



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Кубанский государственный университет»  
в г. Славянске-на-Кубани

УТВЕРЖДАЮ

Проректор по работе с филиалами  
ФГОУ ВО «Кубанский  
государственный университет»

А.А. Евдокимов

«15» декабря 2022 г.



**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**УД.02 ИНФОРМАТИКА**

специальность 09.02.07 Информационные системы и программирование

Краснодар 2022


Рабочая программа учебной дисциплины УД.02 ИНФОРМАТИКА разработана на основе требований ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины УД.02 ИНФОРМАТИКА, с учетом требований примерной рабочей программы общеобразовательной дисциплины «ИНФОРМАТИКА» для профессиональных образовательных организаций, утвержденной на заседании Совета по оценке содержания и качества примерных рабочих программ общеобразовательного и социального-гуманитарного циклов среднего профессионального образования (протокол №14 от 30 ноября 2022 г.), и федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (технологический профиль), утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, (зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936).

Дисциплина	УД.02 ИНФОРМАТИКА
Форма обучения	очная
Учебный год	2023-2024
1 курс	2 семестр
всего 144 часа, в том числе:	
лекции	32 ч.
практические занятия	112 ч.
самостоятельные занятия	—
консультация	—
промежуточная аттестация	—
форма итогового контроля	диф.зачет

Составитель: преподаватель  М.С. Бушуев

Утверждена на заседании предметной (цикловой) комиссии физико-математических дисциплин и специальных дисциплин УГС 09.00.00 Информатика и вычислительная техника протокол № 4 от «24» ноября 2022 г.

Председатель предметной (цикловой) комиссии

 М.С. Бушуев  
«24» ноября 2022 г.

Рецензенты:

Директор МБОУ ООШ № 31 имени  
полного кавалера ордена Славы В.Л. Храпко  
станции Петровской муниципального  
образования Славянский район



В.В. Васильев

Профессор кафедры математики,  
информатики, естественнонаучных  
и общетехнических дисциплин,  
доктор технических наук, профессор



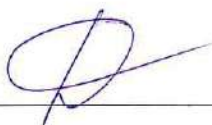
А.А. Маслак

ЛИСТ  
согласования рабочей программы по учебной дисциплине  
УД.02 «Информатика»

Специальность среднего профессионального образования:  
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Нач. УМО филиала



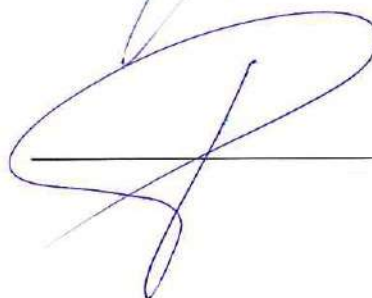
А.С. Демченко  
«25» ноября 2022 г.

Заведующая библиотекой филиала



М.В. Фуфалько  
«25» ноября 2022 г.

Нач. ИВЦ (программно-  
информационное обеспечение  
образовательной программы)



В.А. Ткаченко  
«25» ноября 2022 г.

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	5
1.1. Область применения программы .....	5
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена: .....	5
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины: .....	5
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций) .....	8
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	15
2.1. Объем учебной дисциплины и виды учебной работы .....	15
2.2. Структура дисциплины: .....	15
2.3. Тематический план и содержание учебной дисциплины УД.02 Информатика ..	15
2.4. Содержание разделов дисциплины .....	22
2.4.1. Занятия лекционного типа .....	22
2.4.2. Занятия семинарского типа .....	24
2.4.3. Практические занятия (Лабораторные занятия).....	24
2.4.4. Содержание самостоятельной работы.....	26
2.4.5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине .....	26
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине .....	32
4.2. Перечень необходимого программного обеспечения .....	32
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	33
5.1. Основная литература .....	33
5.3. Периодические издания .....	33
5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	33
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ .....	36
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	36
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ.....	39
7.1. Паспорт фонда оценочных средств.....	39
7.2. Критерии оценки знаний.....	39
7.3. Оценочные средств для проведения текущей аттестации .....	40
7.4. Оценочные средств для проведения промежуточной аттестации .....	51
7.4.1. Примерные вопросы для проведения промежуточной аттестации (Дифференцированный зачет).....	51
7.4.2. Примерные задачи на дифференцированный зачет .....	52
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	59

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ УД.02 ИНФОРМАТИКА

## 1.1. Область применения программы

Общеобразовательная дисциплина «УД.02 Информатика» является обязательной частью общеобразовательного цикла образовательной программы в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование реализуемой на базе основного общего образования. Программа разработана на основании требований ФГОС среднего общего образования.

## 1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Дисциплина входит в обязательные учебные дисциплины, изучаемые на углубленном уровне.

## 1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины:

Содержание программы общеобразовательной дисциплины «Информатика» направлено на достижение следующих **целей**:

- освоение системы базовых знаний, отражающих вклад информатики в формирование современной научной картины мира, роль информационных процессов в современном обществе, биологических и технических системах;
- овладение умениями применять, анализировать, преобразовывать информационные модели реальных объектов и процессов, используя при этом цифровые технологии, в том числе при изучении других дисциплин;
- развитие познавательных интересов, интеллектуальных и творческих способностей путем освоения и использования методов информатики и цифровых технологий при изучении различных учебных предметов;
- воспитание ответственного отношения к соблюдению этических и правовых норм информационной деятельности;
- приобретение опыта использования цифровых технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.

Освоение содержания учебной дисциплины «Информатика» обеспечивает достижение студентами следующих результатов:

### **личностных:**

- осознание обучающимися российской гражданской идентичности;
- готовность к саморазвитию, самостоятельности и самоопределению;
- наличие мотивации к обучению и личностному развитию;
- целенаправленное развитие внутренней позиции личности на основе духовно-нравственных ценностей народов Российской Федерации, исторических и национально-культурных традиций, формирование системы значимых ценностно-смысловых установок, антикоррупционного мировоззрения, правосознания, экологической культуры, способности ставить цели и строить жизненные планы;

### **метапредметных:**

- освоенные обучающимися межпредметные понятия и универсальные учебные действия (регулятивные, познавательные, коммуникативные);

– способность их использования в познавательной и социальной практике, готовность к самостоятельному планированию и осуществлению учебной деятельности, организации учебного сотрудничества с педагогическими работниками и сверстниками, к участию в построении индивидуальной образовательной траектории;

– овладение навыками учебно-исследовательской, проектной и социальной деятельности;

**предметных:**

1) владение представлениями о роли информации и связанных с ней процессов в природе, технике и обществе; понятиями «информация», «информационный процесс», «система», «компоненты системы», «системный эффект», «информационная система», «система управления»; владение методами поиска информации в сети Интернет; умение критически оценивать информацию, полученную из сети Интернет; умение характеризовать большие данные, приводить примеры источников их получения и направления использования;

2) понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владение навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации;

3) наличие представлений о компьютерных сетях и их роли в современном мире; об общих принципах разработки и функционирования интернет-приложений;

4) понимание угроз информационной безопасности, использование методов и средств противодействия этим угрозам, соблюдение мер безопасности, предотвращающих незаконное распространение персональных данных; соблюдение требований техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения; понимание правовых основ использования компьютерных программ, баз данных и работы в сети Интернет;

5) понимание основных принципов дискретизации различных видов информации; умение определять информационный объем текстовых, графических и звуковых данных при заданных параметрах дискретизации;

6) умение строить неравномерные коды, допускающие однозначное декодирование сообщений (префиксные коды); использовать простейшие коды, которые позволяют обнаруживать и исправлять ошибки при передаче данных;

7) владение теоретическим аппаратом, позволяющим осуществлять представление заданного натурального числа в различных системах счисления; выполнять преобразования логических выражений, используя законы алгебры логики; определять кратчайший путь во взвешенном графе и количество путей между вершинами ориентированного ациклического графа;

8) умение читать и понимать программы, реализующие несложные алгоритмы обработки числовых и текстовых данных (в том числе массивов и символьных строк) на выбранном для изучения универсальном языке программирования высокого уровня (Паскаль, Python, Java, C++, C#); анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных; модифицировать готовые программы для решения новых задач, использовать их в своих программах в качестве подпрограмм (процедур, функций);



9) умение реализовать этапы решения задач на компьютере; умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщенных характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива;

10) умение создавать структурированные текстовые документы и демонстрационные материалы с использованием возможностей современных программных средств и облачных сервисов; умение использовать табличные (реляционные) базы данных, в частности, составлять запросы в базах данных (в том числе вычисляемые запросы), выполнять сортировку и поиск записей в базе данных; наполнять разработанную базу данных; умение использовать электронные таблицы для анализа, представления и обработки данных (включая вычисление суммы, среднего арифметического, наибольшего и наименьшего значений, решение уравнений);

11) умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде;

12) умение организовывать личное информационное пространство с использованием различных средств цифровых технологий; понимание возможностей цифровых сервисов государственных услуг, цифровых образовательных сервисов; понимание возможностей и ограничений технологий искусственного интеллекта в различных областях; наличие представлений об использовании информационных технологий в различных профессиональных сферах.

13) умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов;

14) наличие представлений о базовых принципах организации и функционирования компьютерных сетей;

15) умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объема данных и характеристик канала связи;

16) умение строить код, обеспечивающий наименьшую возможную среднюю длину сообщения при известной частоте символов; пояснять принципы работы простых алгоритмов сжатия данных;

17) умение использовать при решении задач свойства позиционной записи чисел, алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; умение выполнять арифметические операции в позиционных системах счисления; умение строить

логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения; умение решать алгоритмические задачи, связанные с анализом графов (задачи построения оптимального пути между вершинами графа, определения количества различных путей между вершинами ориентированного ациклического графа); умение использовать деревья при анализе и построении кодов и для представления арифметических выражений, при решении задач поиска и сортировки; умение строить дерево игры по заданному алгоритму; разрабатывать и обосновывать выигрышную стратегию игры;

18) понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки; умение определять сложность изучаемых в курсе базовых алгоритмов (суммирование элементов массива, сортировка массива, переборные алгоритмы, двоичный поиск) и приводить примеры нескольких алгоритмов разной сложности для решения одной задачи;

19) владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода;

20) умение разрабатывать и реализовывать в виде программ базовые алгоритмы; умение использовать в программах данные различных типов с учетом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; знать функциональные возможности инструментальных средств среды разработки; умение использовать средства отладки программ в среде программирования; умение документировать программы;

21) умение создавать веб-страницы; умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними; умение использовать табличные (реляционные) базы данных и справочные системы.

Максимальная учебная нагрузка обучающегося 144 часов, в том числе:

– обязательная аудиторная учебная нагрузка обучающегося 144 часов.

#### **1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)**

Учащийся должен обладать общими компетенциями, включающими в себя



способность:

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

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

Код и наименование формируемых компетенций	Планируемые результаты освоения дисциплины	
	Общие	Дисциплинарные
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	<p><b>В части трудового воспитания:</b></p> <ul style="list-style-type: none"> <li>- готовность к труду, осознание ценности мастерства, трудолюбие;</li> <li>- готовность к активной деятельности технологической и социальной направленности, способность инициировать, планировать и самостоятельно выполнять такую деятельность;</li> <li>- интерес к различным сферам профессиональной деятельности,</li> </ul> <p><b>Овладение универсальными учебными познавательными действиями:</b></p> <p><b>а) базовые логические действия:</b></p> <ul style="list-style-type: none"> <li>- самостоятельно формулировать и актуализировать проблему, рассматривать ее всесторонне;</li> <li>- устанавливать существенный признак или основания для сравнения, классификации и обобщения;</li> <li>- определять цели деятельности, задавать параметры и критерии их достижения;</li> <li>- выявлять закономерности и противоречия в рассматриваемых явлениях;</li> <li>- вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности;</li> <li>- развивать креативное мышление при решении жизненных проблем</li> </ul> <p><b>б) базовые исследовательские действия:</b></p> <ul style="list-style-type: none"> <li>- владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем;</li> <li>- выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу ее решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения;</li> <li>- анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях;</li> <li>- уметь переносить знания в познавательную и практическую области жизнедеятельности;</li> <li>- уметь интегрировать знания из разных предметных областей;</li> <li>- выдвигать новые идеи, предлагать оригинальные подходы и решения;</li> <li>- способность их использования в познавательной и социальной практике</li> </ul>	<ul style="list-style-type: none"> <li>- понимать угрозу информационной безопасности, использовать методы и средства противодействия этим угрозам, соблюдение мер безопасности, предотвращающих незаконное распространение персональных данных; соблюдение требований техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения; понимание правовых основ использования компьютерных программ, баз данных и работы в сети Интернет;</li> <li>- уметь организовывать личное информационное пространство с использованием различных средств цифровых технологий; понимание возможностей цифровых сервисов государственных услуг, цифровых образовательных сервисов; понимание возможностей и ограничений технологий искусственного интеллекта в различных областях; наличие представлений об использовании информационных технологий в различных профессиональных сферах</li> <li>- уметь реализовать этапы решения задач на компьютере; умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщенных характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива;</li> </ul>

<p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности</p>	<p>В области ценности научного познания:</p> <ul style="list-style-type: none"> <li>- сформированность мировоззрения, соответствующего современному уровню развития науки и общественной практики, основанного на диалоге культур, способствующего осознанию своего места в поликультурном мире;</li> <li>- совершенствование языковой и читательской культуры как средства взаимодействия между людьми и познания мира;</li> <li>- осознание ценности научной деятельности, готовность осуществлять проектную и исследовательскую деятельность индивидуально и в группе;</li> </ul> <p>Овладение универсальными учебными познавательными действиями:</p> <p>в) работа с информацией:</p> <ul style="list-style-type: none"> <li>- владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления;</li> <li>- создавать тексты в различных форматах с учетом назначения информации и целевой аудитории, выбирая оптимальную форму представления и визуализации;</li> <li>- оценивать достоверность, легитимность информации, ее соответствие правовым и морально-этическим нормам;</li> <li>- использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;</li> <li>- владеть навыками распознавания и защиты информации, информационной безопасности личности</li> </ul>	<ul style="list-style-type: none"> <li>- владеть представлениями о роли информации и связанных с ней процессов в природе, технике и обществе; понятиями «информация», «информационный процесс», «система», «компоненты системы» «системный эффект», «информационная система», «система управления»; владеть методами поиска информации в сети Интернет; уметь критически оценивать информацию, полученную из сети Интернет; характеризовать большие данные, приводить примеры источников их получения и направления использования;</li> <li>- понимать основные принципы устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владеть навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации;</li> <li>- иметь представления о компьютерных сетях и их роли в современном мире; об общих принципах разработки и функционирования интернет-приложений;</li> <li>- понимать основные принципы дискретизации различных видов информации; уметь определять информационный объем текстовых, графических и звуковых данных при заданных параметрах дискретизации;</li> <li>- уметь строить неравномерные коды, допускающие однозначное декодирование сообщений (префиксные коды); использовать простейшие коды, которые позволяют обнаруживать и исправлять ошибки при передаче данных;</li> <li>- владеть теоретическим аппаратом, позволяющим осуществлять представление заданного натурального числа в различных системах счисления; выполнять преобразования логических выражений, используя законы алгебры логики; определять кратчайший путь во взвешенном графе и количество путей между вершинами ориентированного ациклического графа;</li> <li>- уметь читать и понимать программы, реализующие несложные алгоритмы обработки числовых и текстовых данных (в том числе массивов и символьных строк) на выбранном для изучения универсальном языке программирования высокого уровня (Паскаль, Python, Java, C++, C#); анализировать алгоритмы с использованием таблиц трассировки; определять без использования</li> </ul>
---	---	--

