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

Инженерная академия

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

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

(наименование дисциплины/модуля)

Рекомендована МССН для направления подготовки/специальности:

27.04.04 УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ

(код и наименование направления подготовки/специальности)

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

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, МАШИННОЕ ОБУЧЕНИЕ И КОСМИЧЕСКИЕ НАУКИ

(наименование (профиль/специализация) ОП ВО)

2026 г.

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Технологии программирования» входит в программу магистратуры «Искусственный интеллект, машинное обучение и космические науки» по направлению 27.04.04 «Управление в технических системах» и изучается в 1 семестре 1 курса. Дисциплину реализует Кафедра механики и процессов управления. Дисциплина состоит из 26 разделов и 65 тем и направлена на изучение базовых алгоритмов сортировки и поиска, алгоритмов на графах, методов динамического программирования, современных парадигм программирования, подходами к технологиям параллельного и распределенного программирования

Целью освоения дисциплины является овладение студентами практическими навыками алгоритмизации и программирования.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Технологии программирования» 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-1	Способен анализировать и выявлять естественно-научную сущность проблем управления в технических системах на основе положений, законов и методов в области естественных наук и математики	ОПК-1.1 Знает основные законы, положения и методы в области естественных наук и математики;; ОПК-1.2 Умеет выявлять естественно-научную сущность проблем управления в технических системах руководствуясь законами и методами естественных наук и математики;; ОПК-1.3 Владеет инструментами анализа проблем управления в технических системах.;
ОПК-2	Способен формулировать задачи управления в технических системах и обосновывать методы их решения	ОПК-2.1 Знает основные методы решения задач управления в технических системах;; ОПК-2.2 Умеет обосновывать методы решения задач управления в технических системах;; ОПК-2.3 Владеет методами постановки задач управления в технических системах.;
ОПК-3	Способен самостоятельно решать задачи управления в технических системах на базе последних достижений науки и техники	ОПК-3.1 Знает основные подходы к решению задач управления в технических системах;; ОПК-3.2 Умеет применять основные подходы на базе последних достижений науки и техники к решению задач управления в технических системах;; ОПК-3.3 Владеет методами решения задач управления в технических системах, основанных на последних достижениях науки и техники.;

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Programming Technology» 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 «Programming Technology».

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-1	Способен анализировать и		Undergraduate Training;

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
	выявлять естественно-научную сущность проблем управления в технических системах на основе положений, законов и методов в области естественных наук и математики		Advanced Methods of Space Flight Mechanics; Advanced Methods of Earth Remote Sensing; Geoinformation Systems and Applications;
ОПК-2	Способен формулировать задачи управления в технических системах и обосновывать методы их решения		Undergraduate Training; Dynamics and Control of Space Systems;
ОПК-3	Способен самостоятельно решать задачи управления в технических системах на базе последних достижений науки и техники		Dynamics and Control of Space Systems; Advanced Methods of Space Flight Mechanics; Research work / Научно-исследовательская работа; Undergraduate Training;

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

** - элективные дисциплины /практики

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Технологии программирования» 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)**
			1
Контактная работа, ак.ч	*34*		*34*
Лекции (**ЛК**)	17		17
Лабораторные работы (**ЛР**)	17		17
Практические/семинарские занятия (**СЗ**)	0		0
Самостоятельная работа обучающихся, ак.ч.	*218*		*218*
Контроль (экзамен/зачет с оценкой), ак.ч.	*36*		*36*
Общая трудоемкость дисциплины ак.ч.	**ак.ч.**	**288**	**288**
	зач.ед.	**8**	**8**

