

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шаратович

Должность: Ректор

Дата подписания: 30.01.2026 15:34:40

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f9fa4304cc

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Грозненский государственный нефтяной технический университет
имени академика М.Д. Миллионщикова**



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.06 «Основы алгоритмизации и программирования»

Специальность

09.02.11 Разработка и управление программным обеспечением

Квалификация

Программист

Грозный – 2026 г.

СОДЕРЖАНИЕ ПРОГРАММЫ

1. **Общая характеристика РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....**Ошибка!
Закладка не определена.
 - 1.1. *Цель и место дисциплины в структуре образовательной программы Ошибка!*
Закладка не определена.
 - 1.2. *Планируемые результаты освоения дисциплиныОшибка! Закладка не определена.*
2. **Структура и содержание ДИСЦИПЛИНЫ.....** Ошибка! Закладка не определена.
 - 2.1. *Трудоемкость освоения дисциплиныОшибка! Закладка не определена.*
 - 2.2. *Содержание дисциплиныОшибка! Закладка не определена.*
3. **Условия реализации ДИСЦИПЛИНЫ.....** Ошибка! Закладка не определена.
 - 3.1. *Материально-техническое обеспечениеОшибка! Закладка не определена.*
 - 3.2. *Учебно-методическое обеспечениеОшибка! Закладка не определена.*
4. **Контроль и оценка результатов освоения ДИСЦИПЛИНЫ....** Ошибка! Закладка не определена.

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

«Основы алгоритмизации и программирования» (наименование дисциплины)

1.1. Цель и место дисциплины в структуре образовательной программы

Цель дисциплины «ОП.06 Основы алгоритмизации и программирования»: изучение основных принципов объектно-ориентированного программирования на примере алгоритмического языка

Дисциплина «ОП.06 Основы алгоритмизации и программирования» включена в обязательную часть общепрофессионального цикла образовательной программы.

1.2. Планируемые результаты освоения дисциплины

Результаты освоения дисциплины соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника (п. 4.3 ОПОП-П).

В результате освоения дисциплины обучающийся должен:

Код ОК	Уметь	Знать	Владеть
ОК.01	распознавать задачу или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы выявлять и эффективно искать информацию, необходимую для решения задачи или проблемы владеть актуальными методами работы в профессиональной и смежных сферах оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	актуальный профессиональный и социальный контекст, в котором приходится работать и жить структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте методы работы в профессиональной и смежных сферах порядок оценки результатов решения задач профессиональной деятельности	

ОК.02	определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска оценивать практическую значимость результатов поиска применять средства информационных технологий для решения профессиональных задач использовать современное программное обеспечение в профессиональной деятельности использовать различные цифровые средства для решения профессиональных задач	номенклатура информационных источников, применяемых в профессиональной деятельности приемы структурирования информации формат оформления результатов поиска информации современные средства и устройства информатизации, порядок их применения и программное обеспечение в профессиональной деятельности, в том числе цифровые средства	
ОК.05	грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке проявлять толерантность в рабочем коллективе	правила оформления документов правила построения устных сообщений	
ОК.09	понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы участвовать в диалогах на знакомые общие и профессиональные темы строить простые высказывания о себе и о своей профессиональной деятельности кратко обосновывать и объяснять свои действия (текущие и планируемые) писать простые связные сообщения на знакомые или интересующие профессиональные темы	правила построения простых и сложных предложений на профессиональные темы основные общеупотребительные глаголы (бытовая и профессиональная лексика) лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности особенности произношения правила чтения текстов профессиональной направленности	

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Трудоемкость освоения дисциплины

Наименование составных частей дисциплины	Объем в часах	В т.ч. в форме практ. подготовки
Учебные занятия	28	14
<i>Курсовая работа (проект)</i>	-	-
Самостоятельная работа	154	-
Промежуточная аттестация в <i>форме (зачет)</i>	10	-
Всего	192	14

