

Автономная некоммерческая образовательная организация высшего образования
«Контур Университет»



Ректор

С.Е. Стрельникова

2026 г.

Учебный план 09.03.03_ИИС_2026.plx
09.03.03 Прикладная информатика

Квалификация	бакалавр
--------------	----------

Форма обучения **очная**

Общая трудоемкость 0,3 ЕТ

Часов по учебному плану 328

в том числе:

аудиторные занятия 128

самостоятельная работа 192

часов на контроль	8
-------------------	---

Виды контроля в семестрах:

зачеты 2, 3, 4, 5

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		3 (2.1)		4 (2.2)		5 (3.1)		Итого	
Неделя	17 1/6		17 2/6		17 1/6		17 3/6			
Вид занятий	уп	рп	уп	рп	уп	рп	уп	рп	уп	рп
Практические	32	32	32	32	32	32	32	32	128	128
Итого ауд.	32	32	32	32	32	32	32	32	128	128
Контактная работа	32	32	32	32	32	32	32	32	128	128
Сам. работа	26	26	26	26	74	74	66	66	192	192
Часы на контроль	2	2	2	2	2	2	2	2	8	8
Итого	60	60	60	60	108	108	100	100	328	328

Рабочая программа дисциплины

Спортивное программирование

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки
09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

Одобрена Ученым советом вуза

Протокол от 15.09.2026 № 05.

Одобрена учебно-методической комиссией

Протокол от 10.09.2026 г. № 01

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Цель освоения дисциплины:
1.2	Формирование у обучающихся системы знаний и практических навыков в области спортивного программирования, развитие алгоритмического мышления, навыков решения сложных вычислительных задач в условиях ограниченного времени, а также способности применять современные инструменты и методы программирования для участия в соревновательной деятельности и профессионального развития.
1.3	Задачи изучения дисциплины:
1.4	Ознакомить с историей развития спортивного программирования, его структурой и статусом официального вида спорта в Российской Федерации.
1.5	Сформировать знания о классификации спортивных дисциплин и правилах соревнований.
1.6	Обучить основным алгоритмическим парадигмам, структурам данных и методам их применения для решения соревновательных задач.
1.7	Сформировать навыки работы в соревновательных системах (отладка, оптимизация кода по времени и памяти).
1.8	Развить умение анализировать условия задач, выбирать оптимальный алгоритм и структуру данных, оценивать вычислительную сложность.
1.9	Сформировать устойчивую мотивацию к участию в соревнованиях и развитию навыков командной работы.
1.10	Воспитать стрессоустойчивость, концентрацию внимания, настойчивость и самоконтроль.
1.11	Сформировать навыки самостоятельной подготовки к соревнованиям и работы с онлайн-тренажёрами.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.ДВ.01
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Знания, умения и навыки, приобретённые при освоении основной образовательной программы среднего общего образования.
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Интеллектуальные информационные системы
2.2.2	Производственная практика (технологическая (проектно-технологическая))

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ	
УК-7: Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности	
УК-7.1: Знает виды физических упражнений; научно-практические основы физической культуры и здорового образа и стиля жизни	
УК-7.2: Умеет применять на практике разнообразные средства физической культуры, спорта и туризма для сохранения и укрепления здоровья, психофизической подготовки и самоподготовки к будущей жизни и профессиональной деятельности; использовать творчески средства и методы физического воспитания для профессионально-личностного развития, физического самосовершенствования, формирования здорового образа и стиля жизни	
УК-7.3: Владеет средствами и методами укрепления индивидуального здоровья, физического самосовершенствования	

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1.			
1.1	Введение в спортивное программирование /Пр/	2	2	
1.2	Введение в спортивное программирование /Ср/	2	1	
1.3	Организация и правила соревнований /Пр/	2	6	
1.4	Организация и правила соревнований /Ср/	2	5	
1.5	Языки программирования в спортивном программировании /Пр/	2	12	
1.6	Языки программирования в спортивном программировании /Ср/	2	10	
1.7	Базовые алгоритмические конструкции /Пр/	2	12	
1.8	Базовые алгоритмические конструкции /Ср/	2	10	
1.9	/Зачёт/	2	2	
	Раздел 2.			
2.1	Оценка сложности алгоритмов /Пр/	3	6	
2.2	Оценка сложности алгоритмов /Ср/	3	5	
2.3	Основные структуры данных /Пр/	3	6	
2.4	Основные структуры данных /Ср/	3	5	
2.5	Алгоритмы сортировки и поиска /Пр/	3	6	
2.6	Алгоритмы сортировки и поиска /Ср/	3	5	
2.7	Математические алгоритмы /Пр/	3	6	
2.8	Математические алгоритмы /Ср/	3	5	
2.9	Жадные алгоритмы /Пр/	3	8	
2.10	Жадные алгоритмы /Ср/	3	6	
2.11	/Зачёт/	3	2	
	Раздел 3.			
3.1	Динамическое программирование (базовый уровень) /Пр/	4	16	
3.2	Динамическое программирование (базовый уровень) /Ср/	4	36	
3.3	Динамическое программирование (продвинутый уровень) /Пр/	4	16	
3.4	Динамическое программирование (продвинутый уровень) /Ср/	4	38	
3.5	/Зачёт/	4	2	
	Раздел 4.			
4.1	Теория графов и алгоритмы на графах /Пр/	5	8	
4.2	Теория графов и алгоритмы на графах /Ср/	5	16	
4.3	Работа со строками /Пр/	5	8	

4.4	Работа со строками /Ср/	5	16	
4.5	Продуктовое программирование (хакатоны) /Пр/	5	8	
4.6	Продуктовое программирование (хакатоны) /Ср/	5	18	
4.7	Здоровьесбережение в профессиональной деятельности /Пр/	5	8	
4.8	Здоровьесбережение в профессиональной деятельности /Ср/	5	16	
4.9	/Зачёт/	5	2	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации

Тема 1. Введение в спортивное программирование

Практическое задание: Подготовить краткий доклад (5–7 минут) об истории возникновения и развития спортивного программирования, его статусе как официального вида спорта в Российской Федерации, основных дисциплинах и известных российских спортсменах-программистах. Ознакомиться с правилами проведения соревнований, структурой турниров и системами тестирования. Представить презентацию с ключевыми фактами и примерами соревнований.