		<p>компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных; модифицировать готовые программы для решения новых задач, использовать их в своих программах в качестве подпрограмм (процедур, функций);</p> <ul style="list-style-type: none"> <li>- уметь создавать структурированные текстовые документы и демонстрационные материалы с использованием возможностей современных программных средств и облачных сервисов; умение использовать табличные (реляционные) базы данных, в частности, составлять запросы в базах данных (в том числе вычисляемые запросы), выполнять сортировку и поиск записей в базе данных; наполнять разработанную базу данных; умение использовать электронные таблицы для анализа, представления и обработки данных (включая вычисление суммы, среднего арифметического, наибольшего и наименьшего значений, решение уравнений);</li> <li>- уметь использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде;</li> <li>- уметь классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов;</li> <li>- иметь представления о базовых принципах организации и функционирования компьютерных сетей;</li> <li>- уметь определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объема данных и характеристик канала связи;</li> <li>- уметь строить код, обеспечивающий наименьшую возможную среднюю длину сообщения при известной частоте символов; пояснять принципы работы простых алгоритмов сжатия данных;</li> <li>- уметь использовать при решении задач свойства позиционной записи чисел, алгоритмы построения записи числа в позиционной системе счисления с</li> </ul>
--	--	---

		<p>заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; уметь выполнять арифметические операции в позиционных системах счисления; умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения; уметь решать алгоритмические задачи, связанные с анализом графов (задачи построения оптимального пути между вершинами графа, определения количества различных путей между вершинами ориентированного ациклического графа); уметь использовать деревья при анализе и построении кодов и для представления арифметических выражений, при решении задач поиска и сортировки; уметь строить дерево игры по заданному алгоритму; разрабатывать и обосновывать выигрышную стратегию игры;</p> <p>- понимать базовые алгоритмы обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многозначных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки; умение определять сложность изучаемых в курсе базовых алгоритмов (суммирование элементов массива, сортировка массива, переборные алгоритмы, двоичный поиск) и приводить примеры нескольких алгоритмов разной сложности для решения одной задачи;</p> <p>- владеть универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; уметь осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода;</p> <p>- уметь разрабатывать и реализовывать в виде программ базовые алгоритмы; использовать в программах данные различных типов с учетом ограничений</p>
--	--	--

		<p>на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; знать функциональные возможности инструментальных средств среды разработки; умение использовать средства отладки программ в среде программирования; умение документировать программы;</p> <p>- уметь создавать веб-страницы; умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); владеть основными сведениями о базах данных, их структуре, средствах создания и работы с ними; использовать табличные (реляционные) базы данных и справочные системы</p>
<p>ПК Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p>	<p>1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p>	<p>Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства.</p> <p>- уметь использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде;</p> <p>- уметь классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов;</p> <p>- иметь представления о базовых принципах организации и функционирования компьютерных сетей;</p> <p>- уметь определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объема данных и характеристик канала связи;</p> <p>- владеть представлениями о роли информации и связанных с ней процессов в природе, технике и обществе; понятиями «информация», «информационный процесс», «система», «компоненты системы» «системный эффект», «информационная система», «система управления»; владеть методами поиска информации в сети Интернет; уметь критически оценивать</p>

		<p>информацию, полученную из сети Интернет; характеризовать большие данные, приводить примеры источников их получения и направления использования;</p> <ul style="list-style-type: none"><li>- понимать основные принципы устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владеть навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации;</li><li>- иметь представления о компьютерных сетях и их роли в современном мире; об общих принципах разработки и функционирования интернет-приложений.</li></ul>
--	--	--

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	144
Обязательная аудиторная учебная нагрузка (всего)	144
в том числе:	
занятия лекционного типа	32
практические занятия	112
лабораторные занятия	–
Самостоятельная работа обучающегося (всего)	–
Промежуточная аттестация в форме	диф. зачета

### 2.2. Структура дисциплины:

Наименование разделов и тем	Количество аудиторных часов		
	Всего	Теоретическое обучение	Практические и лабораторные занятия
Раздел 1. Информация и информационная деятельность человека	22	8	14
Раздел 2. Использование программных систем и сервисов	22	–	22
Раздел 3. Информационное моделирование	28	8	20
Прикладной модуль 2. Аналитика и визуализация данных на Python	36	2	34
Прикладной модуль 7. Введение в веб-разработку на языке JavaScript	36	14	22
<b>Всего по дисциплине</b>	<b>144</b>	<b>32</b>	<b>112</b>

### 2.3. Тематический план и содержание учебной дисциплины УД.02 Информатика

Наименование разделов и тем	Содержание учебного материала (основное и профессионально-ориентированное), лабораторные и практические занятия, прикладной модуль (при наличии)	Объем часов	Формируемые компетенции
<b>Основное содержание</b>			
Раздел 1.	Информация и информационная деятельность человека	22	
Тема 1.1. Информация и информационные процессы	Основное содержание	2	ОК 02
	Понятие «информация» как фундаментальное понятие современной науки. Представление об основных информационных процессах, о системах. Кодирование информации Информация и информационные		



	процессы		
	Теоретическое обучение	2	
<b>Тема 1.2. Подходы к измерению информации</b>	Основное содержание	<b>4</b>	ОК 02
	Подходы к измерению информации (содержательный, алфавитный, вероятностный). Единицы измерения информации. Информационные объекты различных видов. Универсальность дискретного (цифрового) представления информации. Передача и хранение информации. Определение объемов различных носителей информации. Архив информации		
	Практические занятия	4	
<b>Тема 1.3. Компьютер и цифровое представление информации. Устройство компьютера</b>	Основное содержание	<b>2</b>	ОК 02
	Принципы построения компьютеров. Принцип открытой архитектуры. Магистраль. Аппаратное устройство компьютера. Внешняя память. Устройства ввода-вывода. Поколения ЭВМ. Архитектура ЭВМ 5 поколения. Основные характеристики компьютеров. Программное обеспечение: классификация и его назначение, сетевое программное обеспечение		
	Теоретическое обучение	2	
<b>Тема 1.4. Кодирование информации. Системы счисления</b>	Основное содержание	<b>4</b>	ОК 02
	Представление о различных системах счисления, представление вещественного числа в системе счисления с любым основанием, перевод числа из десятичной позиционной системы счисления в десятичную, перевод вещественного числа из 10 СС в другую СС, арифметические действия в разных СС. Представление числовых данных: общие принципы представления данных, форматы представления чисел. Представление текстовых данных: кодовые таблицы символов, объем текстовых данных. Представление графических данных. Представление звуковых данных. Представление видеоданных. Кодирование данных произвольного вида		
	Практические занятия	4	
<b>Тема 1.5. Элементы комбинаторики, теории множеств и математической логики</b>	Основное содержание	<b>2</b>	ОК 02
	Основные понятия алгебры логики: высказывание, логические операции, построение таблицы истинности логического выражения. Графический метод алгебры логики. Понятие множества. Мощность множества. Операции над множествами. Решение логических задач графическим способом		
	Практические занятия	2	
<b>Тема 1.6. Компьютерные сети: локальные сети, сеть Интернет</b>	Основное содержание	<b>2</b>	ОК 01 ОК 02
	Компьютерные сети их классификация. Работа в локальной сети. Топологии локальных сетей. Обмен данными. Глобальная сеть Интернет. IP-адресация.		