2.2. Содержание дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем, акад. ч / в том числе в форме практической подготовки, акад. ч	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1. Основы алгоритмизации		16/ 8	
Тема 1.1. Понятие алгоритма и его свойства	Содержание	4	ОК 01 ОК 02 ОК 05 ОК 09
	1. Понятие алгоритма. Свойства и виды алгоритмов	2	
	В том числе практических занятий	2	
	1. Составление и оформление блок-схем простых алгоритмов.	2	
	Самостоятельная работа обучающихся	30	
	1. Свойства алгоритмов: разбор примеров на дискретность, результативность, конечность, определенность, массовость.		
	2. Способы записи алгоритмов: сравнение естественного языка, псевдокода, блок-схем, структурограмм.		
	3. Разработка блок-схем по ГОСТ 19.002–80 и ГОСТ 19.003–80: примеры для линейных, разветвляющихся и циклических алгоритмов.		
	4. Анализ сложности алгоритмов: О-нотация, сравнение алгоритмов сортировки по времени выполнения.		
	5. Метод частных целей и метод подъема в разработке алгоритмов: практические примеры.		
	6. Линейные алгоритмы: расчёт геометрических величин (площадь, периметр, объём).		
	7. Условные операторы: реализация ветвлений (if-else, switch) для решения задач классификации.		
	8. Циклы с предусловием (while) и постусловием (do-while): задачи на накопление, поиск.		
	9. Циклы с параметром (for): обработка последовательностей, вычисление сумм/произведений.		
	10. Вложенные циклы: вывод таблиц умножения, обработка матриц.		
	11. Простые типы данных: преобразование между типами, приоритеты операций.		
	12. Строковый тип: анализ и модификация строк (поиск подстроки, замена, реверсирование).		
	13. Массивы (одномерные и двумерные): ввод/вывод, поиск минимума/максимума, сортировка (пузырьком, выбором).		
	14. Структурные типы данных: создание и использование структур (struct) для хранения записей.		
	15. Динамические структуры: основы работы со списками (добавление, удаление, обход).		

Тема 1.2. Методы разработки алгоритмов	Содержание	4	ОК 01 ОК 02 ОК 05 ОК 09
	Основные методы и этапы проектирования алгоритмов: постановка задачи, математическое описание – математическая модель. Нисходящее, модульное и восходящее проектирование.	2	
	В том числе практических занятий	2	
	1. Проектирование и оформление алгоритмов сортировки.	2	
	Самостоятельная работа обучающихся 1. Создание пользовательских функций: передача параметров по значению и по ссылке. 2. Рекурсия: вычисление факториала, чисел Фибоначчи, обход деревьев. 3. Стандартные библиотеки: использование математических функций (sin, cos, sqrt) и генераторов с случайных чисел. 4. Модульное программирование: разделение кода на файлы (заголовки и реализации). 5. Отладка функций: поиск и исправление ошибок (синтаксических, логических). 6. Сортировка: сравнение методов (пузырьковая, вставками, быстрая) по скорости и памяти. 7. Поиск: линейный и бинарный поиск, анализ эффективности. 8. Работа с матрицами: сложение, умножение, транспонирование. 9. Алгоритмы на графах: поиск в ширину (BFS), поиск в глубину (DFS) — реализация на массивах/списках. 10. Жадные алгоритмы: задача о размене монет, планирование задач. 11. Последовательный доступ к файлам: чтение/запись текстовых данных. 12. Двоичные файлы: сохранение и загрузка структур данных. 13. Обработка CSV-файлов: анализ данных, фильтрация, экспорт результатов. 14. Консольный ввод-вывод: валидация данных, обработка ошибок ввода. 15. Логирование: запись событий программы в файл для отладки.	30	
Тема 1.3 Базовые понятия программирова ния	Содержание	4	ОК 01 ОК 02 ОК 05 ОК 09
	1. Классификация и генеалогия актуальных языков программирования. Понятие системы программирования.	2	
	В том числе практических занятий	2	
	1. Изучение инструментария среды программирования	2	
	Самостоятельная работа обучающихся 1. Классы и объекты: проектирование простых классов (например, «Точка», «Студент»). 2. Инкапсуляция: использование приватных полей и публичных методов. 3. Наследование: создание иерархии классов (например, «Транспорт» → «Автомобиль», «Велосипед»). 4. Полиморфизм: переопределение методов, виртуальные функции. 5. Конструкторы и деструкторы: управление ресурсами, инициализация объектов. 6. Шифр Цезаря: реализация кодирования/декодирования текста.	30	

