

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Карпова Елизавета Александровна

Должность: директор

Дата подписания: 24.07.2026 16:53:53

Уникальный программный ключ:

ad9053b6a9e639199a21a41d1a80dd3f5c40650966aaf85dff11a7fd7d02ebad



СОЦИАЛЬНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Информатика и программирование

рабочая программа дисциплины (модуля)

Закреплена за кафедрой **Цикловая комиссия по информатике и информационной безопасности**

Учебный план **09.02.01 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И КОМПЛЕКСЫ**

Учебный год начала подготовки **2026-2027**

Квалификация **специалист по компьютерным системам**

Форма обучения **очная**

Часов по учебному плану	109	Виды контроля в семестрах: экзамены 3
в том числе:		
аудиторные занятия	34	
самостоятельная работа	75	

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		Итого	
Неделя	17			
Вид занятий	уп	рп	уп	рп
Лекции	17	17	17	17
Практические	17	17	17	17
Итого ауд.	34	34	34	34
Контактная работа	34	34	34	34
Сам. работа	75	75	75	75
Итого	109	109	109	109

Рабочая программа дисциплины

Информатика и программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности
09.02.01 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И КОМПЛЕКСЫ (приказ Минобрнауки России от 25.05.2022 г. № 362)

составлена на основании учебного плана:

09.02.01 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И КОМПЛЕКСЫ

утвержденного на заседании Педагогического Совета АНО ПО "СТК" 26.02.2026 протокол № 5.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	-Создать необходимую основу для использования современных средств вычислительной техники и прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин;
1.2	-Освоение предусмотренного программой теоретического материала и приобретение практических навыков использования информационных систем и технологий на базе современных ПК.
1.3	

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:		ОП
2.1	Требования к предварительной подготовке обучающегося:	
2.1.1	Математика	
2.1.2	Теория систем и системный анализ	
2.1.3	Методы принятия управленческих решений	
2.1.4	Студент в среде e-learning	
2.1.5	Философия	
2.1.6	Дискретная математика	
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:	
2.2.1	WEB - программирование	
2.2.2	Информационно-поисковые системы и машины	
2.2.3	Информационные системы в экономической сфере	
2.2.4	Объектно-ориентированное программирование	
2.2.5	Технологии программирования	
2.2.6	3d-моделирование	
2.2.7	Мультимедиа технологии и системы	
2.2.8	Производственная практика (технологическая (проектно-технологическая) практика)	
2.2.9	Производственная практика (эксплуатационная)	
2.2.10	Распределенные информационные ресурсы	
2.2.11	Электронные библиотеки и архивы	
2.2.12	Языки программирования	
2.2.13	Интеллектуальные информационные системы в экономике	
2.2.14	Надежность информационных систем	
2.2.15	Сетевое программирование	
2.2.16	Управление качеством в информационных системах	
2.2.17	Облачные ресурсы и технологии	
2.2.18	Разработка прикладных программных приложений	
2.2.19	Системная архитектура информационных систем	
2.2.20	Управление облачными информационными ресурсами	
2.2.21	Управление проектами информационных систем	
2.2.22	Применение нейронных сетей в информационной сфере	
2.2.23	Принципы построения нейрокомпьютеров	
2.2.24	Проектирование экономических информационных систем	
2.2.25	Производственная практика (преддипломная практика)	
2.2.26	Выполнение и защита выпускной квалификационной работы	
2.2.27	Подготовка к сдаче и сдача государственного экзамена	
2.2.28	Программная инженерия	
2.2.29	Проектирование управляющих программ компьютерных систем и комплексов	
2.2.30	Проектирование цифровых систем	
2.2.31	Освоение рабочей профессии "Оператор электронно-вычислительных и вычислительных машин"	

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОК 01.: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ПК 2.1.: Проектировать, разрабатывать и отлаживать программный код модулей управляющих программ.

ПК 2.2.: Владеть методами командной разработки программных продуктов.