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*
Раздел 1	Основные элементы синтаксиса языка Python	1.1	Базовый синтаксис языка Python 3. Модель памяти и основны типы данных	Базовые правила написания кода на Python: отступы для выделения блоков, комментарии, именование переменных. Модель памяти языка с динамической типизацией и ссылочным хранением объектов. Основные типы данных: числа целые и с плавающей точкой, строки, булевы значения, списки, кортежи, словари и множества. Различие между изменяемыми и неизменяемыми типами данных.	ЛК, ЛР
		1.2	Циклы и списки. Функции.	Конструкции циклов: for для перебора последовательностей и while для повторения до выполнения условия. Способы работы со списками: индексация, срезы, добавление и удаление элементов, списковые включения. Создание функций через ключевое слово def, аргументы позиционные и именованные, возврат значения через return. Области видимости переменных: локальные и глобальные.	ЛК, ЛР
Раздел 2	Элементы теории алгоритмов	2.1	Понятие алгоритма. Машина Тьюринга. Вычислимость. Теория сложности. Возведение в степень: анализ алгоритма (умное возведение в степень).	Определение алгоритма как конечной последовательности строго определённых инструкций. Машина Тьюринга как абстрактная модель вычислителя с лентой, головкой чтения-записи и таблицей правил. Понятие вычислимости: функция считается вычислимой при наличии алгоритма для её нахождения. Теория сложности с разделением на временную и ёмкостную сложность. Алгоритм умного возведения в степень через метод двоичного разложения показателя.	ЛК, ЛР
		2.2	Задача о рюкзаке. Жадный алгоритм. Метод градиентного спуска как пример жадного алгоритма. Стратегия «Разделяй и властвуй». Рекурсивный алгоритм.	Задача о рюкзаке как выбор предметов с заданными весами и стоимостями при ограничении на общий вес. Жадный алгоритм, выбирающий на каждом шаге локально оптимальное решение. Метод градиентного спуска как пример жадного алгоритма для минимизации функции. Стратегия «Разделяй и властвуй» с разбиением задачи на подзадачи, решением каждой и объединением результатов. Рекурсивный алгоритм, вызывающий сам себя для решения подзадач.	ЛК, ЛР
Раздел 3	Парадигмы программирования. Объектноориентированное программирование	3.1	Основные принципы программирования. Процедурное программирование.	Основные принципы программирования: абстракция, декомпозиция, модульность, иерархическая организация. Процедурное программирование с акцентом на последовательное выполнение инструкций и выделение процедур. Структурное программирование как развитие процедурного с отказом от безусловных переходов.	ЛК, ЛР
		3.2	Объектно-ориентированное программирование (ООП). Функциональное программирование.	Объектно-ориентированное программирование как парадигма, организующая код вокруг объектов и их взаимодействия. Функциональное программирование с упором на чистые функции, неизменяемые данные и отсутствие побочных эффектов. Сравнение трёх парадигм: процедурной, объектно-ориентированной и функциональной.	ЛК, ЛР
		3.3	Особенности ООП. Классы и объекты. Наследование. Реализация ООП в языке Python	Особенности объектно-ориентированного программирования: инкапсуляция, наследование и полиморфизм. Классы как шаблоны для создания объектов и объекты как экземпляры классов. Наследование, позволяющее создавать новые классы на основе существующих. Реализация ООП в языке Python с синтаксисом class, методами self, конструктором init и множественным наследованием.	ЛК, ЛР
Раздел 4	Алгоритмы сортировки и поиска	4.1	Сортировка выбором. Сортировка вставками. Сортировка “Методом	Сортировка выбором с поиском минимального элемента и обменом его с текущей позицией. Сортировка вставками с построением отсортированной части путём вставки очередного элемента на нужное место. Сортировка пузырьком с последовательным	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			Пузырька". Сортировка слиянием. Быстрая сортировка	сравнением и обменом соседних элементов. Сортировка слиянием как рекурсивный алгоритм, делящий массив пополам и сливающий отсортированные половины. Быстрая сортировка с выбором опорного элемента и разделением массива на две части относительно опоры.	
		4.2	Нахождение медианы. Последовательный поиск. Методы сужения области. Сортировка в Python.	Нахождение медианы как элемента, находящегося в середине отсортированного набора. Последовательный поиск с перебором всех элементов до нахождения нужного. Методы сужения области: бинарный поиск в отсортированном массиве с делением интервала пополам. Сортировка в Python через встроенную функцию sorted и метод sort списков с параметрами key и reverse.	ЛК, ЛР
Раздел 5	Алгоритмы на графах	5.1	Графы и их анализ. Представление графов. Обход графа в глубину и ширину	Графы как совокупность вершин и рёбер с классификацией на ориентированные и неориентированные. Способы представления графов: матрица смежности, список смежности, список рёбер. Обход графа в глубину с рекурсивным посещением всех достижимых вершин вдоль каждого пути. Обход графа в ширину с поэтапным расширением фронта посещения от начальной вершины.	ЛК, ЛР