	Правовые основы работы в сети Интернет		
	Теоретическое обучение	2	
<b>Тема 1.7. Услуги Интернета</b>	Основное содержание	2	ОК 02
	Услуги и сервисы Интернета (электронная почта, видеоконференции, форумы, мессенджеры, социальные сети). Поиск в Интернете. Электронная коммерция. Цифровые сервисы государственных услуг. Достоверность информации в Интернете		
	Практические занятия	2	
<b>Тема 1.8. Сетевое хранение данных и цифрового контента</b>	Основное содержание	2	ОК 01 ОК 02
	Организация личного информационного пространства. Облачные хранилища данных. Разделение прав доступа в облачных хранилищах. Коллективная работа над документами. Соблюдение мер безопасности, предотвращающих незаконное распространение персональных данных		
	Практические занятия	2	
<b>Тема 1.9. Информационная безопасность</b>	Основное содержание	2	ОК 01 ОК 02
	Информационная безопасность. Защита информации. Информационная безопасность в мире, России. Вредоносные программы. Антивирусные программы. Безопасность в Интернете (сетевые угрозы, мошенничество). Тренды в развитии цифровых технологий; риски и прогнозы использования цифровых технологий при решении профессиональных задачи		
	Теоретическое обучение	2	
<b>Раздел 2.</b>	<b>Использование программных систем и сервисов</b>	<b>22</b>	
<b>Тема 2.1. Обработка информации в текстовых процессорах</b>	Основное содержание	4	ОК 02
	Текстовые документы. Виды программного обеспечения для обработки текстовой информации. Создание текстовых документов на компьютере (операции ввода, редактирования, форматирования)		
	Практические занятия	4	
<b>Тема 2.2. Технологии создания структурированных текстовых документов</b>	Основное содержание	4	ОК 02
	Многостраничные документы. Структура документа. Гипертекстовые документы. Совместная работа над документом. Шаблоны.		
	Практические занятия	4	
<b>Тема 2.3. Компьютерная графика и мультимедиа</b>	Основное содержание	4	ОК 02
	Компьютерная графика и её виды. Форматы мультимедийных файлов. Графические редакторы (ПО Gimp, Inkscape). Программы по записи и редактированию звука (ПО АудиоМастер). Программы редактирования видео (ПО Movavi)		
	Практические занятия	4	
<b>Тема 2.4. Технологии обработки</b>	Основное содержание	4	ОК 02
	Технологии обработки различных объектов		

<b>графических объектов</b>	компьютерной графики (растровые и векторные изображения, обработка звука, монтаж видео)		
	Практические занятия	4	
<b>Тема 2.5. Представление профессиональной информации в виде презентаций</b>	Основное содержание	2	ОК 02
	Виды компьютерных презентаций. Основные этапы разработки презентации. Анимация в презентации. Шаблоны. Композиция объектов презентации		
	Практические занятия	2	
<b>Тема 2.6. Интерактивные и мультимедийные объекты на слайде</b>	Основное содержание	2	ОК 02
	Принципы мультимедиа. Интерактивное представление информации		
	Практические занятия	2	
<b>Тема 2.7. Гипертекстовое представление информации</b>	Основное содержание	2	ОК 02
	Язык разметки гипертекста HTML. Оформление гипертекстовой страницы. Веб-сайты и веб-страницы		
	Практические занятия	2	
<b>Раздел 3.</b>	<b>Информационное моделирование</b>	<b>28</b>	
<b>Тема 3.1. Модели и моделирование. Этапы моделирования</b>	Основное содержание	2	ОК 02
	Представление о компьютерных моделях. Виды моделей. Адекватность модели. Основные этапы компьютерного моделирования		
	Теоретическое обучение	2	
<b>Тема 3.2. Списки, графы, деревья</b>	Основное содержание	2	ОК 02
	Структура информации. Списки, графы, деревья. Алгоритм построения дерева решений		
	Теоретическое обучение	2	
<b>Тема 3.3. Математические модели в профессиональной области</b>	Основное содержание	2	ОК 02
	Алгоритмы моделирования кратчайших путей между вершинами (Алгоритм Дейкстры, Метод динамического программирования). Элементы теории игр (выигрышная стратегия)		
	Практические занятия	2	
<b>Тема 3.4. Понятие алгоритма и основные алгоритмические структуры</b>	Основное содержание	4	ОК 01
	Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма. Основные алгоритмические структуры. Запись алгоритмов на языке программирования (Pascal, Python, Java, C++, C#). Анализ алгоритмов с помощью трассировочных таблиц		
	Практические занятия	4	
<b>Тема 3.5. Анализ алгоритмов в профессиональной области</b>	Основное содержание	4	ОК 02
	Структурированные типы данных. Массивы. Вспомогательные алгоритмы. Задачи поиска элемента с заданными свойствами. Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 3.6. Базы данных</b>	Основное содержание	<b>6</b>	ОК 02

<b>как модель предметной области</b>	Базы данных как модель предметной области. Таблицы и реляционные базы данных		
	Теоретическое обучение	2	
	Практические занятия	4	
<b>Тема 3.7. Технологии обработки информации в электронных таблицах</b>	Основное содержание	2	ОК 02
	Табличный процессор. Приемы ввода, редактирования, форматирования в табличном процессоре. Адресация. Сортировка, фильтрация, условное форматирование		
	Практические занятия	2	
<b>Тема 3.8. Формулы и функции в электронных таблицах</b>	Основное содержание	2	ОК 02
	Формулы и функции в электронных таблицах. Встроенные функции и их использование. Математические и статистические функции. Логические функции. Финансовые функции. Текстовые функции. Реализация математических моделей в электронных таблицах		
	Практические занятия	2	
<b>Тема 3.9. Визуализация данных в электронных таблицах</b>	Основное содержание	2	ОК 02
	Визуализация данных в электронных таблицах		
	Практические занятия	2	
<b>Тема 3.10. Моделирование в электронных таблицах (на примерах задач из профессиональной области)</b>	Основное содержание	2	ОК 02
	Моделирование в электронных таблицах (на примерах задач из профессиональной области)		
	Практические занятия	2	
<b>Профессионально-ориентированное содержание (содержание прикладного модуля)</b>			
<b>Прикладной модуль 2</b>	<b>Аналитика и визуализация данных на Python</b>	<b>36</b>	
<b>Тема 2.1. Введение в язык программирования Python</b>	Содержание	2	ОК 02 <i>ПК.1.1</i>
	Интерактивная среда программирование на Python. Ввод и вывод данных. Функции print(), input(). Типы данных. Математические операции с целыми и вещественными числами		
<b>Тема 2.2. Основные алгоритмические конструкции на Python</b>	Содержание	4	ОК 02 <i>ПК.1.1</i>
	Понятие логических выражений и операций. Дизъюнкция, конъюнкция, отрицание. Таблица истинности. Проверка условия в Python. Синтаксис инструкций if, if-else, if-elif-else. Реализация циклических алгоритмов в Python. Функция range(). Синтаксис цикла for, цикла while		
	Практические занятия		
<b>Тема 2.3. Работа со списками и словарями</b>	Содержание	6	ОК 02 <i>ПК.1.1</i>
	Понятие списка в Python. Создание и считывание списков. Функции и методы списков. Понятие словаря. Отличия словарей от списков. Создание словаря. Методы словарей. Применение списков и словарей в реальных задачах.		

	Практические занятия	4	
	Контрольные работы	2	
<b>Тема 2.4. Аналитика данных на Python</b>	Содержание	8	ОК 02 ПК.1.1
	Понятие данных, больших данных. Наборы данных. Платформа Kaggle. Библиотека Pandas. Объекты Series и DataFrame. Получение общей информации о данных. Индексация по условиям и изменение данных в таблицах.		
	Практические занятия	8	
<b>Тема 2.5. Анализ данных на практических примерах</b>	Содержание	6	ОК 02 ПК.1.1
	Понятие статистики, описательной статистики. Описательный анализ данных. Основные описательные статистические величины (частота, среднее арифметическое, медиана, мода, размах, стандартное отклонение). Функции описательной статистики в Python Pandas. Практика вычисления описательных статистических величин в Python Pandas		
	Практические занятия	6	
<b>Тема 2.6. Основы визуализации данных</b>	Содержание	6	ОК 02 ПК.1.1
	Необходимость визуализации данных для анализа. Понятие научной графики. Библиотека Matplotlib. Понятие рисунка в Matplotlib. Основные виды графиков (гистограммы, диаграммы рассеяния, диаграмма размаха, линейный график, круговая диаграмма, тепловые карты). Основные графические команды в Matplotlib		
	Практические занятия	6	
<b>Тема 2.7. Проектная работа «Анализ больших данных в профессиональной сфере»</b>	Содержание	4	ОК 02 ПК.1.1
	Характеристика основных этапов процесса анализа данных. Подготовка данных. Исследование и визуализация данных. Построение предсказательной модели. Интерпретация результатов анализа. Реализация основных этапов процесса анализа данных на примере набора данных из профессиональной сферы		
	Практические занятия	4	
<b>Прикладной модуль 7</b>	<b>Введение в веб-разработку на языке JavaScript</b>	<b>36</b>	
<b>Тема 7.1. Синтаксис и основные понятия JavaScript</b>	Содержание	2	ОК 02 ПК.1.1
	Выражения, операторы, побочные эффекты, инструкции, ввод-вывод. Понятие объекта и литерала. Эволюция стандарта ECMAScript		
	Теоретическое обучение	2	
<b>Тема 7.2. Управление пакетами и зависимостями</b>	Содержание	2	ОК 02 ПК.1.1
	Система пакетов npm. Инициализация проекта. Создание файла package.json. Девелоперские зависимости		
	Практические занятия	2	
<b>Тема 7.3. Переменные и области видимости.</b>	Содержание	2	ОК 02 ПК.1.1
	Объявление переменных. Этап компиляции и этап		