ОК 04.: Эффективно взаимодействовать и работать в коллективе и команде;

В результате освоения дисциплины обучающийся должен

3.1	Знать:
3.1.1	-Наиболее широко используемые классы информационных моделей и основные математические методы получения, хранения, обработки, передачи и использования информации;
3.1.2	-Современные технические и программные средства взаимодействия с компьютером, современные технологии разработки алгоритмов и программ, методы тестирования, отладки и решения задач, средства и методы машинной графики, методику объектно-ориентированного программирования.
3.2	Уметь:
3.2.1	-Управлять ПК при работе в автономном режиме;
3.2.2	-Создавать и редактировать текстовые документы с помощью одного из текстовых редакторов;
3.2.3	-Пользоваться электронными таблицами;
3.2.4	-Подготовить задачу для решения на ПК, включая ее математическую постановку, выбор метода решения, описание алгоритма и составление программы;
3.2.5	-Самостоятельно применять компьютеры для решения учебных задач, используя для этого соответствующие инструментальные средства;
3.2.6	-Применять математический аппарат анализа и синтеза информационных систем;
3.2.7	-Применять методы программирования и навыки работы с математическими пакетами для решения практических задач хранения и обработки информации;
3.2.8	-Использовать современные информационные технологии методов сбора, представления, хранения, обработки и передачи информации с использованием компьютеров.
3.3	Владеть:
3.3.1	-Навыками работы на персональных компьютерах в современных операционных средах, использования современных программных средств, работы пользователя и программиста в интегрированных средах, использующих "оконный интерфейс", пользования компьютерной техникой и информационными технологиями; навыками создания, отладки и тестирования программ, представления результатов в удобном для пользователя виде, создания диалоговых и графических программ.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов
	Раздел 1. Информация. Предмет и структура информации. Представление данных. Математические основы информатики. Технические средства реализации информационных процессов.		
1.1	Математические основы информатики; методы и модели оценки количества информации; системы счисления; перевод из одной системы в другую. /Лек/	3	2
1.2	Системы классификации и кодирования информации. /Пр/	3	2
1.3	Математические основы информатики; методы и модели оценки количества информации; системы счисления; перевод из одной системы в другую. /Ср/	3	10
1.4	Общая характеристика процессов сбора, передачи, обработки и хранения информации /Лек/	3	2
1.5	Представление чисел в различных системах счисления. Перевод чисел из одной системы в другую. /Пр/	3	2
1.6	Общее представление об информации. Техническая, биологическая и социальная информация. Кодированная информация. Понятие носителя информации. Место и роль понятия "информация" в курсе информатики. /Лек/	3	2
1.7	Общее представление об информации. Техническая, биологическая и социальная информация. Кодированная информация. Понятие носителя информации. Место и роль понятия "информация" в курсе информатики. /Ср/	3	12
	Раздел 2. Технические и программные средства реализации информационных процессов		
2.1	Основные функциональные части компьютера. Взаимодействие процессора и памяти при выполнении команд и программ. /Лек/	3	2

2.2	Овладение навыками работы с клавиатурой, мышью, монитором /Пр/	3	2
2.3	Программа как последовательность действий компьютера. Понятие о машинном языке и языке Ассемблер. Исходная и объектная программа. Трансляция как процесс преобразования исходного кода в объектный. Революция персональных компьютеров. /Ср/	3	14
2.4	Устройство персонального компьютера. Периферийные устройства /Пр/	3	2
Раздел 3. Языки и системы программирования. Системное ПО.			
3.1	Понятие операционной системы. Диалог пользователей с операционной системой. /Лек/	3	2
3.2	Работа с пользовательской оболочкой операционной системы /Пр/	3	4
3.3	Классификация программного обеспечения. Системное и прикладное программное обеспечение. Файлы и их имена. Каталоги. Путь к файлу. /Лек/	3	2
3.4	Классификация программного обеспечения. Системное и прикладное программное обеспечение. Файлы и их имена. Каталоги. Путь к файлу. /Ср/	3	24
Раздел 4. Прикладное ПО. Операционные системы вычислительных машин. Информационные технологии (ИТ). Программные продукты Microsoft. Основы ИБ.			
4.1	Понятие информационных технологий. Средства представления, хранения и обработки текстовой и числовой информации. Понятие об экономических и правовых аспектах информационных технологий, аксиоматический метод /Лек/	3	5
4.2	Понятие информационных технологий. Средства представления, хранения и обработки текстовой и числовой информации. Понятие об экономических и правовых аспектах информационных технологий, аксиоматический метод /Ср/	3	11
4.3	Знакомство с прикладным программным обеспечением: с текстовым процессором, антивирусными программами и архиваторами /Пр/	3	5
4.4	/Экзамен/	3	4