Тема 2. Организация и правила соревнований

Практическое задание: Зарегистрироваться на одной из платформ для спортивного программирования (Codeforces, AtCoder, Яндекс.Контест). Изучить интерфейс системы тестирования, правила отправки решений, форматы ввода-вывода данных. Решить 2–3 простые задачи из архива платформы, разобраться с форматом отчёта об ошибках (WA, TLE, RE, CE). Зафиксировать результаты и проанализировать ошибки.

Тема 3. Языки программирования в спортивном программировании

Практическое задание: Написать решение одной и той же задачи на трёх языках программирования (C++, Python, Java). Сравнить время выполнения и объём кода. Составить таблицу сравнения языков по критериям: скорость компиляции/интерпретации, удобство написания, доступность стандартных библиотек, наличие встроенных структур данных. Сделать вывод о выборе языка для соревнований.

Тема 4. Базовые алгоритмические конструкции

Практическое задание: Решить 5–7 задач на условные операторы, циклы (while, for), работу с массивами и списками из архива задач (например, Codeforces, ACMР). Для каждой задачи записать алгоритм решения в виде блок-схемы или псевдокода, затем реализовать на выбранном языке программирования. Протестировать решения на наборе тестовых данных. Зафиксировать время выполнения и использованную память.

Тема 5. Оценка сложности алгоритмов

Практическое задание: Для 5 алгоритмов (линейный поиск, бинарный поиск, пузырьковая сортировка, быстрая сортировка, сортировка слиянием) рассчитать теоретическую временную сложность в лучшем, среднем и худшем случаях. Запустить каждый алгоритм на массивах размером 10, 100, 1000, 10 000 элементов. Зафиксировать время выполнения (в микросекундах), построить график зависимости времени от размера входных данных. Сравнить теоретические и экспериментальные данные, объяснить расхождения. Проанализировать, для каких задач каждый алгоритм предпочтителен.

Тема 6. Основные структуры данных

Практическое задание: Реализовать базовые операции для каждой структуры данных: стек, очередь, дек, ассоциативный массив (хэш-таблица), множество. Для стека и очереди использовать как массив, так и односвязный список (если язык позволяет). Сравнить скорость выполнения операций (вставка, удаление, поиск) для разных реализаций. Решить 2 задачи на использование стека (например, проверка сбалансированности скобок) и 2 задачи на использование очереди (например, задачи на BFS). Зафиксировать результаты. Для C++ использовать STL, для Python — встроенные типы list, deque, dict, set. Оценить временную сложность операций в каждом случае.

Тема 7. Алгоритмы сортировки и поиска

Практическое задание: Реализовать алгоритмы сортировки: пузырьковую, быструю, слиянием и сортировку подсчётом. Сравнить их производительность на массивах разных типов (случайный, почти отсортированный, обратный порядок) размером 10, 100, 1000, 10 000 элементов. Построить графики времени выполнения и числа сравнений для каждого алгоритма. Реализовать бинарный поиск и линейный поиск, сравнить их эффективность. Найти задачу на бинарный поиск в архиве и решить её (поиск элемента в отсортированном массиве, поиск первого вхождения). Сделать вывод о применимости каждого алгоритма.

Тема 8. Математические алгоритмы

Практическое задание: Решить 5–7 задач на использование алгоритма Евклида (НОД, НОК), бинарное возведение в степень, проверку простоты числа (тест Миллера–Рабина), решение диофантовых уравнений. Реализовать функции для НОД, НОК, быстрого возведения в степень. Проверить их работу на наборе тестов (как минимум 10 примеров для каждой).

функции). Сравнить время выполнения с встроенными функциями (если доступны). Проанализировать, как изменение входных данных (размер чисел) влияет на скорость работы. Решить олимпиадную задачу на одну из тем.

Тема 9. Жадные алгоритмы

Практическое задание: Решить 3–4 задачи на жадные алгоритмы из архива задач: задача о выборе заявок, задача о минимальном покрытии отрезков, задача о рюкзаке (дробная версия), задача о размене монет. Для каждой задачи: 1) сформулировать жадную стратегию, 2) доказать её корректность (кратко), 3) реализовать решение на выбранном языке, 4) протестировать на наборе тестовых данных, 5) сравнить с результатом полного перебора для небольших входных данных (чтобы убедиться в корректности). Зафиксировать время выполнения для каждого решения.

Тема 10. Динамическое программирование (базовый уровень)

Практическое задание: Решить 3–4 задачи на одномерное и двумерное динамическое программирование: поиск наибольшей возрастающей подпоследовательности (LIS), задача о рюкзаке (0/1), задача о наибольшей общей подпоследовательности (LCS), задача о минимальном количестве монет для суммы. Для каждой задачи: 1) определить состояние и рекуррентное соотношение, 2) реализовать решение с использованием итеративного DP, 3) оптимизировать по памяти (если возможно), 4) протестировать на наборе тестов. В отчёт включить обоснование выбора состояния и рекуррентного соотношения.

Тема 11. Динамическое программирование (продвинутый уровень)

Практическое задание: Решить 2–3 задачи на динамическое программирование на подотрезках: задача о правильной скобочной последовательности, задача о палиндромах, задача об оптимальной триангуляции многоугольника. Решить 2 задачи на динамическое программирование на подмножествах: задача о коммивояжёре (TSP), задача о назначениях. Для каждой задачи реализовать решение с использованием DP по маскам. Сравнить время выполнения для разных размеров входных данных ($n = 10, 15, 20$). Проанализировать экспоненциальный рост времени выполнения в зависимости от n .