<b>Примитивные и объектные типы данных</b>	исполнения. Ошибка ReferenceError и возбуждение исключения. Глобальные переменные. Видимость на уровне блока. Сравнение примитивных значений		
	Практические занятия	2	
<b>Тема 7.4. TypeScript и статическая типизация. Функции как структурный элемент сценария и как тип данных</b>	Содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Типы данных. Объявление с аннотацией типа. Транспиляция и запуск проекта. Объявление (в том числе с аннотацией) и вызов функций		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 7.5. Управляющие конструкции</b>	Основное содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Императивный подход к созданию кода программы. Инструкции как противоположность выражений. Тернарный оператор и инструкция If..else Циклы со счётчиком, предусловием/постусловием, итерационные		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 7.6. Строки и бинарные данные. Регулярные выражения</b>	Содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Строка как примитивный тип данных. Перебор строки с помощью итераций for..of, использование Юникода в JavaScript. Отличие бинарных данных от строк. Поиск совпадений с регулярным выражением		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 7.7. Массивы и множества</b>	Содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Массивы как наборы значений разных типов, допускающих итерацию. Задание массива литералом. Методы массивов, в том числе forEach и reduce. Взаимные преобразования массивов и строк. Множества как наборы не повторяющихся значений. Получение множества из массива		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 7.8. Литеральные объекты. Прототипы и конструкторы. Свойства и методы</b>	Содержание	<b>6</b>	ОК 02 <i>ПК.1.1</i>
	Литеральные объекты как коллекции свойств и методов. Отличия литеральных объектов от блоков и массивов. Доступ к свойствам и методам. Использование ссылки this. Вызов методов одного объекта относительно другого. Доступ к прототипу объекта. Создание объекта с помощью конструктора		
	Теоретическое обучение	2	
	Практические занятия	4	
<b>Тема 7.9. Модули и транспиляция. DOM</b>	Содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Модули как единицы независимого изолированного кода. Импорт и экспорт из модулей в стиле ES2015. Использование возможностей планируемых следующих версий стандарта – преобразование кода с помощью		

	Vabel. Введение в Document Object Model – объектную модель документа веб-страницы		
	Теоретическое обучение	2	
	Практические занятия	2	
<b>Тема 7.10. Проектная работа. «Создание простейшего серверного веб-приложения»</b>	Содержание	<b>4</b>	ОК 02 <i>ПК.1.1</i>
	Проектная работа «Создание простейшего серверного веб-приложения»		
	Практические занятия	4	
<b>Всего</b>		<b>144ч.</b>	

## 2.4. Содержание разделов дисциплины

### 2.4.1. Занятия лекционного типа

№ раздела	Наименование раздела/темы	Содержание раздела	Форма текущего контроля
1	2	3	4
<i>2 семестр</i>			
1	<b>Раздел 1. Информация и информационная деятельность человека</b>	Понятие «информация» как фундаментальное понятие современной науки. Представление об основных информационных процессах, о системах. Кодирование информации Информация и информационные процессы. Принципы построения компьютеров. Принцип открытой архитектуры. Магистраль. Аппаратное устройство компьютера. Внешняя память. Устройства ввода-вывода. Поколения ЭВМ. Архитектура ЭВМ 5 поколения. Основные характеристики компьютеров. Программное обеспечение: классификация и его назначение, сетевое программное обеспечение. Компьютерные сети их классификация. Работа в локальной сети. Топологии локальных сетей. Обмен данными. Глобальная сеть Интернет. IP-адресация. Правовые основы работы в сети Интернет. Информационная безопасность. Защита информации. Информационная безопасность в мире, России. Вредоносные программы. Антивирусные программы. Безопасность в Интернете (сетевые угрозы, мошенничество). Тренды в развитии цифровых технологий; риски и прогнозы использования цифровых технологий при решении профессиональных задачи	У
2	<b>Раздел 2. Использование программных систем и сервисов</b>	–	У



№ раздела	Наименование раздела/темы	Содержание раздела	Форма текущего контроля
1	2	3	4
3	<b>Раздел 3. Информационное моделирование</b>	Представление о компьютерных моделях. Виды моделей. Адекватность модели. Основные этапы компьютерного моделирования. Структура информации. Списки, графы, деревья. Алгоритм построения дерева решений. Структурированные типы данных. Массивы. Вспомогательные алгоритмы. Задачи поиска элемента с заданными свойствами. Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов. Базы данных как модель предметной области. Таблицы и реляционные базы данных	У
4	<b>Прикладной модуль 2. Аналитика и визуализация данных на Python</b>	Интерактивная среда программирование на Python. Ввод и вывод данных. Функции print(), input(). Типы данных. Математические операции с целыми и вещественными числами	У
5	<b>Прикладной модуль 7. Введение в веб-разработку на языке JavaScript</b>	Выражения, операторы, побочные эффекты, инструкции, ввод-вывод. Понятие объекта и литерала. Эволюция стандарта ECMAScript. Типы данных. Объявление с аннотацией типа. Транспилиция и запуск проекта. Объявление (в том числе с аннотацией) и вызов функций. Императивный подход к созданию кода программы. Инструкции как противоположность выражений. Тернарный оператор и инструкция If.else Циклы со счётчиком, предусловием/постусловием, итерационные. Строка как примитивный тип данных. Перебор строки с помощью итераций for..of, использование Юникода в JavaScript. Отличие бинарных данных от строк. Поиск совпадений с регулярным выражением. Массивы как наборы значений разных типов, допускающих итерацию. Задание массива литералом. Методы массивов, в том числе forEach и reduce. Взаимные преобразования массивов и строк. Множества как наборы не повторяющихся значений. Получение множества из массива. Литеральные объекты как коллекции свойств и методов. Отличия литеральных объектов от блоков и массивов. Доступ к свойствам и методам. Использование ссылки this. Вызов методов одного объекта относительно другого. Доступ к прототипу объекта. Создание объекта с помощью конструктора. Модули как единицы независимого изолированного кода. Импорт и экспорт из модулей в стиле ES2015. Использование возможностей планируемых следующих версий стандарта – преобразование кода с помощью Babel. Введение в Document Object Model – объектную модель документа веб-страницы.	У
Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа			