5. ОЦЕНОЧНЫЕ СРЕДСТВА

5.1. Вопросы для самоконтроля и текущей аттестации

Экзаменационные вопросы по дисциплине «Информатика и программирование» для направления подготовки 09.02.01 включают как теоретические, так и прикладные вопросы, направленные на проверку базовых знаний, понимания основ программирования и владения ключевыми понятиями информатики.

Часть 1. Основные понятия и определения

Что такое информация и какова её роль в современном обществе?
 Перечислите основные единицы измерения количества информации.
 Чем отличается аппаратное обеспечение от программного обеспечения?
 Назовите и охарактеризуйте основные этапы жизненного цикла программного продукта.
 Поясните понятие алгоритма и назовите его свойства.
 Какие существуют способы представления алгоритмов?
 Охарактеризуйте понятие компьютерной сети и перечислите её основные элементы.
 Что такое операционная система и какую роль она играет в работе компьютера?
 Что означает термин «интерфейс» и какими бывают интерфейсы?
 Приведите классификацию компьютеров по назначению и уровню сложности.

Часть 2. Алгоритмизация и основы программирования

Изложите правила записи условных операторов в языке Python.
 Что такое цикл, и какие конструкции циклов используются в языке Си?
 Определите понятия «переменная» и «константа». Чем они отличаются друг от друга?
 Расскажите о принципах работы оператора присваивания.
 Почему важно соблюдать принципы модульности и абстрагирования при разработке программ?
 Чем различаются методы сортировки «пузырек» и «быстрая сортировка»?
 Что такое рекурсия и каково её применение в программировании?
 Объясните разницу между компиляторами и интерпретаторами.
 Перечислите стандартные типы данных в языке программирования JavaScript.
 Какие операции применяются для работы с файлами в Python?

Часть 3. Структуры данных и алгоритмы

Назовите различия между массивом и списком в Python.

Когда целесообразно использовать динамические структуры данных (связные списки)?
 Поясните работу алгоритма двоичного поиска.
 Что такое хэш-таблица и каким образом обеспечивается высокая скорость поиска в ней?
 Какие преимущества имеет дерево поиска перед линейным поиском?
 Перечислите общие характеристики стека и очереди как абстрактных типов данных.
 По какому принципу работают хэши и зачем они нужны?
 Какие проблемы возникают при недостаточной обработке исключительных ситуаций в программах?
 Дайте характеристику двумерному массиву и расскажите, как его инициализировать.
 Что представляет собой очередь приоритетов и как она реализуется?

Часть 4. Принципы проектирования ПО

Что такое парадигма программирования и какие бывают парадигмы?
 Объясните сущность принципа SOLID в объектно-ориентированном программировании.
 Перечислите известные паттерны проектирования и дайте короткую характеристику одному из них.
 Зачем используют рефакторинг в процессе разработки программного обеспечения?
 Что понимается под архитектурой программного обеспечения?
 Что такое CASE-технологии и для чего они предназначены?
 Назовите причины появления ошибок в программе и меры их предотвращения.
 Что подразумевают под совместимостью программного обеспечения?
 В чём состоит суть подхода Agile-разработки программного обеспечения?
 Опишите цели и назначение документации в жизненном цикле программного продукта.

