

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Дагестанский государственный университет»
Колледж

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.11 Теория алгоритмов и языки программирования

по программе подготовки специалистов среднего звена (ППССЗ) среднего
профессионального образования

<i>Специальность:</i>	10.02.05. Обеспечение информационной безопасности автоматизированных систем.
<i>Обучение:</i>	по программе базовой подготовке
<i>Уровень образования, на базе которого осваивается ППССЗ:</i>	основное общее образование
<i>Квалификация:</i>	техник по защите информации
<i>Форма обучения:</i>	очная

Рабочая программа дисциплины «Теория алгоритмов и языки программирования» разработана на основе требований ФГОС для среднего профессионального образования (СПО) по специальности 10.02.05 «Обеспечение информационной безопасности автоматизированных систем» для реализации основной профессиональной образовательной программы СПО на базе основного общего образования с получением среднего общего образования.

Организация-разработчик: Колледж федерального государственного бюджетного образовательного учреждения высшего образования «Дагестанский государственный университет».

Разработчики:

Магомедова П. Р.- к.ю.н., доцент, зав кафедрой общепрофессиональных дисциплин Колледжа ДГУ.

Хангишиева А. Х. – преподаватель кафедры общепрофессиональных дисциплин Колледжа ДГУ.

Рецензент:

Исмиханов З.Н.- к.э.н., доцент, зав. кафедрой информационных систем и технологий программирования факультета ИИТ ДГУ.

Рабочая программа дисциплины рассмотрена и рекомендована к утверждению на заседании кафедры общепрофессиональных дисциплин Колледжа ДГУ.

Протокол № 7 от « 31 » марта 2022 г.

Зав. кафедры Магомедова П.Р. /Магомедова П.Р.
подпись Фамилия И.О.

Утверждена на заседании учебно-методического совета колледжа ДГУ

Ст. методист Шамсутдинова У.А. /Шамсутдинова У.А.
подпись Фамилия И.О.

Рабочая программа дисциплины согласована с учебно-методическим управлением

«31» 03 2022 г.

Шамсутдинова У.А.
подпись

СОДЕРЖАНИЕ

1. Паспорт программы учебной дисциплины «Теория алгоритмов и языки программирования»
2. Структура и содержание дисциплины
3. Условия реализации дисциплины
4. Контроль и оценка результатов освоения дисциплины

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Теория алгоритмов и языки программирования

1.1. Область применения программы

Рабочая программа дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 10.02.05. Обеспечение информационной безопасности автоматизированных систем для очного обучения студентов, имеющих основное общее образование, по программе базовой подготовки.

Рабочие программы дисциплин, адаптированные для обучения лиц с ограниченными возможностями здоровья, разрабатываются с учетом конкретных ограничений здоровья лиц, зачисленных в колледж, и утверждаются в установленном порядке.

1.2. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Теория алгоритмов и языки программирования» относится к общепрофессиональным дисциплинам профессионального цикла ППСЗ

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины

Содержание программы учебной дисциплины «Теория алгоритмов и языки программирования» направлено на достижение следующих целей:

- изучение и применение на практике основных методов и средств обработки, хранения, передачи и поиска информации с использованием различных программных средств,

Задачи учебной дисциплины «Теория алгоритмов и языки программирования» в профессиональной деятельности:

1 Понимать сущность и социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности

2 Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество

3 Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития

4 Использовать информационно-коммуникационные технологии в профессиональной деятельности

5 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации

6 Ориентироваться в условиях частой смены технологий в профессиональной деятельности

7 Владеть основными методами и средствами разработки программного обеспечения

8 Решать частные технические задачи, возникающие при проведении всех видов плановых и внеплановых контрольных проверок, при аттестации объектов, помещений, программ, алгоритмов.

Освоение содержания учебной дисциплины «Теория алгоритмов и языки программирования» обеспечивает достижение студентами следующих результатов:

Общие компетенции:

ОК-01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК-02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК-03. Планировать и реализовывать собственное профессиональное и личностное развитие

Профессиональные компетенции:

ПК-2.1. Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.

ПК-2.2. Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.

ПК-2.3.Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.

ПК-2.4.Осуществлять обработку, хранение и передачу информации ограниченного доступа.

ПК-2.6.Осуществлять регистрацию основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак.

В результате освоения дисциплины обучающийся должен знать:

- типы данных, базовые конструкции изучаемых языков программирования;
- интегрированные среды изучаемых языков программирования.

