Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 05.08.2026 14:58:51
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет химической технологии и биотехнологии

УТВЕРЖДАЮ
Декан
_____ / А.С. Соколов /
«26» февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы алгоритмизации и программирования»

Направление подготовки
15.03.04 Автоматизация технологических процессов и производств

Профиль «Автоматизация технологических производств»

Квалификация
Бакалавр

Форма обучения
Очная

Москва, 2026

Разработчик:

Доцент, к.т.н. доцент кафедры «Аппаратурное оформление и автоматизация технологических производств имени профессора М.Б. Генералова»



/Д.В. Зубов /

Согласовано:

Заведующий кафедрой «Аппаратурное оформление и автоматизация технологических производств имени профессора М.Б. Генералова»,



/А.С. Кирсанов/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 6
3.1 Виды учебной работы и трудоемкость 6
3.2 Тематический план изучения дисциплины 6
3.3 Содержание дисциплины 7
3.4 Тематика семинарских/практических и лабораторных занятий 7
3.5 Тематика курсовых проектов (курсовых работ) 8
4 Учебно-методическое и информационное обеспечение 8
4.1 Нормативные документы и ГОСТы 8
4.2 Основная литература 8
4.3 Дополнительная литература 8
4.4 Электронные образовательные ресурсы 9
4.5 Лицензионное и свободно распространяемое программное обеспечение 9
4.6 Современные профессиональные базы данных и информационные справочные системы 9
5 Материально-техническое обеспечение 9
6 Методические рекомендации 9
6.1 Методические рекомендации для преподавателя по организации обучения 9
6.2 Методические указания для обучающихся по освоению дисциплины 10
7 Фонд оценочных средств 11
7.1 Методы контроля и оценивания результатов обучения 11
7.2 Шкала и критерии оценивания результатов обучения 13
7.3 Оценочные средства 17

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины «Основы алгоритмизации и программирования» является формирование у студентов фундаментальных знаний, навыков и практической компетенции в области алгоритмизации и разработки алгоритмов для решения разнообразных задач. Кроме того, дисциплина направлена на развитие логического и абстрактного мышления, способности анализа и проектирования алгоритмов, а также на овладение основами использования различных типов данных и структур данных.

Задачи изучения дисциплины:

- Ознакомление со структурой и принципами работы алгоритмов.
- Понимание различных видов представлений алгоритмов и их применения.
- Формирование умения анализировать задачи и выбирать оптимальные алгоритмические решения.
- Развитие умения программировать и реализовывать алгоритмы на практике.
- Применение полученных знаний и навыков для решения задач из различных областей.

Обучение по дисциплине «Основы алгоритмизации и программирования» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**	**Наименование показателя оценивания**
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ИОПК-4.1. Знает основы информационных технологий ИОПК-4.2. Умеет выполнять практические работы по настройке компьютерной техники ИОПК-4.3. Владеет навыками работы с прикладным программным обеспечением ИОПК-4.4. Понимает принципы работы современных информационных технологий и программных средств, применимых для решения задач профессиональной деятельности; ИОПК-4.5. Выбирает информационные технологии и программные средства для решения задач профессиональной деятельности; ИОПК-4.6. Применяет современные информационные технологии и программные средства для решения задач профессиональной деятельности.	**Знать:** Основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования. **Уметь:** Анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения **Владеть:** Навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования.

ОПК-14. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-14.1. Понимает основные подходы к построению алгоритмов и языки программирования, применимые для написания компьютерных программ; ОПК-14.2. Применяет основные алгоритмические структуры для написания компьютерных программ, пригодных для практического применения; ОПК-14.3. Осуществляет написание и отладку компьютерных программ, пригодных для практического применения.	**Знать:** Основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании. Принципы работы с языком программирования. **Уметь:** Разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные конструкции языка программирования. **Владеть:** Навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части блока Б1 «Дисциплины (модули)». Дисциплина непосредственно связана со следующими дисциплинами и практиками ООП:

Интеллектуальный анализ данных;

Информационные технологии;

Комплексы технических средств в системах автоматического управления;

Компьютерное зрение;

Объектно-ориентированное программирование;

Цифровая грамотность.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часов).