		5.2	Восстановление кратчайшего пути. Задача о перемещении шахматного коня	Восстановление кратчайшего пути через сохранение предков для каждой вершины при обходе графа. Задача о перемещении шахматного коня на шахматной доске с поиском минимального количества ходов. Представление доски как графа, где вершины — клетки, а рёбра — возможные ходы коня.	ЛК, ЛР
		5.3	Алгоритм Дейкстры. Очередь и стек. Очередь и стек в Python	Алгоритм Дейкстры для нахождения кратчайших путей от одной вершины до всех остальных во взвешенном графе с неотрицательными весами. Очередь как структура данных с принципом первый вошёл — первый вышел. Стек как структура данных с принципом последним вошёл — первым вышел. Реализация очереди через collections.deque и стека через обычный список в Python.	ЛК, ЛР
Раздел 6	Динамическое программирование	6.1	Принцип оптимальности Беллмана. Понятие восходящего и нисходящего решения.	Принцип оптимальности Беллмана: оптимальное решение задачи может быть построено из оптимальных решений её подзадач. Нисходящее решение с рекурсивным разбиением задачи и запоминанием результатов для повторного использования. Восходящее решение с заполнением таблицы от меньших подзадач к большей.	ЛК, ЛР
		6.2	Задача о количестве маршрутов. Сходства и отличие динамического программирования и концепция «разделяй и властвуй»	Задача о количестве маршрутов на сетке с подсчётом числа способов добраться из одной точки в другую. Сходство динамического программирования и стратегии «разделяй и властвуй» в разбиении на подзадачи. Отличие: в динамическом программировании подзадачи перекрываются, а в стратегии «разделяй и властвуй» — независимы.	ЛК, ЛР
		6.3	Задача о банкомате. Динамическое программирование и игры	Задача о банкомате как задача выдачи суммы минимальным количеством монет или банкнот. Решение через динамическое программирование с построением оптимальных сумм от нуля до целевой. Применение динамического программирования в играх: анализ выигрышных и проигрышных позиций через рекуррентные соотношения.	ЛК, ЛР
Раздел 7	Параллельные алгоритмы	7.1	Предпосылки. Классификация вычислительных систем. CPU и GPU процессоры.	Предпосылки появления параллельных вычислений: достижение физических пределов тактовой частоты и рост объёмов обрабатываемых данных. Классификация вычислительных систем по Флинну: SISD, SIMD, MISD, MIMD. Центральные процессоры с небольшим числом мощных ядер и графические процессоры с тысячами маломощных ядер.	ЛК, ЛР
		7.2	Характеристики	Характеристики параллельных алгоритмов: ускорение, эффективность,	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			параллельных алгоритмов. Типы непоследовательного программирования в Python.	масштабируемость, затраты на синхронизацию. Типы непоследовательного программирования в Python: многопоточность для операций ввода-вывода, многопроцессность для вычислительных задач, асинхронное программирование для конкурентного выполнения.	
		7.3	Процессы и Потоки в Python. Асинхронные программы.	Процессы как изолированные экземпляры программы с собственным адресным пространством. Потоки как легковесные единицы выполнения внутри одного процесса с разделяемой памятью. Асинхронные программы с событийной моделью на основе корутин async и await.	ЛК, ЛР
Раздел 8	Оптимизация программ	8.1	Методы оптимизации и ускорения программ на Python. Профилирование программ на языке Python.	Методы оптимизации программ на Python: выбор эффективных алгоритмов, использование встроенных функций, избегание медленных конструкций. Профилирование как процесс измерения времени выполнения отдельных участков кода. Стандартный модуль cProfile для профилирования на уровне функций.	ЛК, ЛР
		8.2	Модуль line_profiler. Компиляция Python: Ahead-of-time и Just-in-time компиляция. Модуль Numba.	Модуль line_profiler для построчного профилирования кода и выявления самых медленных строк. Ahead-of-time компиляция с преобразованием кода до выполнения в машинный. Just-in-time компиляция с компиляцией во время выполнения программы. Модуль Numba для JIT-компиляции функций через декораторы.	ЛК, ЛР
		8.3	Cython как расширение языка Python. Особенности разработки программы на Cython	Cython как надмножество Python, позволяющее вызывать функции C и объявлять статические типы переменных. Особенности разработки на Cython: создание файлов pyx, объявление типов через ключевое слово cdef, компиляция в модуль расширения. Преимущества Cython: скорость, близкая к нативному C, при сохранении удобства Python.	ЛК, ЛР
Раздел 9	C\C++. Введение	9.1	C и C++ особенности языков, история и эволюция. Машинноориентированные языки программирования и принципы действия компьютера.	Язык C как процедурный язык низкого уровня, созданный для системного программирования. Язык C++ как объектно-ориентированное расширение C с сохранением обратной совместимости. История и эволюция: от ранних версий C до современного C++20 с добавлением лямбд, умных указателей и концептов.	ЛК, ЛР