Тема 12. Теория графов и алгоритмы на графах

Практическое задание: Реализовать представление графов (список смежности, матрица смежности). Решить 2 задачи на DFS (поиск компонент связности, обход графа), 2 задачи на BFS (поиск кратчайшего пути в невзвешенном графе). Реализовать алгоритм Дейкстры для поиска кратчайшего пути во взвешенном графе и решить 2 задачи на его применение. Реализовать систему непересекающихся множеств (DSU) и алгоритм Краскала для поиска минимального остовного дерева. Протестировать алгоритмы на нескольких графах (не менее 5). Сравнить производительность алгоритмов на графах с разным количеством вершин и рёбер.

Тема 13. Работа со строками

Практическое задание: Реализовать алгоритмы поиска подстроки: префикс-функция, z-функция, алгоритм Кнута-Морриса-Праката (КМП). Для каждого алгоритма рассчитать время выполнения на строках разной длины (100, 1000, 10 000, 100 000 символов). Сравнить производительность алгоритмов. Решить задачу на поиск всех вхождений подстроки в строку (из архива) с использованием каждого из трёх алгоритмов. Сравнить время выполнения. Реализовать хеширование строк (полиномиальный хеш) и решить задачу на поиск подстроки с использованием хешей.

Тема 14. Продуктовое программирование (хакатоны)

Практическое задание: В составе команды из 2–4 человек разработать прототип приложения, веб-сервиса или телеграм-бота за ограниченное время (4–6 часов) по заданной теме. Тематика задания определяется преподавателем (например, «Сервис для учёта задач», «Бот для планирования времени», «Аналитический дашборд»). Распределить роли в команде (тимлид, разработчик интерфейса, разработчик бэкенда, тестировщик). Подготовить презентацию результата (5–7 минут) с демонстрацией работающего прототипа. Провести ретроспективу: что удалось, что не удалось, какие навыки нужно развивать. Защитить проект перед преподавателем или группой.

Тема 15. Здоровьесбережение в профессиональной деятельности

Практическое задание: Провести самооценку психоэмоционального состояния до и после 2-часовой тренировки по решению олимпиадных задач (с использованием опросников, например, шкала САН или опросник самочувствия). Оценить уровень утомления, концентрации внимания, стресса. Разработать индивидуальный план профилактики профессиональных заболеваний для программиста на основе рекомендаций (включить рекомендации по организации рабочего места, гимнастике для глаз и спины, режиму труда и отдыха, дыхательным упражнениям). Составить чек-лист здоровьесберегающих привычек на неделю. Зафиксировать результаты. Подготовить краткий доклад (5–7 минут) по теме «Здоровьесберегающие технологии в профессиональной деятельности программиста» с презентацией рекомендаций. Провести анализ и сделать выводы.

Примерный перечень вопросов к зачету:

Теоретические вопросы

1. История возникновения и развития спортивного программирования как вида интеллектуального спорта. Основные этапы становления в России и мире.
2. Статус спортивного программирования как официального вида спорта в Российской Федерации. Дисциплины и разряды.
3. Структура соревнований по спортивному программированию. Основные форматы: командные (ICPC), индивидуальные (Codeforces, AtCoder), оптимизационные (хакатоны).
4. Правила проведения соревнований. Системы тестирования (Яндекс.Контест, Codeforces, AtCoder). Принципы

оценки решений.

5. Языки программирования в спортивном программировании. Сравнительная характеристика C++, Python, Java. Особенности использования стандартных библиотек.
 6. Оценка сложности алгоритмов. Временная и пространственная сложность. O-нотация и её применение для анализа эффективности решений .
 7. Основные структуры данных в спортивном программировании: стек, очередь, дек, ассоциативные массивы (хеш- таблицы), множества. Стандартные контейнеры STL .
 8. Алгоритмы сортировки и поиска. Сравнительная характеристика методов. Сортировка слиянием и её применения .
 9. Математические алгоритмы: алгоритм Евклида (НОД, НОК), расширенный алгоритм Евклида, бинарное возведение в степень. Решение диофантовых уравнений .
 10. Жадные алгоритмы. Принцип оптимальности. Решение задач о выборе заявок, минимальном покрытии, дробном рюкзаке.
 11. Динамическое программирование. Принципы и основные задачи: поиск наибольшей возрастающей подпоследовательности (LIS), наибольшей общей подпоследовательности (LCS), задача о рюкзаке .
 12. Динамическое программирование на подотрезках и подмножествах. Задача об оптимальной траектории.
 13. Теория графов. Представление графов (список смежности, матрица смежности). Алгоритмы DFS и BFS .
 14. Алгоритмы на графах: алгоритм Дейкстры, алгоритм Краскала, система непересекающихся множеств (DSU) .
 15. Работа со строками. Алгоритмы поиска подстроки: префикс-функция, z-функция, алгоритм Кнута-Морриса-Практа (КМП). Хеширование строк .
 16. Продуктовое программирование (хакатоны). Особенности разработки приложений и сервисов в условиях ограниченного времени. Командная работа.
 17. Стратегии решения задач в условиях соревнований. Анализ вердиктов тестирующей системы (WA, TLE, RE, CE) .
 18. Здоровьесбережение в профессиональной деятельности программиста. Влияние интеллектуальных нагрузок на когнитивное здоровье. Методы профилактики профессиональных заболеваний.
- Практические задания (демонстрация навыков)
19. Решить задачу на базовые алгоритмические конструкции (условные операторы, циклы, работа с массивами) .
 20. Решить задачу на оценку сложности алгоритма: рассчитать время выполнения для заданного набора входных данных.
 21. Реализовать алгоритм сортировки (быстрая сортировка или сортировка слиянием) и продемонстрировать его работу на примере .
 22. Решить задачу на использование стека или очереди (например, проверка сбалансированности скобок).
 23. Решить задачу на алгоритм Евклида (НОД или расширенный алгоритм Евклида) .
 24. Решить задачу на жадный алгоритм (например, задача о выборе заявок или о размене монет).
 25. Решить задачу на динамическое программирование (задача о рюкзаке, LIS или LCS) .
 26. Решить задачу на графы: поиск кратчайшего пути (алгоритм Дейкстры) или поиск в глубину/ширину .
 27. Решить задачу на работу со строками (поиск подстроки с использованием префикс-функции или КМП) .
 28. Решить задачу на основы теории чисел (проверка простоты, бинарное возведение в степень).
 29. Продемонстрировать навыки работы в соревновательной системе: отправить решение, проанализировать вердикт, исправить ошибки.
 30. Проанализировать решение олимпиадной задачи: оценить временную и пространственную сложность, предложить оптимизацию .

