Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 21.05.2026 18:44:38
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет машиностроения

УТВЕРЖДАЮ

Декан

_____ /Е.В. Сафонов/

«26» февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование на Python

Направление подготовки

15.04.04 Автоматизация технологических процессов и производств

Профиль

Промышленная автоматизация

Квалификация

Магистр

Формы обучения

очная

Москва, 2026 г.

Разработчик(и):

Доцент кафедры «Автоматика и управление», к.т.н.  / Т.А. Лисовская

Согласовано:

Заведующий кафедрой «Автоматика и управление», д.т.н., профессор  / А.А. Радионов

Руководитель образовательной программы
Профессор кафедры «Автоматика и управление», д.т.н., доцент  / В.Р. Гасияров

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 6
3.5 Тематика курсовых проектов (курсовых работ) 7
4 Учебно-методическое и информационное обеспечение 7
4.1 Нормативные документы и ГОСТы 7
4.2 Основная литература 7
4.3 Дополнительная литература 7
4.4 Электронные образовательные ресурсы 8
4.5 Лицензионное и свободно распространяемое программное обеспечение 8
4.6 Современные профессиональные базы данных и информационные справочные системы 8
5 Материально-техническое обеспечение 8
6 Методические рекомендации 8
6.1 Методические рекомендации для преподавателя по организации обучения 8
6.2 Методические указания для обучающихся по освоению дисциплины 9
7 Фонд оценочных средств 10
7.1 Методы контроля и оценивания результатов обучения 10
7.2 Шкала и критерии оценивания результатов обучения 12
7.3 Оценочные средства 15

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины «Программирование на Python» 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 «Программирование на Python» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**	**Наименование показателя оценивания**
ОПК-12. Способен разрабатывать и оптимизировать алгоритмы и современные цифровые системы автоматизированного проектирования технологических процессов, создавать программы изготовления деталей и узлов различной сложности на станках с числовым программным управлением, проектировать алгоритмы функционирования гибких производственных систем	ИОПК-12.1. Разрабатывает алгоритмы автоматизированного проектирования технологических процессов. ИОПК-12.2. Создает и оптимизирует программы для станков с числовым программным управлением и гибких производственных систем. ИОПК-12.3. Оценивает эффективность разработанных алгоритмов и цифровых систем автоматизированного проектирования.	**Знать:** основы автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python **Уметь:** разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы **Владеть:** навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода.

2 Место дисциплины в структуре образовательной программы

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

SCADA-системы в автоматизированном производстве;

Интеллектуальные системы управления;

Машинное обучение.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетные единицы (72 часов).

3.1 Виды учебной работы и трудоемкость

№ п/п	Вид учебной работы	Количество часов	Семестр
			2
1	**Аудиторные занятия**	**36**	**36**
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия	18	18
1.3	Лабораторные работы	–	–
2	**Самостоятельная работа**	**36**	**36**
	В том числе:		
2.1	Подготовка отчётов по практическим работам	9	9
2.2	Работа с конспектом лекций	9	9
2.3	Подготовка к диф.зачёту	18	18
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен		зачет
	Итого	**72**	**72**

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	**Раздел 1. Основы алгоритмизации**	**20**	**6**	**4**			**10**
1.1	Тема 1. Основные понятия и классификации алгоритмов.		2	1			2
1.2	Тема 2. Типы и структуры данных		2	2			2
1.3	Тема 3. Анализ алгоритма		2	1			6
2	**Раздел 2. Численные алгоритмы**	**14**	**4**	**4**			**6**
2.1	Тема 1. Численные методы в алгоритмах		2	2			4