		9.2	Трансляция кода. Виды трансляции. Отличия интерпретаторов и компиляторов.	Трансляция кода как преобразование исходного текста в машинные инструкции. Виды трансляции: компиляция с полным преобразованием до выполнения и интерпретация с построчным выполнением. Отличия компиляторов от интерпретаторов: компиляторы создают исполняемый файл, интерпретаторы выполняют код без отдельного этапа компиляции.	ЛК, ЛР
		9.3	Сопоставление программ на Python и C/C++. Область применения и языков C/C++.	Сопоставление Python и C/C++ по скорости выполнения, удобству разработки, контролю над памятью. Область применения C и C++: системное программирование, игровые движки, высокопроизводительные вычисления, встроенные системы.	ЛК, ЛР
Раздел 10	Основные элементы синтаксиса	10.1	Блочное устройство программ на языках C/C++, синтактические правила выделения блоков и их типы.	Блочное устройство программ с группировкой инструкций в блоки через фигурные скобки. Синтаксические правила выделения блоков: каждый блок образует новую область видимости. Типы блоков: тело функции, тело цикла, тело условной инструкции, анонимные блоки.	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
		10.2	Базовые инструкции: ветвление (или условная инструкция), циклы (while, do while и for), оператор безусловного перехода, оператор множественного выбора.	Ветвление через условную инструкцию if и конструкцию if-else с проверкой логических выражений. Циклы: while с предпроверкой условия, do while с постпроверкой, for со счётчиком. Оператор безусловного перехода goto для передачи управления на метку внутри функции. Оператор множественного выбора switch для выбора ветви по целочисленному значению.	ЛК, ЛР
		10.3	Синтаксические конструкции для работы с функциями: объявление, определение, вызов. Стек вызовов. Сравнение goto и return	Объявление функции прототип с указанием типа возврата, имени и параметров. Определение функции с телом в фигурных скобках. Вызов функции по имени с передачей аргументов. Стек вызовов как область памяти для хранения локальных переменных и адресов возврата. Сравнение оператора goto и return: return завершает функцию, goto передаёт управление внутри функции.	ЛК, ЛР
Раздел 11	Массивы и указатели	11.1	Указатели и адреса. Работа с указателями и адресами. Массив как структура данных: хранение в памяти, доступ к элементам.	Указатели как переменные, хранящие адреса памяти других переменных. Работа с указателями: оператор взятия адреса, оператор разыменования для доступа к значению по адресу. Массив как непрерывная область памяти для хранения однотипных элементов. Доступ к элементам массива через индексацию и через арифметику указателей.	ЛК, ЛР
		11.2	Создание статических массивов. Адресная арифметика	Создание статических массивов с фиксированным размером, известным на этапе компиляции. Адресная арифметика: сложение и вычитание указателей для перемещения по памяти. Связь между именем массива и указателем на первый элемент. Вычисление смещения элемента через индекс.	ЛК, ЛР
Раздел 12	Статическая и динамическая память.	12.1	Правила создание статических массивов, его инициализация и использование. Одномерные и многомерные статические массивы. Динамическая память (С стиль).	Правила создания статических массивов: размер должен быть константным выражением, память выделяется на стеке. Инициализация статических массивов при объявлении списком значений. Одномерные и многомерные статические массивы с размерами, указанными в квадратных скобках. Динамическая память в стиле С: функции malloc, calloc, realloc, free из библиотеки stdlib.	ЛК, ЛР
		12.2	Динамическая память (С++ стиль). Функции для работы с динамической памятью, операции выделения и освобождения памяти	Динамическая память в стиле С++: операции new для выделения памяти и delete для освобождения. Операция new с инициализацией значения в скобках. Операция delete для одиночных объектов и delete[] для массивов.	ЛК, ЛР
		12.3	Создание одномерных и многомерных динамических массивов	Создание одномерного динамического массива через операцию new с указанием размера. Создание многомерного динамического массива как массива указателей с последующим выделением памяти для каждой строки. Освобождение многомерного динамического массива в обратном порядке: сначала строки, затем массив указателей.	ЛК, ЛР
Раздел 13	Структурированные типы данных	13.1	Массивы, строки символов, структуры, объединение, перечислимый тип данных, битовые поля.	Массивы как наборы однотипных элементов с индексированным доступом. Строки символов как массивы типа char с нуль-терминатором в конце. Структуры для объединения разнотипных данных под одним именем. Объединения для хранения разных типов в одной области памяти. Перечислимый тип enum для именования	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			Синтаксические особенности объявления, инициализации и работы.	целочисленных констант. Битовые поля для экономии памяти при хранении флагов.	