В результате освоения дисциплины обучающийся должен уметь:

- использовать языки программирования высокого уровня;
- работать в интегрированной среде программирования.

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	115
Обязательная аудиторная учебная нагрузка (всего)	94
в том числе:	
теоретическое обучение	38
лабораторные работы	-
практические занятия	56
контрольные работы	
курсовой проект	-
консультация	1
экзамен	
дифференцированный зачет	
Самостоятельная работа обучающегося (всего)	20
в том числе:	
самостоятельная работа над курсовой работой	
внеаудиторная самостоятельная работа	
Промежуточная аттестация в форме	дифференцированный зачет

2.2. Тематический план и содержание дисциплины Теория алгоритмов и языки программирования

Наименование разделов и тем	Содержание учебного материала, лекций, лабораторные и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Формы и методы контроля и оценки результатов обучения	
1	2	3	4	
Раздел 1.				
Тема 1.1. Неформальное определение алгоритма. Модели вычислений.	Лекции	6		
	1	Интуитивное представление об алгоритмах	2	
	2	Рекурсивные функции	2	
	3	Функции Аккермана, оператор минимизации; общерекурсивные и частично-рекурсивные функции; тезис Чёрча	2	
	Практические занятия:		12	
	1.	Решение задач и упражнений на тему «Алгоритмы линейной, разветвляющейся и циклической структуры»	4	Устный опрос, Оценка фактического материала
	2.	Программная система «Исполнители»	2	
	3.	Графический вид (язык блок-схем). ГОСТ 19.003-80, ИСО 1028-73. Регламент использования графических примитивов.	2	
	4.	Программирование ветвлений. Пример: найти наименьшее из трех чисел.	2	
	5.	Использование циклов с параметром для обработки массивов	2	
Самостоятельная работа обучающихся: * Выяснение конечной цели и выработка общего подхода к решению задачи. Выясняется, сколько решений имеет задача и имеет ли их вообще. Изучаются общие свойства рассматриваемого физического явления или объекта.		4		
Тема 1.2. Алгоритмы и исполнители	Лекции	2		
	1	Основные алгоритмы обработки информации, алгоритмы над числами; алгоритмы последовательного и бинарного поиска; алгоритмы сортировки; хеш-функции и методы исключения коллизий	2	
	Практические занятия		4	
	Алгоритм в виде блок-схемы		4	Устный опрос, Оценка фактического материала
	Самостоятельная работа обучающихся:* - домашние практические работы на построение алгоритмов решения простейших задач с использованием данных различных типов		2	
Раздел 2.				
Тема 2.1.	Лекции	4		

Алгоритм. Свойства алгоритмов.	1	Алгоритм как определенным образом организованная последовательность действий, за конечное число шагов приводящая к решению задачи.	2	
	2	Свойства: Определенность. Дискретность. Целенаправленность. Конечность. Массовость	2	
	Практические занятия			
	1. Нормальные алгоритмы Маркова		4	Устный опрос, Оценка фактического материала
	2. Машина с неограниченными регистрами		2	
	Самостоятельная работа обучающихся: -домашние практические работы на программирование алгоритмов для консольного приложения.		2	
Раздел 3.				
Тема 3.1. Алгоритмическая конструкция ветвления	Лекции		4	
	1	Ветвление - управляющая структура, организующая выполнение лишь одного из двух указанных действий в зависимости от справедливости некоторого условия.	2	
	2	Рекурсия. Программирование рекурсивных алгоритмов.	2	
	Практические занятия:		12	
	Программирование алгоритмов с использованием процедур и функций.		4	Устный опрос, Оценка фактического материала
	Организация функций.		4	Устный опрос, Оценка фактического материала
	Решение задач и упражнений на тему «Рекурсивные функции»		4	Устный опрос, Оценка фактического материала
	Самостоятельная работа обучающихся: домашние практические работы на создание приложения для отображения случайных чисел		2	
Тема 3.2. Алгоритмическая конструкция цикла	Лекции		4	
	1	Параметр цикла, его начальное и конечное значения и шаг должны быть одного типа.	2	
	2	Основы структурного программирования. Методы структурного программирования.	2	