2.2	Тема 2. Аппроксимация		2	2			2
3	**Раздел 3. Сортировки и поиск**	**14**	**4**	**4**			**6**
3.1	Тема 1. Сортировочные алгоритмы		2	2			4
3.2	Тема 2. Алгоритмы поиска		2	2			2
4	**Раздел 4. Сетевые и распределённые алгоритмы**	**24**	**4**	**6**			**14**
4.1	Тема 1. Деревья решений. Построение и использование деревьев решений		1	2			8
4.2	Тема 2. Сетевые алгоритмы. Алгоритмы на графах: поиск кратчайшего пути, алгоритмы обхода графов.		2	2			4
4.3	Тема 3. Распределённые алгоритмы		1	2			2
	Итого	**72**	**18**	**18**			**36**

3.3 Содержание дисциплины

Раздел 1. Основы алгоритмизации

В данном разделе студенты изучат базовые понятия алгоритмизации, изучат различные классификации алгоритмов и их представлений, а также приобретут понимание ключевых понятий, связанных с разработкой алгоритмов. Также будут рассмотрены основные типы и структуры данных для построения адекватных алгоритмах для их обработки.

Раздел 2. Численные алгоритмы

В этом разделе студенты получат основы численных методов и алгоритмов, которые используются для приближенного решения математических задач. Они изучат различные численные методы решения уравнений, а также методы аппроксимации функций. Этот раздел позволит студентам понять, как численные алгоритмы могут быть применены в реальных задачах, требующих вычислительных приближений.

Раздел 3. Сортировки и поиск

В данном разделе студенты расширят свои знания о различных алгоритмах сортировки и поиска. Они изучат разные методы сортировки данных и научатся анализировать их эффективность. Кроме того, студенты будут исследовать алгоритмы поиска элементов в массивах и прочих структурах данных, а также познакомятся с понятием хэш-таблиц и способами их использования. Важной частью данного раздела будет ознакомление с рекурсивными алгоритмами, которые позволяют эффективно решать задачи, разбивая их на более простые подзадачи.

Раздел 4. Сетевые и распределённые алгоритмы

В этом разделе студенты изучат алгоритмы, связанные с обработкой данных в сетевых и распределенных системах. Они познакомятся с понятием деревьев решений и научатся строить и использовать такие структуры для принятия решений. Студенты будут исследовать алгоритмы поиска кратчайшего пути и обхода графов. Раздел также охватывает распределенные алгоритмы, которые позволяют эффективно решать задачи на нескольких компьютерах или устройствах, что является важной частью современных информационных технологий.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Семинар 1. Основные понятия алгоритмизации

Семинар 2. Типы и структуры данных
Семинар 3. Численные методы
Семинар 4. Сортировки и алгоритмы поиска: практические задачи
Семинар 5. Базовые принципы рекурсивных алгоритмов
Семинар 6. Построение и использование деревьев решений
Семинар 7. Сетевые алгоритмы
Семинар 8-9. Распределённые алгоритмы

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

Не предусмотрены

4.2 Основная литература

1. Трофимов, В. В. Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская. – 4-е изд. – Москва: Издательство Юрайт, 2025. – 108 с. – (Высшее образование). – ISBN 978-5-534-20430-8. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/562040.

2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений: учебник для вузов / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2026. — 91 с. — (Высшее образование). — ISBN 978-5-534-21288-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585824.

3. Аверина, Т. А. Численные методы. Алгоритмы моделирования систем со случайной структурой: учебник для вузов / Т. А. Аверина. – Москва: Издательство Юрайт, 2025. – 156 с. – (Высшее образование). – ISBN 978-5-534-07204-4. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/564783.

4.3 Дополнительная литература

1. Дискретная математика: прикладные задачи и сложность алгоритмов: учебник и практикум для вузов / А. Е. Андреев, А. А. Болотов, К. В. Коляда, А. Б. Фролов. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2026. — 317 с. — (Высшее образование). — ISBN 978-5-534-04246-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585404.

2. Аверина, Т. А. Численные методы. Верификация алгоритмов решения систем со случайной структурой: учебник для вузов / Т. А. Аверина. – Москва: Издательство Юрайт, 2025. – 179 с. – (Высшее образование). – ISBN 978-5-534-07205-1. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/564782.