3.1 Виды учебной работы и трудоемкость

№ п/п	Вид учебной работы	Количество часов	Семестр
			3
1	**Аудиторные занятия**	**54**	**54**
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия	36	36
1.3	Лабораторные работы	–	–
2	**Самостоятельная работа**	**54**	**54**
	В том числе:		

2.1	Подготовка отчётов по практическим работам	18	18
2.2	Работа с конспектом лекций	20	20
2.3	Подготовка к зачёту	16	16
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен		зачет
	Итого	**108**	**108**

3.2 Тематический план изучения дисциплины

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	**Аудиторная работа**				**Самостоятельная работа**
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1.1	Основные понятия теории алгоритмов. Анализ вычислительной сложности алгоритмов. Необходимые ресурсы для реализации алгоритмов. Основные алгоритмы. Формы записи алгоритмов.	10	2	2			6
1.2	Современные языки программирования и области их применения. Выбор языка программирования для решения прикладных задач.	10	2	2			6
1.3	Рекурсивные алгоритмы. Реентерабельные процедуры.	10	2	2			6
1.4	Алгоритмы поиска. Работа с хеш-таблицами.	9	2	2			5
1.5	Алгоритмы аппроксимации и фильтрации.	11	2	4			5
1.6	Работа с файлами, портами ввода/вывода, сетевыми портами	11	4	4			5
1.7	Работа с IDE (интегрированные среды разработки)	10	2	4			4
1..8	Вайб-программирование	23	2	16			5
	Итого	**108**	**18**	**36**			**54**

3.3 Содержание дисциплины

Тема 1.

Основные понятия теории алгоритмов. Анализ вычислительной сложности алгоритмов (линейная сложность, экспоненциальная сложность, факториальная сложность). Необходимые ресурсы для реализации алгоритмов оперативная память, место на жёстких дисках. Основные алгоритмы. Формы записи алгоритмов.

Тема 2.

Современные языки программирования и области их применения. Выбор языка программирования для решения прикладных задач. Рассматриваются компилируемые и

интерпретируемые языки (C++, C#, Python, Java, Go), области их применения, достоинства и недостатки. Явная и неявная типизация. Борьба с утечками памяти.

Тема 3.
Рекурсивные алгоритмы. Реентерабельные процедуры. Семафоры.

Тема 4.
Алгоритмы поиска в линейном списке, бинарном дереве. Работа с хеш-таблицами.

Тема 5.
Работа с файлами, потоками ввода/вывода, портами ввода/вывода, сетевыми портами.

Тема 6.
Работа с IDE (интегрированные среды разработки).

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Семинар 1. Выбор языка программирования
Семинар 2. Рекурсивные алгоритмы в решении учебных и научных задач
Семинар 3. Алгоритмы аппроксимации
Семинар 4. Работа с портами ввода/вывода.
Семинар 5. Выбор IDE

Практическое занятие 1. Графическое представление алгоритмов
Практическое занятие 2. Алгоритмы поиска
Практическое занятие 3. Фильтрация данных с датчиков, с учётом разного периода опроса, дискретности и точности.
Практическое занятие 4. Запись в файл/чтение из файла, работа с сетевыми портами (открыть, послать, считать, закрыть)
Практическое занятие 5. Работа в IDE
Практическое занятие 6. Создание прикладных программ с помощью ИИ (Gemini, Claude AI)
Практическое занятие 7. Анализ программ, созданных с помощью ИИ
Практическое занятие 8-13. Развёртывание локальных LLM для программирования и создание прикладных программных проектов с помощью ИИ, обученного на локальных разработках.