## 2.4.2. Занятия семинарского типа

– не предусмотрены

## 2.4.3. Практические занятия (Лабораторные занятия)

№	Наименование раздела/темы	Наименование практических (лабораторных) работ	Форма текущего контроля
<i>2 семестр</i>			
1	2	3	4
1.	<b>Раздел 1. Информация и информационная деятельность человека</b>	Практические занятия 1-2 Подходы к измерению информации	ПР
		Практические занятия 3-4 Кодирование информации. Системы счисления	
		Практическое занятие 5 Элементы комбинаторики, теории множеств и математической логики	
		Практическое занятие 6 Службы Интернета	
		Практическое занятие 7 Сетевое хранение данных и цифрового контента	
2.	<b>Раздел 2. Использование программных систем и сервисов</b>	Практические занятия 8-9 Обработка информации в текстовых процессорах	ПР
		Практические занятия 10-11 Технологии создания структурированных текстовых документов	
		Практические занятия 12-13 Компьютерная графика и мультимедиа	
		Практические занятия 14-15 Технологии обработки графических объектов	
		Практическое занятие 16 Представление профессиональной информации в виде презентаций	
		Практическое занятие 17 Интерактивные и мультимедийные объекты на слайде	
		Практическое занятие 18 Гипертекстовое представление информации	
3.	<b>Раздел 3. Информационное моделирование</b>	Практическое занятие 19 Математические модели в профессиональной области	ПР
		Практические занятия 20-21 Понятие алгоритма и основные алгоритмические структуры	
		Практическое занятие 22 Анализ алгоритмов в профессиональной области	
		Практические занятия 23-24 Базы данных как модель предметной области	
		Практическое занятие 25 Технологии обработки информации в электронных таблицах	
		Практическое занятие 26 Формулы и функции в электронных таблицах	
		Практическое занятие 27 Визуализация данных в электронных таблицах	
		Практическое занятие 28 Моделирование в электронных таблицах (на примерах задач из профессиональной области)	
4.	<b>Прикладной модуль 2.</b>	Практические занятия 29-30 Понятие логических выражений и операций. Дизъюнкция, конъюнкция, отрицание. Таблица	ПР

	<b>Аналитика и визуализация данных на Python</b>	<p>истинности. Проверка условия в Python. Синтаксис инструкций if, if-else, if-elif-else. Реализация циклических алгоритмов в Python. Функция range(). Синтаксис цикла for, цикла while</p> <p>Практические занятия 31-32 Понятие списка в Python. Создание и считывание списков. Функции и методы списков. Понятие словаря. Отличия словарей от списков. Создание словаря. Методы словарей. Применение списков и словарей в реальных задачах.</p> <p>Практические занятия 33 Контрольная работа</p> <p>Практические занятия 34-37 Понятие данных, больших данных. Наборы данных. Платформа Kaggle. Библиотека Pandas. Объекты Series и DataFrame. Получение общей информации о данных. Индексация по условиям и изменение данных в таблицах.</p> <p>Практические занятия 38-40 Понятие статистики, описательной статистики. Описательный анализ данных. Основные описательные статистические величины (частота, среднее арифметическое, медиана, мода, размах, стандартное отклонение). Функции описательной статистики в Python Pandas. Практика вычисления описательных статистических величин в Python Pandas</p> <p>Практические занятия 41-43 Необходимость визуализации данных для анализа. Понятие научной графики. Библиотека Matplotlib. Понятие рисунка в Matplotlib. Основные виды графиков (гистограммы, диаграммы рассеяния, диаграмма размаха, линейный график, круговая диаграмма, тепловые карты). Основные графические команды в Matplotlib</p> <p>Практические занятия 44-45 Характеристика основных этапов процесса анализа данных. Подготовка данных. Исследование и визуализация данных. Построение предсказательной модели. Интерпретация результатов анализа. Реализация основных этапов процесса анализа данных на примере набора данных из профессиональной сферы</p>	
5.	<b>Прикладной модуль 7. Введение в веб-разработку на языке JavaScript</b>	<p>Практические занятия 46 Система пакетов npm. Инициализация проекта. Создание файла package.json. Девелоперские зависимости</p> <p>Практические занятия 47 Объявление переменных. Этап компиляции и этап исполнения. Ошибка ReferenceError и возбуждение исключения. Глобальные переменные. Видимость на уровне блока. Сравнение примитивных значений</p> <p>Практические занятия 48 Типы данных. Объявление с аннотацией типа. Транспиляция и запуск проекта. Объявление (в том числе с аннотацией) и вызов функций</p> <p>Практические занятия 49 Императивный подход к созданию кода программы. Инструкции как противоположность выражений. Тернарный оператор и инструкция If.else Циклы со счётчиком, предусловием/постусловием, итерационные</p> <p>Практические занятия 50 Строка как примитивный тип данных. Перебор строки с помощью итераций for.of, использование Юникода в JavaScript. Отличие бинарных данных от строк. Поиск совпадений с регулярным выражением</p>	<p>ПР</p>

		Практические занятия 51 Массивы как наборы значений разных типов, допускающих итерацию. Задание массива литералом. Методы массивов, в том числе forEach и reduce. Взаимные преобразования массивов и строк. Множества как наборы не повторяющихся значений. Получение множества из массива	
		Практические занятия 52-53 Литеральные объекты как коллекции свойств и методов. Отличия литеральных объектов от блоков и массивов. Доступ к свойствам и методам. Использование ссылки this. Вызов методов одного объекта относительно другого. Доступ к прототипу объекта. Создание объекта с помощью конструктора	
		Практические занятия 54 Модули как единицы независимого изолированного кода. Импорт и экспорт из модулей в стиле ES2015. Использование возможностей планируемых следующих версий стандарта – преобразование кода с помощью Babel. Введение в Document Object Model – объектную модель документа веб-страницы	
		Практические занятия 55-56 Проектная работа «Создание простейшего серверного веб-приложения»	

*Примечание: ПР- практическая работа, ЛР- лабораторная работа; Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа, ДЗ – домашнее задание*

#### **2.4.4. Содержание самостоятельной работы**

*Не предусмотрено учебным планом*

#### **2.4.5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

*Не предусмотрено учебным планом*

### 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для реализации компетентностного подхода предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе преподавания применяются образовательные технологии развития критического мышления. Обязательны компьютерные лабораторные практикумы по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

#### 3.1. Образовательные технологии при проведении лекций

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
1	<b>Раздел 1. Информация и информационная деятельность человека</b>	развивающее обучение	8
2	<b>Раздел 2. Использование программных систем и сервисов</b>		–
3	<b>Раздел 3. Информационное моделирование</b>	развивающее обучение	8*
4	<b>Прикладной модуль 2. Аналитика и визуализация данных на Python</b>	развивающее обучение, игровое обучение, концентрированное обучение	2*
5	<b>Прикладной модуль 6. Введение в веб-разработку на языке JavaScript</b>	развивающее обучение, концентрированное обучение	14*
		Итого по курсу	32
		в том числе интерактивное обучение*	24

#### 3.2. Образовательные технологии при проведении практических занятий (лабораторных работ)

№	Тема занятия	Виды применяемых образовательных технологий	Кол. час
1	Практические занятия 1-2 Подходы к измерению информации	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
2	Практические занятия 3-4 Кодирование информации. Системы счисления	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4
3	Практическое занятие 5 Элементы	решение ситуативных и производственных	2*

	комбинаторики, теории множеств и математической логики	задач, разбор решения задач	
4	Практическое занятие 6 Службы Интернета	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
5	Практическое занятие 7 Сетевое хранение данных и цифрового контента	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
6	Практические занятия 8-9 Обработка информации в текстовых процессорах	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4
7	Практические занятия 10-11 Технологии создания структурированных текстовых документов	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
8	Практические занятия 12-13 Компьютерная графика и мультимедиа	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4
9	Практические занятия 14-15 Технологии обработки графических объектов	решение ситуативных и производственных задач, разбор решения задач	4
10	Практическое занятие 16 Представление профессиональной информации в виде презентаций	решение ситуативных и производственных задач, разбор решения задач	2*
11	Практическое занятие 17 Интерактивные и мультимедийные объекты на слайде	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
12	Практическое занятие 18 Гипертекстовое представление информации	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
13	Практическое занятие 19 Математические модели в профессиональной области	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
14	Практические занятия 20-21 Понятие алгоритма и основные алгоритмические структуры	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
15	Практическое занятие 22 Анализ алгоритмов в профессиональной области	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
16	Практические занятия 23-24 Базы данных как модель предметной области	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4
17	Практическое занятие 25 Технологии	анализ конкретных ситуаций, решение задач	2

