК-1.2, ПК-2.1
Раздел 8	Профилирование, многопоточность, интеграция с Python	8.1	Профилирование и оптимизация производительности	Инструменты профилирования: gprof, perf, Valgrind (callgrind). Анализ горячих точек (hotspots). Оптимизация: выбор структур данных, минимизация аллокаций, кэш-дружественность (data locality). Benchmarking: Google Benchmark. Связь с оптимизацией инференса	СЗ	ОПК-3.3, ПК-2.1
		8.2	Оптимизация матричных операций	Профилирование матричного умножения: наивная реализация vs. блочное умножение. Влияние кэш-линий. Транспонирование для улучшения локальности. Использование флагов компиляции (-O2, -O3, -march=native). Сравнение с BLAS (обзор)	СЗ	ОПК-3.3, ОПК-3.2
		8.3	Многопоточность: std::thread и синхронизация	Потоки: std::thread, передача аргументов, join и detach. Мьютексы: std::mutex, std::lock_guard, std::unique_lock. Проблемы: гонки данных, дедлоки. Атомарные операции: std::atomic. Практика: параллельное вычисление статистик по столбцам матрицы	СЗ	ОПК-3.1, ОПК-3.2
		8.4	Многопоточность: задачи и асинхронность	std::async и std::future: асинхронный запуск задач. std::promise. Пул потоков (обзор концепции). Параллельные алгоритмы STL (std::execution::par, C++17). Связь с параллельной обработкой данных и батчей в ML-пайплайнах	СЗ	ОПК-3.1, ОПК-3.2
		8.5	Взаимодействие C++ и Python: pybind11	Мотивация: реализация критичных по производительности компонентов на C++ с вызовом из Python. Библиотека pybind11: экспорт функций, классов, перегруженных методов. Передача NumPy-массивов. Сборка расширений через CMake	СЗ	ОПК-3.2, ОПК-3.3, ПК-2.1
		8.6	Практикум: экспорт ML-библиотеки в Python через pybind11	Экспорт классов Matrix, LinearRegression, KNN в Python-модуль. Обёртка для DataLoader с поддержкой pandas DataFrame. Тестирование из Python: сравнение результатов C++-реализации с scikit-learn. Замер ускорения	СЗ	ОПК-3.2, ОПК-3.3, ПК-2.1
		8.7	Практикум: параллельная обработка данных	Реализация параллельного DataLoader: загрузка и предобработка данных в фоновых потоках. Многопоточное матричное умножение. Замер ускорения (speedup)	СЗ	ОПК-3.2, ОПК-3.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы *	Формируемые индикаторы
				относительно однопоточной версии. Анализ: закон Амдала на практике		
		8.8	Практикум: итоговый проект — ML-фреймворк на C++	Финальная интеграция: полноценная ML-библиотека на C++ с шаблонными контейнерами, моделями (LinearRegression, KNN), DataLoader, Pipeline, паттернами (фабрика, стратегия), юнит-тестами (gtest), профилированием, многопоточностью, экспортом в Python. Документация (Doxygen). Презентация проекта	СЗ	ОПК-3.2, ОПК-3.3, ПК-1.2, ПК-2.1

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

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Таблица 6.1. Материально-техническое обеспечение дисциплины

Тип аудитории	Оснащение аудитории	Специализированное учебное/лабораторное оборудование, ПО и материалы для освоения дисциплины (при необходимости)
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. Огнева, М. В. Программирование на языке С++: практический курс: учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563618>

2. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке С++: учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2024. — 512 с. + Доп. материалы [Электронный ресурс]. — ISBN 978-5-8199-0699-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/2083383>

Дополнительная литература:

1. Немцова Тамара Игоревна, Голова Светлана Юрьевна, Терентьев Алексей Игоревич. Программирование на языке высокого уровня. Программирование на языке С++. учебное пособие / под редакцией Л.Г. Гагариной [Электронный ресурс]. - М.: ИНФРА-М, 2023. 512 с. ISBN 978-5-16-013214-3 URL: https://mega.rudn.ru/MegaPro/UserEntry?Action=Link_FindDoc&id=508904&idb=0

2. . Панкратов Александр Серафимович, Салпагаров Солтан Исмаилович. Технология программирования на языке С++: динамические структуры, объекты, классы. учебное пособие [Электронный ресурс]. – М.: РУДН, 2021. 73 с. ISBN 978-5-209-10906-8 URL: https://mega.rudn.ru/MegaPro/UserEntry?Action=Link_FindDoc&id=504558&idb=0

Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. ЭБС РУДН и сторонние ЭБС, к которым студенты университета имеют доступ

на основании заключенных договоров

- Электронно-библиотечная система РУДН – ЭБС РУДН

<https://mega.rudn.ru/MegaPro/Web>

- ЭБС «Университетская библиотека онлайн» <http://www.biblioclub.ru>

- ЭБС «Юрайт» <http://www.biblio-online.ru>

- ЭБС «Консультант студента» www.studentlibrary.ru

- ЭБС «Знаниум» <https://znanium.ru/>

2. Базы данных и поисковые системы

- Sage <https://journals.sagepub.com/>

- Springer Nature Link <https://link.springer.com/>

- Wiley Journal Database <https://onlinelibrary.wiley.com/>

- Научометрическая 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!**