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, 2026. — 207 с. — (Высшее образование). — ISBN 978-5-534-12274-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/582837.

4.4 Электронные образовательные ресурсы

Не предусмотрено

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 «Программирование на Python» 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-12	Способен организовывать разработку и применение алгоритмов и современных цифровых программных методов расчетов и проектирования отдельных устройств и подсистем мехатронных и робототехнических систем с использованием стандартных исполнительных и управляющих устройств, средств автоматики, измерительной и вычислительной техники в соответствии с техническим заданием, разрабатывать цифровые алгоритмы и программы управления робототехнических систем

7.1 Методы контроля и оценивания результатов обучения

Перечень оценочных средств по дисциплине «Программирование на Python»

№ 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 пусть достижения результата.
3	Промежуточный	Зачёт	Промежуточная аттестация обучающихся в форме дифференцированного 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 (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено», «не зачтено». Во время проведения дифференцированного 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 «Программирование на Python»

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
	Не зачтено	Зачтено		
знать: автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python	Обучающийся демонстрирует неполное соответствие следующих знаний: автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python 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: автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний: автоматизированного проектирования, методы оптимизации производственных алгоритмов и гибких систем, специализированные библиотеки Python. Свободно оперирует приобретенными знаниями.
уметь: разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы	Обучающийся не умеет или в недостаточной степени умеет: разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы	Обучающийся демонстрирует неполное соответствие следующих умений: разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы. Допускаются значительные ошибки,	Обучающийся демонстрирует частичное соответствие следующих умений: разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы. Умения освоены, но допускаются незначительные ошибки, неточности,	Обучающийся демонстрирует полное соответствие следующих умений: разрабатывать алгоритмы автоматизированного проектирования на Python, генерировать и адаптировать управляющие программы. Свободно оперирует приобретенными

		проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	умениями, применяет их в ситуациях повышенной сложности.
владеть: навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода	Обучающийся не владеет или в недостаточной степени владеет навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода логики.	Обучающийся в недостаточной степени владеет: навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет: навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода. Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет: навыками программирования на Python для автоматизации проектирования, методами обработки данных и моделирования производственных процессов, инструментами анализа и оптимизации кода. 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 показателям, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности, не испытывает затруднений 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 полное отсутствие или недостаточное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент не может оперировать 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 «Практическая работа 1. Графическое представление алгоритмов»

1. Выберите задачу для разработки алгоритма, согласно варианту.
2. Сначала опишите выбранный алгоритм в текстовой форме. Укажите основные шаги, условия и действия, которые выполняются в ходе алгоритма.
3. Затем создайте графическое представление алгоритма с использованием блок-схемы или диаграммы потока данных. Разделите алгоритм на логические блоки, представляющие различные этапы выполнения.
4. Каждый блок должен содержать подробное описание действий, которые выполняются на данном этапе. Для условных операторов (if-else) укажите условия перехода между блоками.
5. При создании графического представления учтите четкость и логичность визуализации. Используйте подходящие символы и стрелки для обозначения потока управления и данных.
6. Приведите примеры входных данных и соответствующих выходных результатов для вашего алгоритма.
7. Предоставьте созданный графический алгоритм в виде файла или рисунка, который можно представить на экране компьютера или напечатать.

Типовое задание «Практическая работа 2. Численные методы в решении учебных и научных задач»

1. Выберите одну из следующих задач для решения с использованием численных методов:
   - Решение дифференциального уравнения первого или второго порядка.
   - Нахождение приближенного значения определенного интеграла.
   - Решение системы линейных алгебраических уравнений.
   - Поиск корней нелинейного уравнения.
   - Аппроксимация функции методом наименьших квадратов.
2. Опишите выбранную задачу, укажите её математическое описание и условия.