	7. Хэш-функции: расчёт простых хэшей (сумма кодов символов), сравнение коллизий. 8. Генерация паролей: алгоритм создания сложных комбинаций. 9. Проверка валидности данных: регулярные выражения для email, телефона. 10. Основы криптографии: симметричное шифрование (XOR), асимметричное (упрощённый RSA). 11. Профилирование: измерение времени выполнения участков кода. 12. Оптимизация циклов: уменьшение числа итераций, вынос вычислений за цикл. 13. Тестирование: написание юнит-тестов для функций (проверка граничных случаев). 14. Отладка: использование отладчика (gdb, IDE-инструменты), анализ стека вызовов. 15. Анализ утечек памяти: поиск не освобождённых ресурсов в C/C++.		
Тема 1.4. Программная реализация алгоритмов	Содержание	4	OK 01 OK 02 OK 05 OK 09
	1. Методы реализации типовых алгоритмов. Переменные: определение, правила именования. Типы данных: значимые и ссылочные. Объявление и инициализация переменных. Область действия и время существования переменных. Константы: определение, виды и правила записи в программе.	2	
	В том числе практических занятий	2	
	Реализация простых циклических алгоритмов.	2	
	Самостоятельная работа обучающихся 1. Калькулятор выражений: парсинг и вычисление арифметических выражений (с учётом приоритетов операций). 2. Менеджер контактов: хранение, поиск, редактирование записей в файле. 3. Игра «Угадай число»: реализация с подсчётом попыток и подсказками. 4. Анализатор текста: подсчёт слов, букв, частотных символов в файле. 5. Генератор отчётов: формирование PDF/HTML-отчётов на основе данных из CSV. 6. Система инвентаризации: учёт товаров с фильтрацией по категориям и сортировкой. 7. Визуализация данных: построение графиков (столбчатых, круговых) на основе числовых данных. 8. Чат-бот для ответов на вопросы: обработка команд, хранение истории. 9. Симулятор очереди: реализация FIFO-структуры для обслуживания клиентов. 10. Программа для работы с комплексными числами: арифметические операции, вывод в алгебраической форме. 11. Решение задач на динамическое программирование: задача о рюкзаке, вычисление чисел Фибоначчи с мемоизацией. 12. Реализация алгоритмов поиска кратчайшего пути: Дейкстры, Беллмана-Форда, A*. 13. Обработка строк: алгоритм Кнута-Морриса-Пратта (KMP) для поиска подстроки. 14. Работа с деревьями: построение бинарного дерева поиска, обход (in-order, pre-order, post-order). 15. Графовые алгоритмы: топологическая сортировка, нахождение компонент связности.	32	