5.2. Темы письменных работ

не предусмотрены

5.3. Фонд оценочных средств

Тестовые задания:

РАЗДЕЛ 1. ВОПРОСЫ НА СООТВЕТСТВИЕ (№ 1–5)

Инструкция: к каждому элементу из левого столба подберите ОДИН соответствующий элемент из правого столбца. Ответ запишите в виде сочетания цифр и букв (например: 1-А, 2-Б, 3-В, 4-Г). Каждая буква используется только один раз.

УК-7.1. Вопрос № 1. Установите соответствие между вердиктом системы тестирования и его описанием:

Вердикт Описание

1. Accepted (OK) А. Программа выполняется дольше допустимого времени
2. Time Limit Exceeded (TLE) Б. Ошибка в процессе выполнения программы (деление на ноль, выход за границы массива)
3. Runtime Error (RE) В. Решение полностью правильное, все тесты пройдены
4. Wrong Answer (WA) Г. Программа дала неверный ответ на тестовых данных

Ответ: 1 __, 2 __, 3 __, 4 __

УК-7.1. Вопрос № 2. Установите соответствие между структурой данных и её характеристикой:

Структура данных Характеристика

1. Стек А. Принцип FIFO (первым пришёл — первым вышел)
2. Очередь Б. Принцип LIFO (последним пришёл — первым вышел)
3. Дек В. Хранит пары «ключ-значение» с доступом по ключу
4. Хеш-таблица Г. Элементы добавляются и удаляются с обоих концов

Ответ: 1 __, 2 __, 3 __, 4 __

УК-7.2. Вопрос № 3. Установите соответствие между алгоритмом и его временной сложностью:

Алгоритм Временная сложность

1. Бинарный поиск в отсортированном массиве А. $O(n^2)$
 2. Пузырьковая сортировка Б. $O(\log n)$
 3. Поиск в ширину (BFS) в графе В. $O(n + m)$
 4. Сортировка слиянием Г. $O(n \log n)$
- Ответ: 1 __, 2 __, 3 __, 4 __

УК-7.2. Вопрос № 4. Установите соответствие между дисциплиной спортивного программирования и её описанием:

Дисциплина	Описание
------------	----------

- | | |
|---|---|
| 1. Алгоритмическое программирование А. | Решение задач на поиск уязвимостей и защиту информации |
| 2. Продуктовое программирование (хакатоны) Б. | Решение алгоритмических задач на время с использованием языков программирования |
| 3. Программирование систем информационной безопасности В. | Разработка приложений и сервисов в команде за ограниченное время |
| 4. Программирование БАС Г. | Разработка программного обеспечения для беспилотных авиационных систем |

Ответ: 1 __, 2 __, 3 __, 4 __

УК-7.1. Вопрос № 5. Установите соответствие между методом профилактики профессиональных заболеваний и его описанием:

Метод профилактики	Описание
--------------------	----------

- | | |
|-----------------------------------|---|
| 1. Гимнастика для глаз А. | Упражнения для снятия напряжения с мышц спины и шеи |
| 2. Эргономика рабочего места Б. | Комплекс упражнений для снятия усталости с глазных мышц |
| 3. Производственная гимнастика В. | Организация рабочего пространства с учётом правильной посадки и освещения |
| 4. Дыхательные упражнения Г. | Снижение уровня тревожности и стабилизация психоэмоционального состояния |

Ответ: 1 __, 2 __, 3 __, 4 __

РАЗДЕЛ 2. ВОПРОСЫ С ОДНИМ ВЕРНЫМ ОТВЕТОМ (№ 6–15)

Инструкция: прочитайте текст, выберите правильный вариант ответа и запишите только букву выбранного варианта ответа (А, Б, В или Г).

УК-7.1. Вопрос № 6. Какая временная сложность у алгоритма бинарного поиска в отсортированном массиве?

- А) $O(1)$
- Б) $O(\log n)$
- В) $O(n)$
- Г) $O(n \log n)$

Ответ:

УК-7.2. Вопрос № 7. Какая структура данных используется для реализации алгоритма поиска в ширину (BFS)?

- А) Стек
- Б) Очередь
- В) Дек
- Г) Хеш-таблица

Ответ:

УК-7.1. Вопрос № 8. Какой алгоритм позволяет найти кратчайший путь во взвешенном графе с неотрицательными весами рёбер?

- А) Алгоритм Дейкстры
- Б) Алгоритм Краскала
- В) Поиск в глубину (DFS)
- Г) Поиск в ширину (BFS)

Ответ:

УК-7.2. Вопрос № 9. Что означает вердикт «Time Limit Exceeded» (TLE) на соревнованиях по спортивному программированию?

- А) Ошибка компиляции
- Б) Превышение допустимого времени выполнения программы
- В) Неверный ответ программы
- Г) Ошибка во время выполнения программы

Ответ:

УК-7.1. Вопрос № 10. Какое упражнение наиболее эффективно для профилактики синдрома запястного канала у программистов?

- А) Вращение головой
- Б) Упражнения для кистей и пальцев рук
- В) Приседания
- Г) Бег на месте

Ответ:

УК-7.2. Вопрос № 11. Какая платформа является официальной системой тестирования на соревнованиях по спортивному программированию в России?