3. Примените один или несколько подходящих численных методов для решения задачи. Обоснуйте выбор методов.
4. Опишите алгоритм решения задачи в виде блок-схема, оформленной согласно требованиям
5. Сравните результаты, полученные численными методами, с аналитическими или другими известными решениями (если такие имеются).
6. Оформите отчет

Типовое задание «Практическая работа 3. Сортировки: практические задачи»

1. Выберите следующие практические задачи для решения с использованием методов сортировки:
   - Сортировка списка студентов по их среднему баллу.
   - Сортировка списка товаров по цене или количеству на складе.
   - Сортировка массива точек координат по их расстоянию до начала координат.
   - Сортировка списка задач по их срокам выполнения.
2. Опишите выбранные задачи и укажите критерии сортировки (например, по возрастанию/убыванию, по определенному полю и т.д.).
3. Реализуйте несколько различных методов сортировки (например, сортировка пузырьком, сортировка выбором, быстрая сортировка и т.д.) в виде программного кода на выбранном вами языке программирования.
4. Примените реализованные методы сортировки к задачам из пункта 1. Протестируйте их на разных наборах данных, включая случаи с разным объемом данных и разным распределением значений.
5. Сравните эффективность различных методов сортировки в различных сценариях. Проанализируйте время выполнения и количество сравнений/перестановок.
6. Оформите отчет

Типовое задание «Практическая работа 4. Работа с хэш-таблицами»

1. Изучите основные концепции и принципы работы хэш-таблиц. Ознакомьтесь с методами разрешения коллизий (например, метод цепочек, метод открытой адресации).
2. Выберите следующие практические задачи для решения с использованием хэш-таблиц:
   - Создание программы для учета и поиска книг в библиотеке по их авторам и названиям.
   - Управление контактами в адресной книге с возможностью быстрого поиска по имени или номеру телефона.
   - Решение задачи о поиске анаграмм (слов, состоящих из одних и тех же букв) в большом тексте.
3. Реализуйте хэш-таблицу в виде блок-схемы алгоритма (или программного кода на выбранном вами языке программирования). Обеспечьте функции добавления элементов, поиска и удаления.
4. Примените созданную хэш-таблицу к задачам из пункта 2. Протестируйте ее на различных сценариях вставки, поиска и удаления элементов.
5. Проанализируйте эффективность хэш-таблицы в решении выбранных задач. Сравните время выполнения операций с использованием хэш-таблицы и других возможных методов.
6. Оформите отчет

Типовое задание «Практическая работа 5. Рекурсивные алгоритмы в решении учебных и научных задач»

1. Выберите одну из следующих задач для решения с использованием рекурсивных алгоритмов:
    - Вычисление факториала натурального числа.
    - Поиск наибольшего общего делителя двух чисел.
    - Вычисление чисел Фибоначчи.
    - Поиск всех перестановок элементов в массиве.
    - Решение задачи о Ханойской башне.
2. Опишите выбранную задачу и укажите, какие рекурсивные вызовы требуются для её решения.
3. Реализуйте рекурсивные алгоритмы для выбранной задачи в виде программного кода на выбранном вами языке программирования.
4. Примените реализованные рекурсивные алгоритмы к задаче из пункта 1. Протестируйте их на разных наборах данных, включая различные входные параметры.
5. Проанализируйте структуру рекурсии в ваших алгоритмах. Оцените использование памяти и время выполнения. Рассмотрите возможность оптимизации рекурсивных вызовов.
6. Оформите отчет

Типовое задание «Практическая работа 6. Построение деревьев решений и алгоритмы на графах»

1. Выберите одну из следующих задач для решения с использованием деревьев решений или алгоритмов на графах:
    - Построение алгоритма выбора оптимального пути для доставки товаров между несколькими точками.
    - Разработка алгоритма выбора оптимальной стратегии в игре с нулевой суммой (например, крестики-нолики).
    - Построение дерева решений для классификации объектов на основе различных признаков.
    - Решение задачи о нахождении кратчайшего пути в графе с использованием алгоритма Дейкстры или алгоритма Флойда-Уоршелла.
    - Разработка алгоритма для определения наличия цикла в ориентированном графе.
2. Опишите выбранную задачу и определите основные шаги, которые необходимо выполнить для её решения.
3. Реализуйте построение дерева решений или алгоритм на графе в виде программного кода на выбранном вами языке программирования.
4. Примените реализованный алгоритм к задаче из пункта 1. Протестируйте его на различных сценариях входных данных.
5. Проанализируйте результаты работы алгоритма. Оцените его эффективность, сложность и точность решения задачи.
6. Оформите отчет