		13.2	Особенности «упаковки» в памяти. Примеры использования. Динамические структуры данных: вектор, очередь (стек), список, как примеры организации работы с структурированными данными в динамическом режиме.	Особенности упаковки структур в памяти: выравнивание полей для эффективного доступа, возможные пустоты между полями. Динамические структуры данных: вектор с автоматическим расширением при заполнении. Очередь стек с дисциплиной обслуживания FIFO. Список с узлами, связанными указателями.	ЛК, ЛР
Раздел 14	Перехват ошибок	14.1	Синтаксис операции обработки исключений. Примеры использования.	Синтаксис обработки исключений в C++ с блоками try, catch и throw. Блок try для оборачивания кода, где возможно возникновение ошибки. Блок catch для перехвата и обработки исключения определённого типа. Операция throw для генерации исключения с передачей значения-исключения.	ЛК, ЛР
Раздел 15	Ввод-вывод данных	15.1	Понятие потока и буфера. Клавиатура, экран и файл как источник и приёмник данных. Организация потоков ввода и вывода данных в C++.	Поток как абстракция последовательности данных, передаваемых между программой и устройством. Буфер как временное хранилище данных при операциях ввода-вывода для повышения эффективности. Клавиатура, экран и файл как типичные источники и приёмники данных. Организация потоков ввода-вывода в C++ через классы iostream, ifstream, ofstream.	ЛК, ЛР
		15.2	Запись данных в поток и чтение данных из потока. Позиционирование данных в потоке. Режимы работы с файлами: чтение-запись, символьныйтекстовый формат и их комбинации	Запись данных в поток через операцию вставки. Чтение данных из потока через операцию извлечения. Позиционирование данных в потоке с функциями seekg для ввода и seekp для вывода. Режимы работы с файлами: чтение, запись, добавление, символьный текстовый формат, бинарный формат и их комбинации.	ЛК, ЛР
		15.3	Текстовые и бинарные файлы, и особенность в них хранения данных. Файлы прямого доступа	Текстовые файлы с хранением данных в виде символов, удобные для чтения человеком. Бинарные файлы с хранением данных в том же виде, что и в памяти, более компактные и быстрые для чтения. Файлы прямого доступа с возможностью перемещения к произвольной записи без чтения предыдущих.	ЛК, ЛР
Раздел 16	Объектно-ориентированное программирование в C++	16.1	Создание классов и объектов. Настройка модификаторов доступа: public, private и protected. Дружественные функции и классы.	Создание классов через ключевое слово class с набором полей и методов. Модификаторы доступа: public для открытого доступа, private для доступа только внутри класса, protected для доступа внутри класса и наследников. Дружественные функции и классы с ключевым словом friend для доступа к закрытым членам.	ЛК, ЛР
		16.2	Ключевое слово this. Организация операции наследования в языке C++. Виртуальные функции и	Ключевое слово this как указатель на текущий объект внутри методов класса. Наследование через двоеточие после имени класса с указанием базового класса. Виртуальные функции с ключевым словом virtual для динамического полиморфизма. Перегрузка функций с одинаковым именем, но разными параметрами. Перегрузка	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			перегрузка функций и операторов.	операторов через ключевое слово operator.	
Раздел 17	Использование библиотек	17.1	Обзор и примеры использования STL и BOOST	Стандартная библиотека шаблонов STL с контейнерами vector, list, map, set. Алгоритмы STL: сортировка, поиск, копирование, преобразование. Итераторы для обхода элементов контейнеров. Библиотека BOOST с расширенными возможностями: умные указатели, регулярные выражения, многопоточность, графы.	ЛК, ЛР
Раздел 18	Параллельные алгоритмы и системы	18.1	Классификация вычислительных систем. CPU и GPU процессоры. Характеристики параллельных алгоритмов	Классификация вычислительных систем по Флинну с четырьмя категориями. Центральные процессоры с архитектурой, оптимизированной для последовательных вычислений. Графические процессоры с массовым параллелизмом для обработки потоков данных. Характеристики параллельных алгоритмов: ускорение по Амдалу, эффективность использования ресурсов, масштабируемость при росте данных и процессоров.	ЛК, ЛР
		18.2	Типы непоследовательного программирования. Стандарты параллельных вычислений: взаимодействие между узлами суперкомпьютера, взаимодействие между ядрами одного CPU внутри одного узла, ускорители внутри одного узла	Типы непоследовательного программирования: параллельное, конкурентное и распределённое. Стандарт MPI для взаимодействия между узлами суперкомпьютера через передачу сообщений. Стандарт OpenMP для взаимодействия между ядрами одного процессора через разделяемую память. Технологии для ускорителей внутри одного узла: CUDA и OpenACC.	ЛК, ЛР
Раздел 19	Алгоритмы во внешней памяти	19.1	Организация вычислений с учётом иерархической структуры памяти. Буферизация при чтении и записи.	Иерархическая структура памяти: регистры, кэш первого и второго уровня, оперативная память, внешние накопители. Организация вычислений с минимизацией обращений к медленным уровням памяти. Буферизация при чтении и записи данных блоками для снижения числа операций ввода-вывода.	ЛК, ЛР