3.4.2 Лабораторные занятия

Не предусмотрены

3.5 Тематика курсовых проектов (курсовых работ)

Не предусмотрены

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. ГОСТ Р 59795-2021. Автоматизированные системы. Требования к содержанию документов. https://docs.cntd.ru/document/1200181351

4.2 Основная литература

2. Трофимов, В.В. Алгоритмизация и программирование: учебник для вузов / В.В. Трофимов, Т.А. Павловская. – Москва: Издательство Юрайт, 2024. – https://urait.ru/viewer/algoritmizaciya-i-programmirovanie-538039

3. Воробейчиков, Л. А. Алгоритмизация и программирование: учебное пособие / Л. А. Воробейчиков, А. В. Загвоздкина, В. Н. Шакин. — Москва: МТУСИ, 2022. — 350 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/333788

4. Кудинов, Ю. И. Основы современной информатики: учебное пособие для вузов / Ю. И. Кудинов, Ф. Ф. Пащенко. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 256 с. — ISBN 978-5-8114-8251-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/392393

4.3 Дополнительная литература

1. Практикум по информатике: учебное пособие / Н. М. Андреева, Н. Н. Василюк, Н. И. Пак, Е. К. Хеннер. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 248 с. — ISBN 978-5-8114-2961-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/205961

2. Зверева, Н. А. Информатика: практикум: учебное пособие / Н. А. Зверева. — Иркутск: ИрГУПС, 2019. — 104 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/157934

3. Судоплатов, С. В. Математическая логика и теория алгоритмов: учебник и практикум для вузов / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва: Издательство Юрайт, 2023. — 207 с. — (Высшее образование). — ISBN 978-5-534-12274-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/535807

4.4 Электронные образовательные ресурсы

ЭОР не разработан.

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Microsoft-Office
2. IDE Spyder
3. Python
4. Microsoft-Windows

4.6 Современные профессиональные базы данных и информационные справочные системы

1. Научная электронная библиотека http://www.elibrary.ru
2. Российская государственная библиотека http://www.rsl.ru
3. Единое окно доступа к образовательным ресурсам Федеральный портал http://window.edu.ru
4. ЭБС «Университетская библиотека онлайн» https://biblioclub.ru/index.php

5 Материально-техническое обеспечение

1. Компьютерный класс с предустановленным программным обеспечением, указанным в п. 4.5, мультимедийное оборудование (проектор, персональный компьютер преподавателя).
2. Аудитория для лекционных, практических занятий. Оборудование и аппаратура: аудиторная доска, возможность использования мультимедийного комплекса.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

На первом занятии по дисциплине необходимо ознакомить студентов с порядком ее изучения (темами курса, формами занятий, текущего и промежуточного контроля), раскрыть место и роль дисциплины в системе наук, ее практическое значение, довести до студентов требования к форме отчетности и применения видов контроля.

При подготовке к практическим работам по перечню объявленных тем преподавателю необходимо уточнить план их проведения, продумать формулировки и содержание учебных вопросов, выносимых на обсуждение.

В ходе работы во вступительном слове раскрыть практическую значимость темы работы, определить порядок ее проведения, время на обсуждение каждого учебного вопроса.

Применяя фронтальный опрос дать возможность выступить всем студентам, присутствующим на занятии.

В заключительной части работы следует подвести ее итоги: дать оценку выступлений каждого студента и учебной группы в целом. Раскрыть положительные стороны и недостатки проведенной работы. Ответить на вопросы студентов. Выдать задания для самостоятельной работы по подготовке к следующему занятию.

Методика преподавания дисциплины «Основы алгоритмизации и программирования» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– подготовка к выполнению и защита практических работ с помощью специализированного программного обеспечения;

– технологии анализа ситуаций для активного обучения, которые позволяют студентам соединить теорию и практику, представить примеры принимаемых решений и их последствий, демонстрировать различные позиции, формировать навыки оценки альтернативных вариантов в вероятностных условиях.

Обучение по дисциплине ведется с применением традиционных потоково-групповых информационно-телекоммуникационных технологий. При осуществлении образовательного процесса по дисциплине используются следующие информационно-телекоммуникационные технологии: презентации с применением проектора и программы PowerPoint.