Типовое задание «Практическая работа 7. Алгоритм поиск кратчайшего пути, алгоритмы обхода графов»

1. Выберите одну из следующих задач для решения с использованием алгоритмов поиска кратчайшего пути и обхода графов:
    - Нахождение кратчайшего пути между двумя вершинами в неориентированном взвешенном графе.

- Решение задачи о нахождении кратчайшего пути между городами на дорожной карте.
- Обход графа в глубину или в ширину с целью поиска определенной информации (например, поиск связных компонент или пути определенной длины).
- Разработка алгоритма для нахождения остовного дерева минимального веса во взвешенном графе.

2. Опишите выбранную задачу и укажите, какие алгоритмы поиска кратчайшего пути и обхода графа могут быть применены для её решения.
3. Реализуйте выбранные алгоритмы поиска кратчайшего пути и обхода графа в виде программного кода на выбранном вами языке программирования.
4. Примените реализованные алгоритмы к задаче из пункта 1. Протестируйте их на различных наборах данных, включая разные структуры графов.
5. Проанализируйте результаты работы алгоритмов. Оцените их эффективность, точность и применимость к разным сценариям.
6. Оформите отчет

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.

Типовое задание «Контрольная работа №4»

Темы:

1. Алгоритмы на графах: обход графа в ширину, топологическая сортировка.
2. Применение алгоритмов в реальных задачах: примеры использования.

Задания:

1. Реализуйте алгоритм обхода графа в ширину (BFS) для поиска кратчайшего пути между вершинами.
2. Приведите пример использования алгоритма Дейкстры для поиска кратчайшего пути в графе, представляющем дорожную сеть.
3. Расскажите о применении алгоритмов на графах в реальных задачах, например, в планировании транспортных маршрутов или в обработке данных.

7.3.2 Промежуточная аттестация

Вопросы к зачету

1. Что такое алгоритм? Приведите определение и основные характеристики.	ОПК-12
2. В чем заключается процесс алгоритмизации? Почему он важен при решении задач?	ОПК-12
3. Какие основные этапы включает процесс разработки алгоритма?	ОПК-12
4. Какие виды представлений алгоритмов вы знаете? Приведите примеры каждого вида.	ОПК-12
5. Что такое структура данных? Какие основные типы данных используются при алгоритмизации?	ОПК-12
6. Опишите понятие "массив" и объясните, как происходит доступ к элементам массива.	ОПК-12
7. В чем разница между линейными и связанными списками? Приведите примеры использования каждой структуры.	ОПК-12
8. Какие типы структур данных обеспечивают быстрый поиск и вставку элементов?	ОПК-12
9. Что представляют собой численные методы? Какие задачи они помогают решать?	ОПК-12
10. Объясните принцип работы метода Ньютона для нахождения корней функции.	ОПК-12
11. Что такое аппроксимация? Приведите примеры областей, где применяются численные методы аппроксимации.	ОПК-12
12. Какие численные методы используются для решения дифференциальных уравнений?	ОПК-12
13. Объясните, как работает алгоритм сортировки пузырьком. Какова его сложность?	ОПК-12
14. Какой алгоритм поиска подходит для упорядоченного массива? Объясните его принцип работы.	ОПК-12
15. Решите задачу: дан массив чисел. Напишите программу для сортировки этого массива методом выбора.	ОПК-12
16. Примените алгоритм бинарного поиска для нахождения индекса элемента в упорядоченном массиве.	ОПК-12

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