- А) LeetCode
 - Б) Яндекс.Контест
 - В) HackerRank
 - Г) TopCoder
- Ответ:

УК-7.1. Вопрос № 12. Какой алгоритм использует принцип «разделяй и властвуй»?

- А) Поиск в ширину (BFS)
 - Б) Поиск в глубину (DFS)
 - В) Сортировка слиянием
 - Г) Линейный поиск
- Ответ:

УК-7.2. Вопрос № 13. Какой метод профилактики профессиональных заболеваний предполагает организацию рабочего места с учётом правильной посадки и освещения?

- А) Гимнастика для глаз
 - Б) Эргономика рабочего места
 - В) Производственная гимнастика
 - Г) Дыхательные упражнения
- Ответ:

УК-7.1. Вопрос № 14. Какой алгоритм позволяет найти минимальное остовное дерево в графе?

- А) Алгоритм Дейкстры
 - Б) Алгоритм Краскала
 - В) Алгоритм Флойда–Уоршелла
 - Г) Поиск в глубину (DFS)
- Ответ:

УК-7.2. Вопрос № 15. Для чего используется префикс-функция в алгоритмах обработки строк?

- А) Для поиска подстроки в строке
 - Б) Для сортировки строк
 - В) Для хеширования строк
 - Г) Для сравнения строк
- Ответ:

РАЗДЕЛ 3. ВОПРОСЫ С ОДНИМ АРГУМЕНТИРОВАННЫМ ОТВЕТОМ (№ 16–20)

Инструкция: Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.

УК-7.1. Вопрос № 16. Какое влияние оказывают регулярные занятия спортивным программированием на когнитивные способности человека?

- А) Снижают концентрацию внимания
 - Б) Развивают алгоритмическое мышление, скорость принятия решений и стрессоустойчивость
 - В) Вызывают повышенную тревожность
 - Г) Не влияют на когнитивные способности
- Ответ:

УК-7.2. Вопрос № 17. Почему спортивное программирование может быть рекомендовано как средство профилактики профессиональных заболеваний программиста?

- А) Развивает выносливость
 - Б) Развивает алгоритмическое мышление, концентрацию и стрессоустойчивость, снижает риск эмоционального выгорания
 - В) Укрепляет сердечно-сосудистую систему
 - Г) Способствует набору мышечной массы
- Ответ:

УК-7.1. Вопрос № 18. Какая стратегия наиболее эффективна для решения сложной задачи на соревнованиях по спортивному программированию?

- А) Немедленно приступить к написанию кода
 - Б) Сначала проанализировать ограничения, выбрать оптимальный алгоритм и структуру данных, затем реализовать решение
 - В) Использовать самый простой алгоритм, даже если он неэффективен
 - Г) Сразу сдавать решение без тестирования
- Ответ:

УК-7.2. Вопрос № 19. Какое значение имеет участие в соревнованиях по спортивному программированию для профессионального развития?

- А) Развивает навыки командной работы, стрессоустойчивость и умение решать задачи в условиях ограниченного времени, что востребовано на рынке труда
- Б) Является обязательным требованием для трудоустройства
- В) Заменяет профессиональное образование

Г) Не имеет практического значения

Ответ:

УК-7.1. Вопрос № 20. Какое влияние оказывает регулярное решение олимпиадных задач на профессиональные навыки программиста?

- А) Не влияет на профессиональные навыки
- Б) Развивает алгоритмическое мышление, умение оптимизировать код и находить эффективные решения
- В) Снижает качество кода из-за спешки
- Г) Увеличивает время разработки

Ответ:

РАЗДЕЛ 4. ВОПРОСЫ С НЕСКОЛЬКИМИ ВЕРНЫМИ ОТВЕТАМИ (№ 21–25)

Инструкция: прочитайте текст, выберите правильные варианты ответов (2 или 3 верных варианта из 4) и запишите только буквы выбранных вариантов ответов (А, Б, В или Г).

УК-7.1. Вопрос № 21. Какие качества развивает спортивное программирование?

- А) Алгоритмическое мышление
- Б) Стрессоустойчивость
- В) Агрессивность
- Г) Скорость принятия решений

Ответ:

УК-7.2. Вопрос № 22. Какие меры профилактики профессиональных заболеваний рекомендуются для программистов?

- А) Регулярная гимнастика для глаз и спины
- Б) Организация эргономичного рабочего места
- В) Постоянная работа без перерывов
- Г) Чередование умственной и физической активности

Ответ:

УК-7.1. Вопрос № 23. Какие алгоритмы относятся к алгоритмам на графах?

- А) Алгоритм Дейкстры
- Б) Поиск в глубину (DFS)
- В) Бинарный поиск
- Г) Поиск в ширину (BFS)

Ответ:

УК-7.2. Вопрос № 24. Какие дисциплины входят в состав официального вида спорта «Спортивное программирование» в Российской Федерации?

- А) Алгоритмическое программирование
- Б) Продуктовое программирование (хакатоны)
- В) Веб-дизайн
- Г) Программирование систем информационной безопасности

Ответ:

УК-7.1. Вопрос № 25. Какие факторы влияют на успешность решения задач на соревнованиях по спортивному программированию?

- А) Умение анализировать ограничения задачи
- Б) Знание стандартных алгоритмов и структур данных
- В) Умение работать в команде (для командных соревнований)
- Г) Наличие дорогого компьютера

Ответ:

РАЗДЕЛ 5. ВОПРОСЫ С ОТКРЫТЫМ РАЗВЕРНУТЫМ ОТВЕТОМ (№ 26–30)

Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ.

УК-7.3. Вопрос № 26. Опишите структуру соревнований по спортивному программированию. Какие форматы соревнований существуют (командные и индивидуальные)? Какие платформы используются для проведения соревнований?