		19.2	Сложные и динамические структуры данных. Алгоритмы на графах во внешней памяти (BFS, DFS, поиск связных компонент, MST)	Сложные и динамические структуры данных, адаптированные для хранения на диске. Алгоритмы на графах во внешней памяти: обход в ширину BFS и обход в глубину DFS с оптимизацией числа обращений к диску. Поиск связных компонент графа без полной загрузки в память. Построение минимального остовного дерева MST для графов, не помещающихся в оперативную память.	ЛК, ЛР
Раздел 20	Технология OpenMP	20.1	Параллельные вычисления с использованием стандарта OpenMP.	Параллельные вычисления с использованием стандарта OpenMP для многопоточного программирования на языках C, C++ и Fortran. Основные сведения о модели параллелизма с разделяемой памятью. Нити как единицы выполнения и процессы как контейнеры для нитей.	ЛК, ЛР
		20.2	Основные сведения. Нити и процессы. Параллельные и последовательные области	Параллельные области с директивой parallel для создания пула нитей. Последовательные области, выполняемые только главной нитью. Директива single для выполнения блока кода одной нитью. Директива master для выполнения блока кода главной нитью.	ЛК, ЛР
		20.3	Параллельные циклы и параллельные области.	Параллельные циклы с директивой for для распределения итераций между нитями. Директива parallel for как объединение создания параллельной области и	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			Автоматическое распараллеливания циклов.	параллельного цикла. Автоматическое распараллеливание циклов компилятором с анализом зависимостей по данным. Клаузула schedule для управления способом распределения итераций.	
Раздел 21	Технология MPI	21.1	Параллельные вычисления с использованием стандарта MPI. Основные сведения. Основные процедуры MPI.	Параллельные вычисления с использованием стандарта MPI для распределённых систем с передачей сообщений. Основные сведения о модели взаимодействия процессов без разделяемой памяти. Основные процедуры MPI: инициализация MPI_Init, завершение MPI_Finalize, определение ранга процесса MPI_Comm_rank, определение числа процессов MPI_Comm_size.	ЛК, ЛР
		21.2	Типы данных MPI. Способы передачи сообщений. Прием и передача сообщений процессами	Типы данных MPI для описания содержимого сообщений: базовые типы MPI_INT, MPI_FLOAT, MPI_DOUBLE и производные. Способы передачи сообщений: блокирующие операции MPI_Send и MPI_Recv с ожиданием завершения. Неблокирующие операции MPI_Isend и MPI_Irecv с проверкой готовности через MPI_Wait или MPI_Test. Коллективные операции: MPI_Bcast для рассылки, MPI_Reduce для свертки, MPI_Barrier для синхронизации.	ЛК, ЛР
Раздел 22	Технология OpenACC	22.1	Параллельные вычисления с использованием стандарта OpenACC	Параллельные вычисления с использованием стандарта OpenACC для программирования ускорителей GPU. Директивы компилятора для переноса вычислений на устройство. Поддержка языков C, C++ и Fortran.	ЛК, ЛР
		22.2	Обзор производительности GPU в различных приложениях. Сравнение вычислительных ускорителей. Основные принципы достижения высокой производительности	Обзор производительности GPU в приложениях: глубокое обучение, научные расчёты, обработка изображений. Сравнение вычислительных ускорителей от разных производителей по пиковой производительности и энергоэффективности. Основные принципы достижения высокой производительности: объединение доступа к памяти, избегание рассинхронизации потоков.	ЛК, ЛР
		22.3	Преимущества OpenACC. Модель исполнения: gangs, workers, vectors. Директивы paralallel, kernels, loop	Преимущества OpenACC перед низкоуровневыми подходами: меньше кода, переносимость между разными ускорителями. Модель исполнения с тремя уровнями параллелизма: gangs группы блоков, workers подгруппы, vectors векторные нити. Директива parallel для параллельной области. Директива kernels для автоматического распараллеливания участка кода. Директива loop с указанием уровней параллелизма gang, worker, vector.	ЛК, ЛР
		22.4	Атрибуты данных. Регионы данных: data, enter data, exit data.Дополнительные конструкции управления данными: cache, update, declare	Атрибуты данных для управления перемещением данных между хостом и устройством. Регионы данных: директива data для длительного пребывания данных на устройстве. Директивы enter data и exit data для ручного управления присутствием данных. Дополнительные конструкции: cache для кэширования данных в разделяемой памяти, update для обновления данных, declare для глобальных переменных.	ЛК, ЛР
		22.5	Асинхронное исполнение - async и wait. Атомарные операции. Глобальные переменные. OpenACC в C++	Асинхронное исполнение с директивами async и wait для наложения вычислений и передачи данных. Атомарные операции для защиты от гонок данных при параллельном доступе. Глобальные переменные с директивами declare для использования на устройстве. Применение OpenACC в C++ с поддержкой классов, шаблонов и стандартной библиотеки.	ЛК, ЛР