	обработки информации в электронных таблицах	малыми группами, решение ситуативных и производственных задач, разбор решения задач	
18	Практическое занятие 26 Формулы и функции в электронных таблицах	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
19	Практическое занятие 27 Визуализация данных в электронных таблицах	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
20	Практическое занятие 28 Моделирование в электронных таблицах (на примерах задач из профессиональной области)	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2
21	Практические занятия 29-30 Понятие логических выражений и операций. Дизъюнкция, конъюнкция, отрицание. Таблица истинности. Проверка условия в Python. Синтаксис инструкций if, if-else, if-elif-else. Реализация циклических алгоритмов в Python. Функция range(). Синтаксис цикла for, цикла while	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4
22	Практические занятия 31-32 Понятие списка в Python. Создание и считывание списков. Функции и методы списков. Понятие словаря. Отличия словарей от списков. Создание словаря. Методы словарей. Применение списков и словарей в реальных задачах.	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
23	Практические занятия 33 Контрольная работа	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
24	Практические занятия 34-37 Понятие данных, больших данных. Наборы данных. Платформа Kaggle. Библиотека Pandas. Объекты Series и DataFrame. Получение общей информации о данных. Индексация по условиям и изменение данных в таблицах.	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	8*
25	Практические занятия 38-40 Понятие статистики, описательной статистики. Описательный анализ данных. Основные описательные статистические величины (частота, среднее арифметическое, медиана, мода, размах, стандартное отклонение). Функции описательной статистики в Python Pandas. Практика вычисления описательных статистических величин в Python Pandas	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	6*



26	Практические занятия 41-43 Необходимость визуализации данных для анализа. Понятие научной графики. Библиотека Matplotlib. Понятие рисунка в Matplotlib. Основные виды графиков (гистограммы, диаграммы рассеяния, диаграмма размаха, линейный график, круговая диаграмма, тепловые карты). Основные графические команды в Matplotlib	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	6*
27	Практические занятия 44-45 Характеристика основных этапов процесса анализа данных. Подготовка данных. Исследование и визуализация данных. Построение предсказательной модели. Интерпретация результатов анализа. Реализация основных этапов процесса анализа данных на примере набора данных из профессиональной сферы	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
28	Практические занятия 46 Система пакетов прм. Инициализация проекта. Создание файла package.json. Девелоперские зависимости	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
29	Практические занятия 47 Объявление переменных. Этап компиляции и этап исполнения. Ошибка ReferenceError и возбуждение исключения. Глобальные переменные. Видимость на уровне блока. Сравнение примитивных значений	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
30	Практические занятия 48 Типы данных. Объявление с аннотацией типа. Транспиляция и запуск проекта. Объявление (в том числе с аннотацией) и вызов функций	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
31	Практические занятия 49 Императивный подход к созданию кода программы. Инструкции как противоположность выражений. Тернарный оператор и инструкция If..else Циклы со счётчиком, предусловием/постусловием, итерационные	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
32	Практические занятия 50 Строка как примитивный тип данных. Перебор строки с помощью итераций for..of, использование Юникода в JavaScript. Отличие бинарных данных от строк. Поиск совпадений с регулярным выражением	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
33	Практические занятия 51 Массивы как наборы значений разных типов, допускающих итерацию. Задание массива литералом. Методы массивов, в том числе forEach и reduce. Взаимные преобразования массивов и строк. Множества как наборы не	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*

	повторяющихся значений. Получение множества из массива		
34	Практические занятия 52-53 Литеральные объекты как коллекции свойств и методов. Отличия литеральных объектов от блоков и массивов. Доступ к свойствам и методам. Использование ссылки this. Вызов методов одного объекта относительно другого. Доступ к прототипу объекта. Создание объекта с помощью конструктора	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
35	Практические занятия 54 Модули как единицы независимого изолированного кода. Импорт и экспорт из модулей в стиле ES2015. Использование возможностей планируемых следующих версий стандарта – преобразование кода с помощью Babel. Введение в Document Object Model – объектную модель документа веб-страницы	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	2*
36	Практические занятия 55-56 Проектная работа «Создание простейшего серверного веб-приложения»	анализ конкретных ситуаций, решение задач малыми группами, решение ситуативных и производственных задач, разбор решения задач	4*
		Итого по курсу	112
		в том числе интерактивное обучение*	72*

## 4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

### 4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Реализация дисциплины требует наличия учебной компьютерной лаборатории информатики.

Оборудование компьютерной лаборатории:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- маркерная доска;
- учебно-методическое обеспечение.

Технические средства обучения:

- компьютеры по количеству обучающихся;
- локальная компьютерная сеть и глобальная сеть Интернет;
- лицензионное системное и прикладное программное обеспечение;
- лицензионное антивирусное программное обеспечение;
- лицензионное специализированное программное обеспечение;
- мультимедиапроектор.

### 4.2. Перечень необходимого программного обеспечения

1. 7-zip архиватор; (лицензия на англ.)
2. Adobe Acrobat Reader просмотрщик файлов ; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
3. Adobe Flash Player –графический редактор; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
4. Apache OpenOffice – офисный пакет; (лицензия - <http://www.openoffice.org/license.html>)
5. FreeCommander - проводник; (лицензия - <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>)
6. Google Chrome - браузер;(лицензия - [https://www.google.ru/chrome/browser/privacy/eula\\_text.html](https://www.google.ru/chrome/browser/privacy/eula_text.html))
7. LibreOffice – офисный пакет(в свободном доступе);
8. Mozilla Firefox - браузер.(лицензия - <https://www.mozilla.org/en-US/MPL/2.0/>)

## 5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ИНФОРМАТИКА

### 5.1. Основная литература

1. Цветкова, М. С. Информатика : учебник для студентов учреждений среднего профессионального образования / М. С. Цветкова, И. Ю. Хлобыстова. – 6-е изд., стер. – Москва : Академия, 2020. – 350 с. – (Профессиональное образование. Общеобразовательные дисциплины). – ISBN 978-5-4468-8663-0.

### 5.2 Дополнительная литература

1. Алешина, А.В. Информатика. 10 класс : учебник / А. В. Алешина, А. С. Крикунов, С. Б. Пересветов [и др.] – Москва : КноРус, 2021. – 243 с. – URL: <https://book.ru/book/941162>. – ISBN 978-5-406-08249-2.

2. Алешина, А.В. Информатика. 11 класс : учебник / А. В. Алешина, А. Л. Булгаков, А. С. Крикунов, М. А. Кузнецова. – Москва : КноРус, 2021. – 271 с. – URL: <https://book.ru/book/941161>. – ISBN 978-5-406-08250-8.

### 5.3. Периодические издания

1. Информатика в школе. – URL : <http://dlib.eastview.com/browse/publication/18988>

2. Информатика. – URL: [https://elibrary.ru/title\\_about\\_new.asp?id=64817](https://elibrary.ru/title_about_new.asp?id=64817)

3. Информатика и образование. – URL: <http://dlib.eastview.com/browse/publication/18946>

4. Инфокоммуникационные технологии. – URL : [https://www.elibrary.ru/title\\_about\\_new.asp?id=9585](https://www.elibrary.ru/title_about_new.asp?id=9585)

### 5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «**BOOK.ru**» : [учебные издания – коллекция для СПО, журналы] : сайт. – URL: <https://www.book.ru/cat/576>.
2. ЭБС «**Znanium.com**» : [учебные, научные, научно-популярные материалы различных издательств, журналы] : сайт. – URL: <http://znanium.com/>.
3. ЭБС «**ЮРАЙТ**» : образовательная платформа : [учебники и учебные пособия издательства «ЮРАЙТ», медиа-материалы, тесты] : сайт. – URL: <https://urait.ru/>.
4. ЭБС «**Университетская библиотека ONLINE**» : [учебные, научные издания, первоисточники, художественные произведения различных