Ответ:

УК-7.3. Вопрос № 27. Сравните два алгоритма: бинарный поиск и линейный поиск. Опишите их временную сложность, принцип работы и области применения.

Ответ:

УК-7.3. Вопрос № 28. Объясните, как оценивается сложность алгоритмов в спортивном программировании. Что такое O-нотация? Приведите примеры.

Ответ:

УК-7.3. Вопрос № 29. Опишите роль командной работы на соревнованиях по спортивному программированию (на примере ICPC). Какие роли распределяются в команде и как это влияет на успешность участия?

Ответ:

УК-7.3. Вопрос № 30. Опишите влияние спортивного программирования на здоровье и профессиональное развитие программиста. Какие меры профилактики профессиональных заболеваний следует соблюдать?

Ответ:

РАЗДЕЛ 6. ВОПРОСЫ НА УСТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ (№ 31–32)

Инструкция: Прочитайте текст, установите правильную последовательность действий и запишите аргументы, обосновывающие выбранный порядок.

УК-7.3. Вопрос № 31. Установите правильную последовательность решения задачи на соревнованиях по спортивному программированию:

- А) Написание кода решения
- Б) Чтение и анализ условия задачи
- В) Выбор оптимального алгоритма и структуры данных
- Г) Тестирование на примерах и дополнительных тестах
- Д) Отправка решения в систему тестирования

Ответ:

УК-7.3. Вопрос № 32. Установите правильную последовательность действий при подготовке к соревнованиям по спортивному программированию:

- А) Участие в соревнованиях и анализ результатов
- Б) Решение задач на онлайн-тренажёрах (Codeforces, AtCoder, Яндекс.Контест)
- В) Повторение и изучение основных алгоритмов и структур данных
- Г) Анализ ошибок и разбор решений других участников

Ответ:

5.4. Перечень видов оценочных средств и критерии оценивания

Контроль результатов освоения дисциплины является формой управления качеством образования.

Контроль результатов освоения дисциплины осуществляется в виде текущего контроля и промежуточной аттестации (зачета).

Система текущего контроля включает:

1. контроль работы обучающихся на лекционных и практических занятиях;
2. контроль участия в совместной работе группы;
3. контроль выполнения индивидуальных заданий;
4. контроль выполнения обучающимися заданий для самостоятельной работы.

Промежуточная аттестация осуществляется в виде зачетов в каждом семестре, которые проводятся в устной либо письменной форме.

Задачи промежуточной аттестации:

1. определить общий уровень освоения дисциплины в целом;
2. определить уровень основных знаний по каждой теме курса;
3. объективизировать результат контроля, минимизировать возможную субъективность преподавателя.

Критерии и рубрики оценивания Элемент контроля/Критерий/Индикаторы/Уровни (0–5)/Комментарий к уровню «5»

Практическая работа № 1 «Введение в спортивное программирование»

Элемент контроля: Доклад и презентация / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент представил краткий доклад (5–7 минут) об истории возникновения и развития спортивного программирования, его статусе как официального вида спорта в Российской Федерации; презентация содержит ключевые факты о дисциплинах (алгоритмическое программирование, продуктивное программирование, программирование систем информационной безопасности, программирование БАС) и известных российских спортсменах-программистах; продемонстрировано понимание правил проведения соревнований, структуры турниров и систем тестирования (Яндекс.Контест, Codeforces, AtCoder); выводы аргументированы, работа структурирована, изложена профессиональным языком.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, презентация структурирована, выводы аргументированы.
- 3 балла — задание выполнено с незначительными недочётами (неполное освещение истории, отсутствие примеров известных спортсменов).
- 0 баллов — задание не выполнено или выполнено формально (отсутствие презентации, неправильное понимание дисциплин).

Практическая работа № 2 «Организация и правила соревнований»

Элемент контроля: Отчёт о регистрации и решении задач / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент зарегистрировался на одной из платформ для спортивного программирования (Codeforces, AtCoder, Яндекс.Контест); изучил интерфейс системы тестирования, правила отправки решений, форматы ввода-вывода данных; решил 2–3 простые задачи из архива платформы; разобрался с форматом отчёта об ошибках (WA, TLE, RE, CE); зафиксировал результаты и проанализировал ошибки; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все задачи решены, проведён анализ ошибок.
- 3 балла — задачи решены частично (1–2 из 3), анализ ошибок выполнен поверхностно.

- 0 баллов — регистрация не выполнена или задачи не решены.

Практическая работа № 3 «Языки программирования в спортивном программировании»

Элемент контроля: Сравнительный анализ (отчёт с таблицей) / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент написал решение одной и той же задачи на трёх языках программирования (C++, Python, Java); сравнил время выполнения и объём кода; составил таблицу сравнения языков по критериям: скорость компиляции/интерпретации, удобство написания, доступность стандартных библиотек, наличие встроенных структур данных; сформулировал обоснованный вывод о выборе языка для соревнований; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, таблица заполнена корректно, вывод обоснован.
- 3 балла — решение реализовано только на 2 языках, таблица составлена не полностью.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 4 «Базовые алгоритмические конструкции»

Элемент контроля: Отчёт о решении задач (блок-схемы/псевдокод, код, протокол тестирования) / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент решил 5–7 задач на условные операторы, циклы, работу с массивами и списками; для каждой задачи записал алгоритм решения в виде блок-схемы или псевдокода, реализовал на выбранном языке, протестировал на наборе тестовых данных; зафиксировал время выполнения и использованную память; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все задачи решены корректно.
- 3 балла — решено 3–4 задачи из 5–7, тестирование выполнено не полностью.
- 0 баллов — задачи не решены или решение не представлено.