Часть 5. Практические навыки программирования

Реализуйте алгоритм поиска минимального значения среди элементов одномерного массива.
 Разработайте функцию, определяющую простое число.
 Создайте простую игру «Угадай число», реализовав интерактивный диалог с пользователем.
 Представьте сценарий взаимодействия пользователя с системой регистрации аккаунта на сайте. Разработать соответствующий псевдокод.
 Реализуйте класс «Квадрат», содержащий конструктор, геттеры и сеттеры свойств, а также метод расчета площади фигуры.
 Решите задачу перевода десятичного числа в шестнадцатеричную систему счисления.
 Реализуйте программу подсчета количества четных чисел в заданном диапазоне.
 Постройте диаграмму классов для небольшой библиотеки книг с использованием UML.
 Спроектируйте базу данных для небольшого интернет-магазина товаров.
 Разработайте консольное приложение, демонстрирующее обработку файлов CSV и JSON в Python.

Этот перечень экзаменационных вопросов охватывает широкий спектр тематик дисциплины «Информатика и программирование» и позволяет объективно оценить уровень усвоения студентами ключевых компетенций, необходимых для успешной профессиональной деятельности.

5.2. Темы письменных работ (контрольных и курсовых работ, рефератов)

письменные работы не предусмотрены

5.3. Оценочные средства для промежуточной аттестации

I. Вопросы закрытого типа

Выберите один правильный вариант ответа

1. Что такое переменная?
 - a) Ячейка памяти компьютера, используемая для хранения данных определенного типа.
 - b) Команда программы, выполняющая определенную операцию над данными.
 - c) Специальный символ, обозначающий конец строки текста.
 - d) Объект программы, способный хранить инструкции и команды.
2. Какое расширение имеют файлы исходного кода на языке Python?
 - a) .cpp
 - b) .py
 - c) .java
 - d) .html
3. Какой оператор используется для проверки условия в языке C++?

- a) if
- b) while
- c) for
- d) switch

4. Укажите правильное определение массива:

- a) Массив — это набор данных одного типа, доступ к которым осуществляется по индексу.
- b) Массив — это объект, хранящий значение одной конкретной переменной.
- c) Массив — это структура данных, позволяющая одновременно обращаться ко всем элементам.
- d) Массив — это метод сортировки элементов по возрастанию.

5. Какие типы данных относятся к простым типам данных в большинстве языков программирования?

- a) Числовые, строковые, булевские.
- b) Классы, структуры, объекты.
- c) Связанные списки, очереди, стеки.
- d) Деревья, хеш-таблицы, графы.

II. Вопросы открытого типа

Допишите пропущенное слово/термин

- 6. Компиляция — это процесс преобразования _____ программы в машинный код.
- 7. Переменная, объявленная внутри блока { }, называется _____ переменной.
- 8. Операция сравнения двух значений на равенство записывается символом _____.
- 9. Для объявления целочисленной переменной в JavaScript используется ключевое слово _____, хотя оно часто опускается.
- 10. Цикл, повторяющий блок действий определенное количество раз, называется _____ циклом.

III. Задача на выполнение программы

Прочитайте программу и определите её вывод.

Python

```
x = 5
y = x * 2 + 3
print(y)
```

Варианты ответов:

- a) 10
- b) 13
- c) 8
- d) 15

IV. Заполнение пропусков

Заполните пропуски правильными словами или выражениями.

- 11. В языке Python строки заключаются в _____ или двойные кавычки.
- 12. Чтобы создать список чисел от 1 до 10 включительно, можно воспользоваться функцией range() следующим образом: list(range(____, ____)).
- 13. Оператор ветвления if-else применяется для выбора пути исполнения программы в зависимости от результата

выражения.

14. Функция `input()` в Python служит для чтения данных, введенных пользователем, и возвращает строку, которую можно преобразовать в нужный тип данных с помощью функций вроде `int()`, `float()`.

15. Программный код оформляется комментариями, начинающимися с символов `#` в Python или `//` в языках C++, Java.