	16. Обработка исключений в разных языках: try-catch (C++, Java), try-except (Python).		
Раздел 2. Основы объектно-ориентированного программирования		12/6	
Тема 2.1. Основные понятия объектно- ориентированн ого программирова ния	Содержание	12	ОК 01 ОК 02 ОК 05 ОК 09
	1. Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события.	2	
	2. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция.	2	
	3. Общая форма определения класса	2	
	В том числе практических занятий	6	
	1. Подготовка презентации по образовательным ресурсам	2	
	2. Создание простейших классов.	2	
	3. Создание классов, иерархически связанных между собой	2	
	Самостоятельная работа обучающихся	32	
	1. Реализация стека и очереди на массивах и связанных списках.		
	2. Хэш-таблицы: разработка простой реализации с разрешением коллизий (метод цепочек/открытой адресации).		
	3. Двоичные кучи: построение, вставка, удаление, применение в приоритетной очереди.		
4. Деревья отрезков (Segment Tree): построение и запросы на диапазон.			
5. Префиксные деревья (Trie): хранение и поиск строк, автодополнение.			
6. Рекурсивное решение задачи «Ханойская башня»: анализ сложности, визуализация шагов.			
7. Оптимизация рекурсии: переход к итерации, использование мемоизации.			
8. Генерация перестановок и комбинаций рекурсивным методом.			
9. Разбор выражений: рекурсивный спуск для вычисления арифметических выражений.			
10. Фракталы: рекурсивная генерация кривых (например, кривая Коха).			
11. Сравнение производительности сортировок: пузырьковая, быстрая, слияние, пирамидальная.			
12. Сортировка подсчётом и поразрядная сортировка (Radix Sort): условия применимости.			
13. Стабильность алгоритмов сортировки: примеры и влияние на результат.			
14. Параллельная сортировка: реализация Merge Sort с многопоточностью.			
15. Визуализация работы алгоритмов сортировки: графическое отображение шагов.			
16. Разработка надёжного ввода данных: валидация, обработка некорректных значений.			
Промежуточная аттестация		10	
Всего:		192	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ

3.1. Материально-техническое обеспечение

Кабинет «Информатики и информационных технологий», оснащенный в соответствии с приложением 3 ОПОП-П.

Лаборатория «_» оснащенная в соответствии с приложением 3 ОПОП-П.

3.2. Учебно-методическое обеспечение

3.2.1. Основные печатные или электронные издания

1. Дорохова, Т. Ю. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Ю. Дорохова, И. Е. Ильина. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 139 с. — ISBN 978-5-4488-1531-7, 978-5-4497-1718-4. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/122426>

2. Кудинов, Ю. И. Основы алгоритмизации и программирования : учебное пособие для СПО / Ю. И. Кудинов, А. Ю. Келина. — 2-е изд. — Липецк, Саратов : Липецкий государственный технический университет, Профобразование, 2020. — 71 с. — ISBN 978-5-88247-956-4, 978-5-4488-0757-2. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/92834>

3. Чурина, Т. Г. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Г. Чурина, Т. В. Нестеренко. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2020. — 214 с. — ISBN 978-5-4488-0802-9, 978-5-4497-0465-8. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/96017>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результаты обучения	Показатели освоённости компетенций	Методы оценки
Знает: понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; классификация языков программирования; понятие системы программирования; основные элементы языка, структура программы; методы реализации типовых алгоритмов; операторы и операции, управляющие структуры, структуры данных, классы памяти; понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования	Разработан и оформлен алгоритм для решения поставленной задачи и выполнена оценка его сложности; предложенный алгоритм реализован в среде программирования на одном из актуальных языков программирования; код разработанной программы отлажен, оформлен в соответствии со стандартами кодирования и соответствует алгоритму (результат выполнения соответствует эталонному).	Экспертное наблюдение выполнения практических работ и видов работ по практике Диагностика (тестирование, контрольные работы)
Умеет: разрабатывать и анализировать алгоритмы для решения поставленных задач; определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных		

языках программирования; использовать средства проектирования для создания и графического отображения алгоритмов; оформлять код программ в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы		
---	--	--

Разработчик:

Преподаватель ФСПО

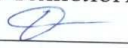


(подпись)

/А.А. Базаева/

Согласовано:

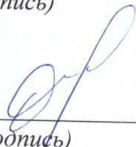
Председатель ПЦК «Информационные технологии»



(подпись)

/И.М. Дубаев/

1-й зам.-зам. декана по УМР ФСПО



(подпись)

/М.И. Дагаев/

Директор ДУМР



/М.А. Магомаева /