6.2 Методические указания для обучающихся по освоению дисциплины

Самостоятельная работа является одним из видов учебных занятий. Цель самостоятельной работы – практическое самостоятельное получение студентами навыков работы в программных продуктах, рассматриваемых в процессе изучения дисциплины.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Задачи самостоятельной работы студента:

- развитие навыков самостоятельной учебной работы;
- освоение содержания дисциплины;
- углубление содержания и осознание основных понятий дисциплины;
- использование материала, собранного и полученного в ходе самостоятельных занятий для эффективной подготовки к зачёту.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к практическим занятиям;
- оформление отчетов по выполненным практическим работам;
- подготовка к зачёту.

Для выполнения любого вида самостоятельной работы необходимо пройти следующие этапы:

- определение цели самостоятельной работы;
- конкретизация познавательной задачи;
- самооценка готовности к самостоятельной работе;
- выбор адекватного способа действия, ведущего к решению задачи;
- планирование работы (самостоятельной или с помощью преподавателя) над заданием;
- осуществление в процессе выполнения самостоятельной работы самоконтроля (промежуточного и конечного) результатов работы и корректировка выполнения работы;
- рефлексия;
- презентация работы.

7 Фонд оценочных средств

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций

- отчёты по практическим работам;
- контрольные работы;
- зачёт.

Оценочные средства текущего контроля успеваемости включают контрольные задания по практическим работам индивидуально для каждого обучающегося.

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	Наименование компетенции выпускника
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

ОПК-14	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

7.1 Методы контроля и оценивания результатов обучения

Перечень оценочных средств по дисциплине «Основы алгоритмизации и программирования»

№ п/п	Вид контроля результатов обучения	Наименование контроля результатов обучения	Краткая характеристика контроля результатов обучения
1	Текущий	Практическая работа	Практическая работа выполняется индивидуально каждым студентом. По результатам выполнения работы студент оформляет отчёт, содержащий подробное описание проделанной работы с наглядным представлением результатов. Оформленный отчет студент сдает преподавателю на проверку в заранее установленный срок. При проверке преподаватель оценивает качество оформления, корректность проделанных шагов, результатов работы и выводов. Часть работ подразумевает устную защиту в формате доклада/презентации.
2	Текущий	Контрольная работа	Контрольная работа выполняется индивидуально каждым студентом. Задание подразумевает выполнения ряда шагов с использованием программного обеспечения, изучаемого в соответствующей теме. Результатом выполнения контрольной работы является электронный документ с кратким описанием выполненных шагов и полученных результатов. При проверке преподаватель оценивает как результат работы, так и путь достижения результата.
			Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине

3	Промежуточный	зачёт	(модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено». Во время проведения зачёта студенту предлагается выполнить 3 практических задания. По результату выполнения студенту задаются вопросы на понимание выполненных задач. Количество дополнительных вопросов зависит от полноты ответа студента. К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Основы алгоритмизации и программирования»

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**

знать: - основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования; - основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: - основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования; - основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании.	Обучающийся демонстрирует неполное соответствие следующих знаний: - основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования; - основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний: - основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования; - основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний: - основные принципы функционирования информационных технологий и программного обеспечения. Современные тренды и технологии в области алгоритмизации и программирования; - основные концепции и принципы разработки алгоритмов. Структуры данных и их использование в программировании. Свободно оперирует приобретенными знаниями.
уметь: - анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения; - разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные конструкции языка программирования.	Обучающийся не умеет или в недостаточной степени умеет: - анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения; - разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные конструкции языка программирования.	Обучающийся демонстрирует неполное соответствие следующих умений: - анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения; - разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные	Обучающийся демонстрирует частичное соответствие следующих умений: - анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения; - разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные	Обучающийся демонстрирует полное соответствие следующих умений: - анализировать требования задачи и выбирать подходящие информационные технологии и программные средства для её решения; - разрабатывать алгоритмы для решения различных задач. Преобразовывать алгоритмы в компьютерные программы на выбранном языке программирования. Понимать и использовать основные

		конструкции языка программирования. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	конструкции языка программирования. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	конструкции языка программирования. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: - навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования; - навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования.	Обучающийся не владеет или в недостаточной степени владеет - навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования; - навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования.	Обучающийся в недостаточной степени владеет: - навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования; - навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет: - навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования; - навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования. Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет: - навыками эффективно использовать информационные технологии и программные средства для решения практических задач в области алгоритмизации и программирования; - навыками написания чистого, структурированного и эффективного кода, отладки и тестирования программ и самостоятельного изучения новых языков и технологий программирования. Свободно применяет полученные навыки в ситуациях повышенной сложности.