V. Составление программы

Напишите короткий фрагмент программы на Python для решения следующей задачи:

16. Напишите программу, запрашивающую у пользователя два числа и выводящую сумму и произведение этих чисел.

Пример правильного решения:

Python

```
# Запрашиваем ввод двух чисел
num1 = float(input("Введите первое число: "))
num2 = float(input("Введите второе число: "))

# Вычисляем сумму и произведение
sum_result = num1 + num2
product_result = num1 * num2

# Выводим результаты
print(f"Сумма равна: {sum_result}")
print(f"Произведение равно: {product_result}")
```

Аналитические задания полезны для развития критического мышления, способности анализировать информацию и самостоятельно находить оптимальные подходы к решению проблем. Они помогают студентам глубже разобраться в изучаемых концепциях и применить полученные знания на практике.

Аналитическое задание №1: Анализ эффективности алгоритмов сортировки

Рассмотрите три наиболее известных алгоритма сортировки: пузырьковую сортировку (Bubble Sort), быструю сортировку (Quicksort) и сортировку вставками (Insertion Sort). Проведите сравнительный анализ этих методов, учитывая следующие аспекты:

- Сложность каждого метода в лучшем, среднем и худшем случаях.
- Объем дополнительной памяти, требуемый каждым алгоритмом.
- Примеры ситуаций, когда использование конкретного алгоритма предпочтительнее остальных.

Сделайте выводы относительно применимости каждого из рассмотренных методов в различных практических ситуациях.

Аналитическое задание №2: Проектирование структуры данных для эффективной работы приложения

Предположим, вам поручено разработать структуру данных для хранения контактной информации пользователей (ФИО, адрес электронной почты, телефон), используемых в корпоративной информационной системе крупной организации. Проанализируйте плюсы и минусы различных структур данных (массивы, связанные списки, хэш-таблицы, бинарные деревья поиска) и выберите оптимальный вариант с точки зрения скорости поиска, добавления новых записей и удаления существующих контактов.

Опишите выбранную вами структуру данных и поясните выбор.

Аналитическое задание №3: Оптимизация программного кода

Вам дан следующий простой пример реализации алгоритма нахождения максимального элемента массива целых чисел на языке Python:

Python

```
def find_max(arr):
    max_value = arr[0]
    for i in range(len(arr)):
        if arr[i] > max_value:
```

```
max_value = arr[i]
```

```
return max_value
```

Проанализируйте этот код и предложите возможные улучшения производительности и читаемости. Оцените временную сложность вашего улучшенного варианта и приведите аргументы, почему именно ваш подход лучше оригинального.

■ Аналитическое задание №4: Выбор подходящего инструмента разработки

Ваш руководитель поручил команде разработчиков выбрать подходящий инструмент для автоматизации тестирования веб-приложений. Рассмотрите такие инструменты, как Selenium WebDriver, Cypress.io и Playwright. Проанализируйте сильные стороны и ограничения каждого инструмента и сделайте обоснованный выбор, учитывая масштаб проекта, поддержку современных браузеров и интеграцию с CI/CD системами.

Приведите подробный отчет с аргументированным выбором и рекомендациями по дальнейшему использованию выбранного инструмента.

■ Аналитическое задание №5: Создание отчета по результатам исследования производительности сервера

Представьте, что ваша задача — проанализировать производительность серверного оборудования и предложить рекомендации по улучшению быстродействия и надежности работы информационной системы предприятия. Выполните тестирование нагрузки на сервер, используя доступные инструменты (например, Apache JMeter или LoadRunner), проанализируйте показатели отклика сервера, загрузку CPU и RAM, объемы передаваемого трафика.

Создайте отчет с результатами анализа, включив графики загрузки ресурса, таблицы результатов тестирования и рекомендации по оптимизации инфраструктуры.