	Самостоятельная работа обучающихся: – Использование циклов с параметром для обработки массивов – Массив задается именем, типом данных и размерностью - максимально возможное количество элементов в массиве.		2	
Тема 3.3. Сложность алгоритмов	Лекции		4	
	1	Оценка сложности алгоритмов	2	
	2	Оценка сложности алгоритмов сортировки и поиска	2	
	Практические занятия:		8	
	1. Временная и пространственная сложность алгоритма; классы сложности – полиномиальность и эффективность, иерархия классов сложности; алгоритмическая сводимость задач		4	Устный опрос, Оценка фактического материала
	2. Анализ и оценка сложности алгоритмов сортировки; анализ и оценка сложности алгоритмов поиска		4	Устный опрос, Оценка фактического материала
Самостоятельная работа обучающихся: - Оценка экспериментальным способом времени выполнения того же алгоритма.		2		
Раздел 4.				
Тема 4.1. Алгоритмы для файлов	Лекции		2	
	1	Файловые операции, описание, открытие, чтение, запись, управление указателем.	2	
	Практическое занятия:		4	
	Сортировка файлов. Через рабочий массив или файл.		4	
Самостоятельная работа обучающихся: домашние практические работы на создание проекта с использованием компонентов для работы с текстом		2		
Раздел 5. Объектно-ориентированное программирование				
Тема 5.1. Объектно - ориентированное программирование	Лекции		8	
	1	Объект как усовершенствование типа запись, в которой описание свойств и параметры моделируемой сущности дополняются методами - описаниями действий с объектом.	4	
	2	Управление объектами на форме и из программы.	4	
	Практические занятия:		8	
	Программирование простейших классов.		4	Устный опрос, Оценка фактического материала
	Классы ООП: виды, назначение, свойства, методы, события.		4	Устный опрос, Оценка фактического материала
	Самостоятельная работа обучающихся:		2	

	- домашние практические работы на создание проекта с использованием компонентов для работы с текстом		
Тема 5.2. Использование функций в приближенных вычислениях.	Лекции	4	
	1 Алгоритмы вычисления интегралов. Правила написания процедур, параметры входные и выходные.	4	
	Практические занятия:	4	
	Использование процедур и функций для повышения надежности программ	4	
	Самостоятельная работа обучающихся: домашние практические работы на создание проекта с использованием компонентов для работы с текстом	2	
	<i>Консультации</i>	1	
Примерная тематика курсовой работы(проекта) <i>если предусмотрены</i>			
<i>Консультации</i>		1	
<i>Лекционные занятия</i>		38	
<i>Практические занятия</i>		56	
<i>Самостоятельная работа</i>		20	
<i>Итого</i>		115	
<i>Форма контроля</i>		Дифференцированный зачет	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебного кабинета.

Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно-наглядных пособий;

Технические средства обучения:

- компьютер по количеству обучающихся с лицензионным программным обеспечением и с доступом к сети Интернет.
- проектор;
- интерактивная доска;

Программное обеспечение:

- Интегрированная среда программирования (Python, Microsoft Visual Studio, и т.п.)
- Операционные системы Windows 7, SuseLinux 10.
- Программы настройки и оптимизации операционной системы: NortonUtilities, SystemUtilities, Sandra, SystemMechanic.
- Архиваторы: WinRar, WinZip, ZipMagic, WinAce.
- Антивирусные средства: Drweb, Avp.
- Программы для работы с изображением: ACDSee, AcrobatReader.
- Программы для работы с Internet и электронной почтой: EtypeDialer, GetRight, TheBat!, Ace FTP, Opera, ICQ. 10
- Программы-оболочки: FAR manager 2.0, Volkov Commander.
- MS Office 20016/2021.
- СУБД ACCESS.
- СерверБД MS SQL.

- Программы для тестирования аппаратных устройств ПЭВМ.
- Текстовые процессоры / редакторы: Word, LaTeX.
- Программы для создания компьютерных презентаций MS PowerPoint

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

Основная литература:

1. *Крупский, В. Н.* Теория алгоритмов. Введение в сложность вычислений : учебное пособие для вузов / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 117 с. — (Высшее образование). — ISBN 978-5-534-04817-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492937>
2. *Трофимов, В. В.* Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2022. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493261>

Дополнительная литература:

1. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с.
2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2022. — 322 с.
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 210 с.
4. Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с

Электронные ресурсы:

1. Официальный сайт Python [Электронный ресурс]:- Режим доступа: <https://www.python.org>
2. Руководство по языку программирования Python – Metanit [Электронный ресурс]:- Режим доступа: <https://metanit.com> > python > tutorial
3. Самоучитель Python | Python 3 для начинающих и чайников [Электронный ресурс]:- Режим доступа: <https://pythonworld.ru> > samouchitel-python
4. Учебник Python 3.1 [Электронный ресурс]:- Режим доступа: <https://ru.wikibooks.org>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля результатов обучения
<p>В результате освоения дисциплины обучающийся должен знать:</p> <ul style="list-style-type: none">- интегрированные среды изучаемых языков программирования- обнаруживать и устранять неисправности системы защиты информации автоматизированной системы согласно документации (06.033 А/01.5)- типы данных, базовые конструкции изучаемых языков программирования;	<p>- Оценка в рамках текущего контроля результатов деятельности обучающихся при выполнении:</p> <ul style="list-style-type: none">- проверка конспектов лекций;- тестирование;- самостоятельная работа;- выполнение индивидуальных практических заданий;- индивидуальный и фронтальный опрос в ходе аудиторных занятий;- защита практической работы
<p>В результате освоения дисциплины обучающийся должен уметь:</p>	
<ul style="list-style-type: none">- использовать языки программирования высокого уровня;- работать в интегрированной системе программирования;	<p>- Оценка в рамках текущего контроля результатов деятельности обучающихся при выполнении:</p> <ul style="list-style-type: none">- проверка конспектов лекций;- тестирование;- самостоятельная работа;- выполнение индивидуальных Практических/лабораторных заданий;- индивидуальный и фронтальный опрос в ходе аудиторных занятий;- защита практической работы

Перечень зачетно-экзаменационных вопросов

1. Основные понятия алгоритмизации.
2. Способы описания алгоритмов.
3. Способ описания алгоритмов записи на естественном языке.
4. Свойства алгоритмов.
5. Линейные алгоритмы.
6. Алгоритмы с ветвлениями.
7. Циклические алгоритмы.
8. Базовые конструкции структурного программирования.
9. Цикл.
10. Цикл с предусловием.
11. Цикл с постусловием
12. Компилятор
13. Интерпретатор
14. Байт-код.
15. Основы языков C / C ++: синтаксис языка, директивы препроцессора,
16. консольное приложение.
17. Коллизия имён и использование пространства имен.
18. Классификация типов данных в C++. Арифметические типы данных.
19. Преобразование типов в выражениях.
20. Переменные, константы, операторы и выражения. Приоритеты операций и
21. операторов в C ++.
22. Потоки и файлы; ввод, вывод, управляющие последовательности.
23. Библиотечные функции.
24. Функции: прототип, определение, реализация. Функция main (). Структура
25. программы.
26. Логические операции. Операторы управления C++-программой.
27. Составной тип – строки. Методы ввода строк.
28. Составные типы – структуры, объединения, перечисления и определяемые
29. пользователем типы.
30. Виды циклов в C++.
31. Указатели и свободная память. Работа с памятью с помощью new и delete .
32. Указатели и строки.
33. Массивы. Одномерные и многомерные массивы. Динамические массивы.
34. Статические и динамические массивы структур.
35. Ссылки. Передача массивов как параметров.
36. Функции пользователя в C++. Передача параметров в функции по значению
37. и по ссылке.
38. Функции и строки. Функции сравнения, копирования строк.
39. Функции и структуры. Передача параметров (типа «структура») в функции.
40. Объекты и классы. Описание класса.
41. Конструкторы, их виды. Деструкторы.
42. Inline -функции, создание inline -функций внутри класса.
43. Статические и динамические классы.
44. Область видимости и классы памяти (auto , register , extern , static).
45. Указатели на объекты. Передача объектов функциям. Объекты в качестве
46. возвращаемых функциями значений.
47. Массивы объектов, организация работы с ними.
48. Полиморфизм. Переопределение функций. Функции с аргументами по
49. умолчанию.
50. Конструктор копий.
51. Виртуальные функции. Абстрактные классы. Раннее и позднее связывание.
52. Указатель this .

53. Традиционные типы конструкторов. Списки инициализаторов.
54. Перегрузка функций. Неоднозначность при перегрузке функций.
55. Перегрузка операторов.
56. Реализация операторной функции =.
57. Реализация операторной функции сравнения ==.
58. Наследование. Виды наследования. Дружественные функции.
59. Перегрузка операторов ввода/вывода.
60. Вывод в файл.
61. Ввод в файл.
62. Шаблоны функций, пример.
63. Шаблоны классов, пример.
64. Обработка исключительных ситуаций.
65. Обзор языков: объектно-ориентированное программирование - инкапсуляция,
66. полиморфизм, наследование.