Шкала оценивания промежуточной аттестации: зачёт

Шкала оценивания	Описание

Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Шкала оценивания текущего контроля

Наименование контроля результатов обучения	Шкала оценивания	Описание
Практическая работа	Зачтено: набрано 2 и более баллов. Не зачтено: набрано 1 и менее баллов. Критерии оценивания: Общий балл при оценке складывается из следующих показателей: - оформление работы соответствует требованиям – 1 балл; - все пункты задания выполнены в полном объёме – 2 балла; - изложение, описание и выводы по работе грамотны и полно описывают содержание практической работы – 2 балла.	В качестве форм текущего контроля знаний студентов используются отчеты по практической работе Отчёт по практической работе содержит описания ряда шагов по выполнению практической работы согласно заданию с подробным описанием проделанных действий и полученными результатами. Защита практических работ (если требуется) осуществляется индивидуально. Студентом предоставляется оформленный отчёт. Студенты, не выполнившие практическую работу, к защите не допускаются

Контрольная работа	Отлично - Работа высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, либо некоторые из выполненных заданий содержат незначительные ошибки Хорошо - Уровень выполнения работы отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, некоторые виды заданий выполнены с ошибками. Удовлетворительно -	Защита темы включает выполнения практического задания по изученному материалу в аудитории в течении одной пары. Билеты состоят из задач, позволяющих оценить сформированные компетенции. На решение отводится 1.5 часа.
	Теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой заданий не выполнено; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий. Неудовлетворительно - Теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, предусмотренные программой задания не выполнены	

7.3 Оценочные средства

7.3.1 Текущий контроль

Типовое задание «Контрольная работа №1»

Темы:

1. Основы алгоритмизации: понятие алгоритма, структуры данных, виды представлений алгоритмов.
2. Численные алгоритмы: численные методы, аппроксимация.
3. Сортировки и поиск: сортировочные алгоритмы, алгоритмы поиска.

Задания:

1. Опишите понятие алгоритма. Приведите примеры различных структур данных.

2. Реализуйте алгоритм нахождения корней квадратного уравнения с использованием метода Ньютона.
3. Дан массив чисел. Напишите программу для сортировки этого массива методом пузырька.
4. Решите задачу о поиске наибольшей общей подпоследовательности (LCS) двух строк.

Типовое задание «Контрольная работа №2»

Темы:

1. Рекурсивные алгоритмы: понятие рекурсии, рекурсивные вызовы, оптимизация рекурсии.
2. Построение деревьев решений: принципы построения, алгоритмы на графах.

Задания:

1. Напишите рекурсивную функцию для вычисления чисел Фибоначчи. Оцените её сложность.
2. Разработайте алгоритм для построения дерева решений выбора оптимального пути в игре "Крестики-нолики".
3. Дан ориентированный граф. Напишите программу для обхода графа в глубину (DFS) и поиска компонент связности.

Типовое задание «Контрольная работа №3»

Темы:

1. Хэш-таблицы: принципы работы, методы разрешения коллизий.
2. Алгоритмы поиска кратчайшего пути: алгоритм Дейкстры, алгоритм Флойда-Уоршелла.

Задания:

1. Реализуйте хэш-таблицу для хранения контактов (имя - номер телефона). Обеспечьте добавление, поиск и удаление записей.
2. Напишите алгоритм Дейкстры для нахождения кратчайшего пути между вершинами графа.
3. Решите задачу о нахождении остовного дерева минимального веса во взвешенном графе.