Практическая работа № 5 «Оценка сложности алгоритмов»

Элемент контроля: Аналитический отчёт с графиками / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент рассчитал теоретическую временную сложность в лучшем, среднем и худшем случаях для 5 алгоритмов; запустил каждый алгоритм на массивах размером 10, 100, 1000, 10 000 элементов; зафиксировал время выполнения, построил график зависимости времени от размера входных данных; сравнил теоретические и экспериментальные данные, объяснил расхождения; проанализировал предпочтительность каждого алгоритма для различных задач; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, графики построены корректно, анализ выполнен глубоко.
- 3 балла — расчёты выполнены для 3–4 алгоритмов, графики построены не для всех случаев.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 6 «Основные структуры данных»

Элемент контроля: Отчёт о реализации структур и решении задач / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент реализовал базовые операции для структур: стек, очередь, дек, ассоциативный массив, множество; для стека и очереди использовал как массив, так и односвязный список; сравнил скорость выполнения операций для разных реализаций; решил 2 задачи на стек и 2 задачи на очередь; зафиксировал результаты и оценил временную сложность операций; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все структуры реализованы, все задачи решены, анализ выполнен.
- 3 балла — структуры реализованы частично (3–4 из 5), задачи решены не все.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 7 «Алгоритмы сортировки и поиска»

Элемент контроля: Отчёт о реализации и сравнении алгоритмов с графиками / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент реализовал алгоритмы сортировки: пузырьковую, быструю, слиянием и сортировку подсчётом; сравнил их производительность на массивах разных типов (случайный, почти отсортированный, обратный порядок) размером 10, 100, 1000, 10 000 элементов; построил графики времени выполнения и числа сравнений; реализовал бинарный и линейный поиск, сравнил их эффективность; решил задачу на бинарный поиск из архива; сделал вывод о применимости каждого алгоритма; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все алгоритмы реализованы и протестированы.
- 3 балла — реализованы 2–3 алгоритма сортировки, графики построены частично.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 8 «Математические алгоритмы»

Элемент контроля: Отчёт о решении задач и реализации функций / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент решил 5–7 задач на использование алгоритма Евклида, бинарное возведение в степень, проверку простоты числа, решение диофантовых уравнений; реализовал функции для НОД, НОК, быстрого возведения в степень; проверил работу на тестах (не менее 10 примеров для каждой функции); сравнил время выполнения с встроенными функциями; проанализировал влияние размера чисел на скорость работы; решил олимпиадную задачу на одну из тем; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все задачи решены.
- 3 балла — решено 3–4 задачи из 5–7, тестирование выполнено не полностью.
- 0 баллов — задачи не решены или решение не представлено.

Практическая работа № 9 «Жадные алгоритмы»

Элемент контроля: Отчёт о решении задач с доказательством корректности / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент решил 3–4 задачи на жадные алгоритмы: задача о выборе заявок, задача о минимальном покрытии отрезков, задача о рюкзаке (дробная версия), задача о размене монет; для каждой задачи сформулировал жадную стратегию, доказал её

корректность, реализовал решение, протестировал на тестовых данных; сравнил с результатом полного перебора для небольших данных; зафиксировал время выполнения; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, доказательство корректности выполнено для всех задач.
- 3 балла — решены 2 задачи из 3–4, доказательство корректности выполнено не для всех.
- 0 баллов — задачи не решены или решение не представлено.

Практическая работа № 10 «Динамическое программирование (базовый уровень)»

Элемент контроля: Отчёт о решении задач с обоснованием состояний / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент решил 3–4 задачи на одномерное и двумерное динамическое программирование (LIS, задача о рюкзаке, LCS, задача о минимальном количестве монет); для каждой задачи определил состояние и рекуррентное соотношение, реализовал итеративное решение, оптимизировал по памяти (если возможно), протестировал на наборе тестов; в отчёт включил обоснование выбора состояния и рекуррентного соотношения; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, обоснование состояний выполнено для всех задач.
- 3 балла — решены 2 задачи из 3–4, обоснование выполнено частично.
- 0 баллов — задачи не решены или решение не представлено.

Практическая работа № 11 «Динамическое программирование (продвинутый уровень)»

Элемент контроля: Отчёт о решении задач с анализом времени выполнения / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент решил 2–3 задачи на динамическое программирование на подотрезках и 2 задачи на динамическое программирование на подмножествах (TSP, задача о назначениях); реализовал решения с использованием DP по маскам; сравнил время выполнения для разных размеров входных данных ($n = 10, 15, 20$); проанализировал экспоненциальный рост времени выполнения; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, анализ времени выполнения выполнен.
- 3 балла — решены 2–3 задачи из 4–5, анализ времени выполнения выполнен частично.
- 0 баллов — задачи не решены или решение не представлено.

Практическая работа № 12 «Теория графов и алгоритмы на графах»

Элемент контроля: Отчёт о реализации алгоритмов и решении задач / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент реализовал представление графов (список смежности, матрица смежности); решил 2 задачи на DFS и 2 задачи на BFS; реализовал алгоритм Дейкстры и решил 2 задачи на его применение; реализовал DSU и алгоритм Краскала; протестировал алгоритмы на нескольких графах (не менее 5); сравнил производительность алгоритмов; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все алгоритмы реализованы и протестированы.
- 3 балла — алгоритмы реализованы частично (3–4 из 5), тестирование выполнено не полностью.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 13 «Работа со строками»

Элемент контроля: Отчёт о реализации алгоритмов поиска / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент реализовал алгоритмы поиска подстроки: префикс-функция, z-функция, КМП; рассчитал время выполнения на строках разной длины (100, 1000, 10 000, 100 000 символов); сравнил производительность алгоритмов; решил задачу на поиск всех вхождений подстроки с использованием каждого из трёх алгоритмов; реализовал хеширование строк и решил задачу с использованием хешей; отчёт структурирован, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, все алгоритмы реализованы и протестированы.
- 3 балла — реализованы 2 из 3 алгоритмов, хеширование выполнено частично.
- 0 баллов — задание не выполнено или выполнено формально.