Раздел 23	Программно-аппаратная	23.1	Архитектура GPU.	Архитектура GPU с множеством потоковых мультипроцессоров, каждый из которых	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
	архитектура CUDA		Иерархия памяти GPU. Программная модель CUDA.	содержит много ядер CUDA. Иерархия памяти GPU: глобальная память для всех потоков, разделяемая память внутри блока, локальная память для отдельной нити, регистры. Программная модель CUDA с сеткой блоков, блоками нитей и отдельными нитями. Функции kernel с квалификатором global для выполнения на GPU.	
		23.2	Использование библиотек C++ для программирования на OpenCL и CUDA	Использование библиотек C++ для программирования на OpenCL: библиотеки обёртки для упрощения работы с платформами, устройствами и очередями команд. Использование библиотек для CUDA: cuBLAS для линейной алгебры, cuFFT для быстрого преобразования Фурье, Thrust для алгоритмов с контейнерами. Сравнение OpenCL как кросс-платформенного стандарта и CUDA как проприетарной технологии NVIDIA.	ЛК, ЛР
Раздел 24	Введение в распределенные объектные технологии	24.1	Понятие распределенной системы обработки информации. Виды и свойства распределенных систем.	Понятие распределённой системы как совокупности независимых компьютеров, взаимодействующих через сеть. Виды распределённых систем: клиент-серверные, одноранговые, гетерогенные. Свойства распределённых систем: масштабируемость, отказоустойчивость, прозрачность, открытость.	ЛК, ЛР
		24.2	Архитектура программного обеспечения информационных систем. Управление взаимодействием разнородных приложений	Архитектура программного обеспечения информационных систем с разделением на уровни представления, бизнес-логики и доступа к данным. Управление взаимодействием разнородных приложений через общие протоколы и форматы данных. Проблемы разнородности: операционные системы, языки программирования, структуры данных.	ЛК, ЛР
		24.3	Основные механизмы распределенных объектных технологий.	Основные механизмы распределённых объектов: удалённый вызов процедур, удалённая активация объектов, маршалинг и демаршалинг параметров. Связывание клиента с удалённым объектом через реестр имён. Управление жизненным циклом распределённых объектов.	ЛК, ЛР
Раздел 25	Основные модели распределенных объектных технологий	25.1	Виды распределенных приложений. Облачные технологии. Определение облачных вычислений. Многослойная архитектура облачных приложений. Компоненты облачных приложений.	Виды распределённых приложений: веб-приложения, корпоративные информационные системы, системы реального времени. Облачные технологии с предоставлением вычислительных ресурсов как услуги. Определение облачных вычислений через пять ключевых характеристик: самообслуживание, широкий сетевой доступ, пул ресурсов, быстрое масштабирование, измеряемость. Многослойная архитектура облачных приложений: инфраструктура как услуга, платформа как услуга, программное обеспечение как услуга.	ЛК, ЛР
		25.2	Достоинства и недостатки облачных вычислений. Наиболее распространенные облачные платформы. GRID-технологии.	Достоинства облачных вычислений: снижение капитальных затрат, эластичность, оплата по факту использования. Недостатки: задержки сети, вопросы безопасности, зависимость от провайдера. Наиболее распространённые облачные платформы: Amazon Web Services, Microsoft Azure, Google Cloud Platform. GRID-технологии для объединения распределённых вычислительных ресурсов в единую систему.	ЛК, ЛР
		25.3	Архитектура GRID. Стандарты GRID. Параметрические модели производительности GRID. Сравнение GRID и	Архитектура GRID с уровнями: фабричный уровень ресурсов, уровень подключения, уровень коллективных сервисов. Стандарты GRID от организации Open Grid Forum. Параметрические модели производительности GRID для оценки пропускной способности. Сравнение GRID и облачных вычислений: GRID ориентирована на пакетную обработку задач, облака — на сервисы по запросу.	ЛК, ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
			Облачных вычислений.		
Раздел 26	Проблемы интеграции приложений	26.1	Проблемы интеграции приложений. Комплексная интеграция приложений. Брокеры сообщений. Модель взаимодействия "публикация/подписка"	Проблемы интеграции приложений: разнородность платформ, несовместимость форматов данных, дублирование информации. Комплексная интеграция приложений как систематический подход к объединению информационных систем предприятия. Брокеры сообщений как промежуточное звено для асинхронной передачи данных между приложениями. Модель взаимодействия публикация-подписка с разделением отправителей и получателей через темы сообщений.	ЛК, ЛР
		26.2	Системы управления рабочим потоком. Серверы приложений.	Системы управления рабочим потоком для координации выполнения бизнес-процессов, состоящих из множества шагов. Маршрутизация задач между участниками, контроль сроков выполнения, журналирование событий. Серверы приложений как платформа для выполнения корпоративных приложений с поддержкой транзакций, безопасности, кластеризации и подключения к базам данных.	ЛК, ЛР