7.3.2 Промежуточная аттестация

Вопросы к зачету

1. Что такое алгоритм? Приведите определение и основные характеристики.	ОПК-4
2. В чем заключается процесс алгоритмизации? Почему он важен при решении задач?	ОПК-4
3. Какие основные этапы включает процесс разработки алгоритма?	ОПК-4
4. Какие виды представлений алгоритмов вы знаете? Приведите примеры каждого вида.	ОПК-4
5. Что такое структура данных? Какие основные типы данных используются при алгоритмизации?	ОПК-4
6. Опишите понятие "массив" и объясните, как происходит доступ к элементам массива.	ОПК-4
7. В чем разница между линейными и связанными списками? Приведите примеры использования каждой структуры.	ОПК-4
8. Какие типы структур данных обеспечивают быстрый поиск и вставку элементов?	ОПК-4
9. Что представляют собой численные методы? Какие задачи они помогают решать?	ОПК-4
10. Объясните принцип работы метода Ньютона для нахождения корней функции.	ОПК-4

11. Что такое аппроксимация? Приведите примеры областей, где применяются численные методы аппроксимации.	ОПК-4
12. Какие численные методы используются для решения дифференциальных уравнений?	ОПК-4
13. Объясните, как работает алгоритм сортировки пузырьком. Какова его сложность?	ОПК-4
14. Какой алгоритм поиска подходит для упорядоченного массива? Объясните его принцип работы.	ОПК-4
15. Решите задачу: дан массив чисел. Напишите программу для сортировки этого массива методом выбора.	ОПК-14
16. Примените алгоритм бинарного поиска для нахождения индекса элемента в упорядоченном массиве.	ОПК-14
17. Что такое рекурсивный алгоритм? Какие задачи удобно решать с его помощью?	ОПК-14
18. Какой базовый принцип лежит в основе рекурсивных вызовов?	ОПК-14
19. Напишите рекурсивную функцию для вычисления факториала числа.	ОПК-14
20. Решите задачу: напишите рекурсивную функцию для вычисления чисел Фибоначчи.	ОПК-14
21. Что такое дерево решений? Какие задачи можно решать с его помощью?	ОПК-14
22. Каким образом происходит построение дерева решений? Приведите пример.	ОПК-14
23. Расскажите о принципе работы алгоритма ID3 для построения дерева решений.	ОПК-14
24. Какие преимущества и ограничения свойственны деревьям решений?	ОПК-14
25. Что такое сетевой граф? В чем состоит задача поиска кратчайшего пути в сетевом графе?	ОПК-14
26. Объясните принцип работы алгоритма Дейкстры для поиска кратчайшего пути.	ОПК-14
27. Какие дополнительные возможности предоставляет алгоритм Беллмана-Форда?	ОПК-14
28. Решите задачу: дан ориентированный граф с весами рёбер. Напишите программу для нахождения кратчайшего пути между двумя вершинами.	ОПК-14
29. В чем заключается идея распределённых алгоритмов? Какие преимущества они предоставляют?	ОПК-14
30. Объясните понятие "распределённая вычислительная сеть".	ОПК-14
31. Как работает алгоритм обхода графа в ширину в распределённой вычислительной сети?	ОПК-14
32. Расскажите о применении распределённых алгоритмов в реальных системах.	ОПК-14
33. Какие способы графического представления алгоритмов вы знаете? Приведите примеры.	ОПК-14
34. Нарисуйте диаграмму алгоритма сортировки пузырьком.	ОПК-14
35. Поясните, как можно использовать графическое представление для анализа сложности алгоритмов.	ОПК-14
36. Какие задачи можно решать с помощью численных методов? Приведите примеры.	ОПК-14
37. Напишите программу для нахождения приближенного значения корня уравнения методом бисекции.	ОПК-14
38. Примените численные методы для решения задачи интерполяции.	ОПК-14
39. Что такое хэш-таблица? Какие методы разрешения коллизий вы знаете?	ОПК-14
40. Реализуйте хэш-таблицу для хранения информации о студентах (фамилия - номер группы).	ОПК-14
41. Напишите программу для поиска студента в хэш-таблице по фамилии.	ОПК-14