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

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год, эл. адрес
Л1.1	Царев Р. Ю., Пупков А. Н., Самарин В. В., Мыльникова Е. В.	Информатика и программирование: учебное пособие: Учебники и учебные пособия для ВУЗов	Красноярск: Сибирский федеральный университет, 2014 http://biblioclub.ru/index.php?page=book&id=364538&sr=1

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год, эл. адрес
Л2.1	Слабнов В. Д.	Программирование на C++: лекции: Учебники и учебные пособия для ВУЗов	Казань: Познание, 2012 http://biblioclub.ru/index.php?page=book&id=364222&sr=1
Л2.2	Хахаев И. А.	Практикум по алгоритмизации и программированию на Python: Курс	Москва: Национальный Открытый Университет «ИНТУИТ», 2016 https://biblioclub.ru/index.php?page=book_red&id=429256&sr=1

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

6.2.1 Перечень программного обеспечения

6.3.1.1	Alt Linux Enterprise Server — отечественный дистрибутив операционной системы Linux, созданный российской компанией Альт Линукс.
6.3.1.2	JetBrains PyCharm Community Edition — среда разработки для Python, созданная российской компанией JetBrains.
6.3.1.3	znanium.ru
6.3.1.4	Notepad++ — бесплатный редактор кода с поддержкой множества языков программирования
6.3.1.5	Sublime Text — известный редактор кода, который популярен среди российских разработчиков
6.3.1.6	LibreOffice — свободный офисный пакет, активно адаптируемый для российского рынка, в частности, проектом LibreOffice Russia.

6.2.2 Перечень информационных справочных систем и ресурсов сети Интернет

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	
7.1	Компьютеры и оборудование:
7.2	
7.3	1. Рабочее место преподавателя: компьютер с монитором и клавиатурой, снабжённый современным оборудованием для демонстрации презентаций и выполнения демонстрационных примеров на занятиях.
7.4	2. Студентские рабочие места: компьютеры, подключённые к локальной сети и Интернету, оснащённые современной периферией (монитор, мышь, клавиатура); рекомендуется наличие USB-портов для подключения внешних устройств (например, флеш-накопителей).
7.5	3. Средства вывода и отображения информации.
7.6	4. Интернет-канал: стабильный доступ к глобальной сети для скачивания учебных пособий, обновления программного обеспечения и использования удалённых сервисов.
7.7	5. Резервные устройства хранения данных: внешние накопители (HDD, SSD), облачные хранилища для резервного копирования учебных материалов и сохранения разработок студентов.
7.8	
7.9	Техническое оснащение аудиторий:
7.10	
7.11	1. Класс-компьютерный зал: помещение оборудованное рабочими местами студентов с персональными компьютерами, соединёнными в единую сеть и объединённую инфраструктуру.
7.12	2. Кабинеты лекций и семинаров: аудитории оборудованы стационарными ПК для ведения занятий преподавателем и предоставления возможностей коллективной работы группам студентов.
7.13	3. Помещения для самостоятельной работы: специализированные зоны с необходимым оборудованием и материалами для индивидуальной и командной работы студентов вне основного расписания занятий.
7.14	
7.15	Особые помещения:
7.16	
7.17	1. Лаборатории: учебные лаборатории оснащены специализированным оборудованием для углубленного изучения конкретных аспектов программирования, архитектуры ЭВМ и информационного моделирования.
7.18	2. Серверная комната: помещение для размещения технических серверов и другого вспомогательного оборудования, необходимого для функционирования внутренней сети учреждения и поддержки специализированных лабораторных работ.
7.19	
7.20	■ Лицензионное программное обеспечение:
7.21	
7.22	Минимальный комплект лицензионного программного обеспечения включает:
7.23	
7.24	- Современные версии ОС семейства Linux или Microsoft Windows.
7.25	- Базовую среду разработки и редактирования кода (IDE).
7.26	- Типичные офисные пакеты (Microsoft Office, OpenOffice, LibreOffice).
7.27	- Специализированные языки программирования и среды разработки: Python.
7.28	
7.29	При необходимости возможно включение дополнительного оборудования, например:
7.30	
7.31	- Ноутбуки или планшеты для отдельных практических занятий и выездных мероприятий.
7.32	- Датчики и исполнительные механизмы для проведения лабораторно-практических работ по робототехнике и управлению техническими устройствами.
7.33	- Специализированные инструменты для настройки беспроводных сетей Wi-Fi, активного сетевого оборудования.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
Обучающимся необходимо помнить, что качество полученного образования в немалой степени зависит от активной роли самого обучающегося в учебном процессе. Обучающийся должен быть нацелен на максимальное усвоение подаваемого лектором материала, после лекции и во время специально организуемых индивидуальных встреч он может задать лектору интересующие его вопросы. Лекционные занятия составляют основу теоретического обучения и должны давать систематизированные основы знаний по дисциплине, раскрывать состояние и перспективы развития соответствующей области науки, концентрировать внимание обучающихся на наиболее сложных и узловых вопросах, стимулировать их активную познавательную деятельность и