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

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Таблица 6.1. Материально-техническое обеспечение дисциплины

Тип аудитории	Оснащение аудитории	Специализированное учебное/лабораторное оборудование, ПО и материалы для освоения дисциплины (при необходимости)
Лекционная	Аудитория для проведения занятий лекционного типа, оснащенная комплектом специализированной мебели; доской (экраном) и техническими средствами мультимедиа презентаций.	
Компьютерный класс	Компьютерный класс для проведения занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная персональными компьютерами (в количестве ____ шт.), доской (экраном) и техническими средствами мультимедиа презентаций.	
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.	

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

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

1. Python 3. Самое необходимое. Прохоренок Н., Дронов В., БХВ-Петербург, 2019 – 610 с.;
2. Python. Экспресс-курс. Седер Н., СПб.: Питер, 2019 – 480 с.;
3. Алгоритмы. Справочник с примерами на C, C++, Java и Python. Хайнеман Дж., Поллис Г., Селков С., СПб.: ООО "Альфа-книга", 2017 – 432 с.;
4. Программирование на языке высокого уровня. C/C++. Хабибуллин И.Ш., СПб.: БХВПетербург, 2006 – 512 с
5. C-C++. Справочник программиста. Г. Шилдт, Вильямс, 2003 - 429 с.;
6. Программирование на C++ в Visual Studio 2010 Express. Прохоренок Н.А., 2010 – 71 с
7. Язык программирования C. Брайан У. Керниган, Д.М. Ритчи, Вильямс, 2015 – 288 с
8. Язык программирования C++. Страуструп Б., Мартынов Н.Н., Москва: Бином, 2011. - 1135 с.
9. Параллельные вычисления. Воеводин В. В., Воеводин Вл. В., СПб.: БХВ-Петербург, 2002
10. Параллельное и распределенное программирование с использованием C++. Хьюз К., Хьюз Т., М.: Издательский дом «Вильямс», 2004;
11. Параллельное программирование с использованием OpenMP. Левин М.П. М.: Бином. Лаборатория знаний, 2008
12. Parallel programming with OpenACC. Farber R., Newnes, 2016 – 316 с.;
13. Технология CUDA в примерах: введение в программирование графических процессоров. Сандере Дж., Кэндрот Э. М.: ДМК Пресс, 2011 - 232 с
14. Распределенные системы. Принципы и парадигмы. Таненбаум Э., ван Стеен М. СПб: Питер, 2003. - 877 с
15. Разработка распределенных приложений на платформе .NET Framework. М. Сара, Р. Билл, Х. Шеннон, Б. Марк. СПб: Питер, 2008. - 608 с

Дополнительная литература:

1. Автоматизация раненных задач с помощью Python: практическое руководстве для начинающих. Свейгарт Эл., М.:"ИД Вильямс", 2017 – 592 с
2. Численные методы: Вычислительный практикум. Вабищевич П.Н., М.: «ЛИБРОКОМ», 2010 – 320 с.;

3. Язык программирования С. Лекции и упражнения. С. Прата, М.: Издательский дом “Вильямс”, 2013 – 960 с.;

4. C++. Священные знания. Дьюхерст С., СПб.: Символ Плюс, 2012 – 240 с

5. Алгоритмы. Справочник с примерами на C, C++, Java и Python. Хайнеман Дж., Поллис Г., Селков С., СПб.: ООО "Альфа-книга", 2017 – 432 с.

6. Алгоритмы построение, анализ и реализация на языке программирования Си. Ворожцов А.В., Винокуров Н.А., Москва: МФТИ, 2007 – 452 с.

7. Программирование и информатика. Антонюк В.А., Иванов А.П., Москва: Физический фак. МГУ им. М. В. Ломоносова, 2015 – 64 с.

8. Последовательные и параллельные алгоритмы. Миллер Р., Боксер Л. М.: Бином. Лаборатория знаний, 2006

9. Введение в параллельные методы решения задач. Якобовский М. В. М.: Издательство Московского университета, 2013 – 328 с.;

10. Отладка приложений для Microsoft .NET и Microsoft Windows. Р. Джон. М.: Microsoft Press. Русская Редакция. 2008.

11. XML. Новые перспективы WWW. Бумфрей Ф, Диренцо О, Дакетт Й. Издательство: "ДМК Пресс", 2006. - 688 с

12. Чертовской В. Д. Базы данных: теория и практика: учебник для бакалавров. Советов Б. Я., Цехановский В. В. М.: ЮРАЙТ, 2011. - 459 с

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 http://lib.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 «Троицкий мост»

2. Базы данных и поисковые системы

- электронный фонд правовой и нормативно-технической документации http://docs.cntd.ru/
- поисковая система Яндекс https://www.yandex.ru/
- поисковая система Google https://www.google.ru/
- реферативная база данных SCOPUS http://www.elsevierscience.ru/products/scopus/

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

1. Курс лекций по дисциплине «Programming Technology».

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

РАЗРАБОТЧИКИ

Доцент	Иванюхин А.В.
Должность	Фамилия И.О

РУКОВОДИТЕЛЬ БУП

Заведующий кафедрой	Разумный Ю.Н.
Должность	Фамилия И.О

РУКОВОДИТЕЛЬ ОП ВО

Профессор	Разумный Ю.Н.
Должность	Фамилия И.О