Практическая работа № 14 «Продуктивное программирование (хакатоны)»

Элемент контроля: Проект и презентация (командная работа) / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент в составе команды разработал прототип приложения, веб-сервиса или телеграм-бота за 4–6 часов; распределил роли в команде; подготовил презентацию (5–7 минут) с демонстрацией работающего прототипа; провёл ретроспективу (что удалось, что не удалось, какие навыки нужно развивать); защитил проект перед преподавателем или группой; работа структурирована, выводы аргументированы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, прототип работает, ретроспектива проведена.
- 3 балла — прототип работает частично, ретроспектива выполнена поверхностно.
- 0 баллов — проект не выполнен или не представлен.

Практическая работа № 15 «Здоровьесбережение в профессиональной деятельности»

Элемент контроля: Отчёт с дневником самооценки и презентация / УК-7.1, УК-7.2, УК-7.3 / 0 / 3 / 5 / Студент провёл самооценку психоэмоционального состояния до и после тренировки с использованием опросников; оценил уровень утомления, концентрации внимания и стресса; разработал индивидуальный план профилактики профессиональных заболеваний (организация рабочего места, гимнастика, режим труда и отдыха, дыхательные упражнения); составил чек-лист здоровьесберегающих привычек на неделю; подготовил краткий доклад (5–7 минут) с презентацией рекомендаций; зафиксировал результаты, провёл анализ и сформулировал выводы.

Уровни:

- 5 баллов — все требования выполнены в полном объёме, план проработан, чек-лист составлен, выводы обоснованы.
- 3 балла — самооценка выполнена, но план разработан поверхностно или чек-лист составлен не полностью.
- 0 баллов — задание не выполнено или выполнено формально.

Тест

Корректность ответов на вопросы теста / % → балл / $\geq 90\%$ правильных ответов — 5; 75–89% — 4; 60–74% — 3; $< 60\%$ — 2.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

1. Трофимов, В. В., Павловская, Т. А. Алгоритмизация и программирование : учебник для вузов / под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2022. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491215>
2. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — URL: <https://urait.ru/bcode/513067>
3. Зыков С. В. Программирование : учебник и практикум для вузов. — Москва : Издательство Юрайт, 2023. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — URL: <https://urait.ru/bcode/489754> .
4. Кудрявцева И. А., Швецкий М. В. Программирование: комбинаторная логика : учебное пособие для СПО. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 524 с. — (Профессиональное образование). — ISBN 978-5-534-15128-2. — URL: <https://urait.ru/bcode/542030> .
5. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561885>.
6. Федоров Д. Ю. Программирование на Python : учебное пособие для вузов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — URL: <https://urait.ru/bcode/556864> .
7. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования : учебное пособие для вузов. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — URL: <https://urait.ru/bcode/536903>.

6.2. Электронные учебные издания и электронные образовательные ресурсы

Э1	Сетевые ресурсы Государственного автономного учреждения культуры Свердловской области «Свердловская областная универсальная научная библиотека им. В.Г. Белинского». Режим доступа: https://book.uraic.ru/el_library/resursy_dostup
Э2	Образовательная платформа Юрайт. Режим доступа: https://urait.ru/
Э3	Научная библиотека Cyberleninka. Режим доступа: https://cyberleninka.ru/

6.3.1 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства

6.3.1.1	Операционная система Astra Linux
6.3.1.2	Офисная система LibreOffice
6.3.1.3	Интерпретатор Python 3, компилятор GCC (свободно распространяемое ПО)
6.3.1.4	Контур.Класс
6.3.1.5	Контур.Толк
6.3.1.6	1С:Университет ПРОФ
6.3.1.7	Среда электронного обучения 3LK (русский Moodle)

7. МТО (оборудование и технические средства обучения)

7.1	Учебные аудитории для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенные оборудованием и техническими средствами обучения: Адрес (местоположение) учебных аудиторий Перечень оборудования и технических средств обучения 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 3 площадью 66,2 м² в помещении 501 Стол – 13 шт. Стул – 25 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Телевизор (75”) – 1 шт. Беспроводная система презентаций – 1 шт. Спикерфон – 1 шт. ПК (ноутбук) – 12 шт. 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 19 площадью 31,7 м² в помещении 501 Стол – 7 шт. Стул – 13 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Магнитно-маркерная доска – 1 шт. Телевизор (75”) – 1 шт. 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет №
-----	---

	21 площадью 35,6 м² в помещении 501 Стол – 7 шт. Стул – 13 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Магнитно-маркерная доска – 1 шт. Телевизор (75”) – 1 шт. Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета: Адрес (местоположение) помещения для самостоятельной работы обучающихся Перечень компьютерной техники и оборудования 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 5 площадью 44,3 м² в помещении 501 ПК (ноутбук) – 9 шт. Стол – 11 шт. Стул – 11 шт.
--	--

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины рекомендуется ознакомиться с рабочей программой, целями и задачами дисциплины, её связями с другими дисциплинами образовательной программы. Также рекомендуется изучить основную и дополнительную литературу, перечень заданий. При подготовке к практическим занятиям изучить основную литературу, ознакомиться с дополнительной и методической литературой, учесть рекомендации преподавателя. В ходе занятий вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных понятий, явлений и процессов, научные выводы и практические рекомендации по их применению. При пропуске занятия самостоятельно изучить материал и предоставить преподавателю отчёт по пропущенной теме в часы индивидуальных консультаций. При подготовке к текущей аттестации тщательно изучать материал по блокам тем, акцентировать внимание на определениях, терминах, содержании понятий. Для этого рекомендуется использовать основную литературу. При подготовке к промежуточной аттестации в дополнение к изучению словарей, учебных пособий, целесообразно пользоваться учебной литературой, рекомендованной в программе, и электронными ресурсами. При подготовке к зачёту или экзамену организовать учёбу так, чтобы все виды работ и заданий, предусмотренные рабочей программой, были выполнены в срок. Основное в подготовке — это повторение всего материала учебной дисциплины.