способствовать формированию творческого мышления.

Главная задача лекционного курса - сформировать у обучающихся системное представление об изучаемом предмете, обеспечить усвоение будущими специалистами основополагающего учебного материала, принципов и закономерностей развития соответствующей научно-практической области, а также методов применения полученных знаний, умений и навыков.

Основные функции лекций: 1. Познавательная-обучающая; 2. Развивающая; 3. Ориентирующе-направляющая; 4.

Активизирующая; 5. Воспитательная; 6. Организующая; 7. информационная.

Выполнение практических заданий служит важным связующим звеном между теоретическим освоением данной дисциплины и применением ее положений на практике. Они способствуют развитию самостоятельности обучающихся, более активному освоению учебного материала, являются важной предпосылкой формирования профессиональных качеств будущих специалистов.

Проведение практических занятий не сводится только к органическому дополнению лекционных курсов и самостоятельной работы обучающихся. Их вместе с тем следует рассматривать как важное средство проверки усвоения обучающимися тех или иных положений, даваемых на лекции, а также рекомендуемой для изучения литературы; как форма текущего контроля за отношением обучающихся к учебе, за уровнем их знаний, а следовательно, и как один из важных каналов для своевременного подтягивания отстающих обучающихся.

При подготовке важны не только серьезная теоретическая подготовка, но и умение ориентироваться в разнообразных практических ситуациях, ежедневно возникающих в его деятельности. Этому способствует форма обучения в виде практических занятий. Задачи практических занятий: закрепление и углубление знаний, полученных на лекциях и приобретенных в процессе самостоятельной работы с учебной литературой, формирование у обучающихся умений и навыков работы с исходными данными, научной литературой и специальными документами. Практическому занятию должно предшествовать ознакомление с лекцией на соответствующую тему и литературой, указанной в плане этих занятий.

При проведении учебных занятий обеспечиваются развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая при необходимости проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплин (модулей) в форме курсов, составленных на основе результатов научных исследований, проводимых организацией, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей). Самостоятельная работа может быть успешной при определенных условиях, которые необходимо организовать. Ее правильная организация, включающая технологии отбора целей, содержания, конструирования заданий и организацию контроля, систематичность самостоятельных учебных занятий, целесообразное планирование рабочего времени позволяет привить студентам умения и навыки в овладении, изучении, усвоении и систематизации приобретаемых знаний в процессе обучения, привить навыки повышения профессионального уровня в течение всей трудовой деятельности.

Для контроля знаний студентов по данной дисциплине необходимо проводить оперативный, рубежный и итоговый контроль.

Оперативный контроль осуществляется путем проведения опросов студентов на семинарских занятиях, проверки выполнения практических заданий, а также учета вовлеченности (активности) студентов при обсуждении мини-докладов, организации ролевых игр и т.п.

Контроль за самостоятельной работой студентов по курсу осуществляется в двух формах: текущий контроль и итоговый. Рубежный контроль (аттестация) подразумевает проведение тестирования по пройденным разделам курса. В тестирование могут быть включены темы, предложенные студентам для самостоятельной подготовки, а также практические задания.