- издательств; журналы; мультимедийная коллекция, карты, онлайн-энциклопедии, словари] : сайт. – URL: [http://biblioclub.ru/index.php?page=main\\_ub\\_red](http://biblioclub.ru/index.php?page=main_ub_red).
5. **ЭБС издательства «Лань»** : [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы] : сайт. – URL: <http://e.lanbook.com>.
  6. **Национальная электронная библиотека** [включает Электронную библиотеку диссертаций РГБ] : федеральная государственная информационная система : сайт / Министерство культуры РФ. – URL: <https://rusneb.ru/> (доступ – в читальных залах библиотеки филиала).
  7. Электронная библиотека трудов учёных КубГУ : [ресурс свободного доступа] // Кубанский государственный университет : сайт.. – URL: <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>.
  8. **Math-Net.Ru** : общероссийский портал : информационная система доступа к научной информации по математике, физике, информационным технологиям и смежным наукам : [ресурс свободного доступа] / Математический институт им. В. А. Стеклова РАН. – URL: <http://www.mathnet.ru/>.
  9. **Научная электронная библиотека публикаций «eLibrary.ru»** : [российские научные журналы, труды конференций – большая часть представлена в свободном доступе; российский информационно-аналитический портал в области науки, технологии, медицины, образования – Российская национальная база данных научного цитирования (РИНЦ)] : сайт. – URL: <http://elibrary.ru>.
  10. **Базы данных компании «ИВИС»** : [российские научные журналы по вопросам педагогики и образования, экономики и финансов, информационным технологиям, экономике и предпринимательству, общественным и гуманитарным наукам, индивидуальные издания, Вестники МГУ, СПбГУ, статистические издания России и стран СНГ] : сайт. – URL: <https://eivis.ru/basic/details>.
  11. **КиберЛенинка** : научная электронная библиотека : [российские научные журналы] : сайт : [ресурс свободного доступа] . – URL: <http://cyberleninka.ru>.
  12. **Российское образование** : федеральный портал : [ресурс свободного доступа]. – URL: <http://www.edu.ru/>.
  13. **Российская электронная школа** : государственная образовательная платформа : сайт : [полный школьный курс уроков – ресурс свободного доступа]. – URL: <https://resh.edu.ru/>.
  14. **Наука.рф** : официальный сайт Десятилетия науки и технологий в России : [ресурс свободного доступа]. – URL: <https://наука.рф/>.
  15. **ГРАМОТА.РУ** : справочно-информационный интернет-портал : сайт : [ресурс свободного доступа] / функционирует при поддержке Министерства цифрового развития, связи и массовых коммуникаций РФ. – URL: <http://www.gramota.ru>.
  16. **Большая российская энциклопедия** : [электронная версия свободного

- доступа] : сайт / Министерство культуры РФ. – URL: <https://bigenc.ru/>.
17. **Энциклопедиум** : [Энциклопедии. Словари. Справочники] : сайт : [полнотекстовый ресурс свободного доступа] / издательство «Директ-Медиа» . – URL: <http://enc.biblioclub.ru/>.
18. **Электронный каталог [Кубанского государственного университета и филиалов]** // Электронная библиотека КубГУ : сайт. – URL: <http://212.192.134.46/MegaPro/Web/Home/About>.

## 6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Информатика» нацелена на формирование профессиональных компетенций, таких как способность демонстрации общенаучных базовых знаний естественных наук, математики, понимание основных фактов, концепций, принципов теорий.

Обучение студентов осуществляется по традиционной технологии (лекции, практики) с включением инновационных элементов.

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;

- запись ведётся очень быстро, чётко, по возможности короткими выражениями;

- не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить. Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;

- имена, даты, названия, выводы, определения записываются точно;

- следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине «Математика» проводятся по схеме:

- устный опрос по теории в начале занятия;

- работа в группах по разрешению различных ситуаций по теме занятия;

- решение практических задач;

- индивидуальные задания для подготовки к практическим занятиям.

Цель практического занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

- вербальные (преобладающим методом должно быть объяснение);

- практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

- библиотечные фонды филиала КубГУ;

- электронная библиотечная система «Университетская библиотека он-лайн»;
- электронная библиотечная система Издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения.

Просмотр книги начинается с титульного листа, следующего после обложки. На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание. Прочив предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. Чем чаще книга издаётся, тем большую ценность она представляет. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

- пометки, замечания, выделение главного;
- план, тезисы, выписки, цитаты;
- конспект, рабочая записка, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике.

Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка.

Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи- записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7-15 раз;
- конспект может быть как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:



– прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;

– на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;

– записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;

– конспектирование ведётся не с целью иметь определённые записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;

– после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обращаться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

– конспектируя книгу большого объёма, запись следует вести в общей тетради;

– на каждой странице слева оставляют поля шириной 25-30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;

– каждая страница тетради нумеруется;

– для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;

– при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.

– не следует сокращать имена и названия, кроме очень часто повторяющихся;

– в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Для написания реферата необходимо выбрать тему, согласовать ее с преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

## 7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

### 7.1. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	<b>Раздел 1. Информация и информационная деятельность человека</b>	ОК 1, ОК 2	Опрос, практическая работа
2.	<b>Раздел 2. Использование программных систем и сервисов</b>	ОК 2	Практическая работа
3.	<b>Раздел 3. Информационное моделирование</b>	ОК 1, ОК 2	Опрос, практическая работа
4.	<b>Прикладной модуль 2. Аналитика и визуализация данных на Python</b>	ОК 1, ОК 2, ПК 1.1	Опрос, практическая работа
5.	<b>Прикладной модуль 7. Введение в веб-разработку на языке JavaScript</b>	ОК 1, ОК 2, ПК 1.1	Опрос, практическая работа

### 7.2. Критерии оценки знаний

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, лабораторных работ, контрольных работ.

Общая/профессиональная компетенция	Раздел/Тема	Тип оценочных мероприятий
ОК 01	Тема 1.6 Тема 1.9 Тема 3.5	Тестирование
ОК 02	Тема 1.1 Тема 1.3 Тема 3.1 Тема 3.2 Тема 1.6 Тема 1.9	
ОК 01	Тема 1.7 Тема 1.8 Тема 2.2 Тема 3.4	Выполнение практических заданий
ОК 02	Тема 1.2 Тема 1.4 Тема 1.5 Тема 2.1 Тема 2.3 Тема 2.4 Тема 2.5 Тема 2.6 Тема 2.7 Тема 3.3 Тема 1.7 Тема 1.8 Тема 2.2 Тема 3.6 Тема 3.7 Тема 3.8 Тема 3.9 Тема 3.10 Тема 3.11 Тема 3.12 Тема 3.13	
ОК 02, ПК 1.1	Прикладные модули 2,7	Проектная работа
ОК 01, ОК 02, ПК 1.1	Все модули	Выполнение заданий дифференцированного зачета

#### Критерии оценки знаний студентов в целом по дисциплине:

**«отлично»** - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение

уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

«хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

### 7.3. Оценочные средств для проведения текущей аттестации

Текущий контроль проводится в форме:

- фронтальный опрос
- индивидуальный устный опрос
- письменный контроль
- практическая (лабораторная) работа
- защита выполненного задания,

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

## Примерные тестовые задания:

### Тема.1.1. Информация и информационные процессы

Ближе всего раскрывается смысл понятия «информация, используемая в бытовом общении» в утверждении:

- последовательность знаков некоторого алфавита;
- сообщение, передаваемое в форме знаков ли сигналов;
- сообщение, уменьшающее неопределенность знаний;
- сведения об окружающем мире, воспринимаемые человеком
- сведения, содержащиеся в научных теориях

Информацию, не зависящую от личного мнения, называют:

- достоверной;
- актуальной;
- объективной;
- полезной;
- понятной

Информацию, отражающую истинное положение дел, называют:

- понятной;
- достоверной;
- объективной;
- полной;
- полезной

Информацию, существенную и важную в настоящий момент, называют:

- полезной;
- актуальной;
- достоверной;
- объективной;
- полной

Информацию, дающую возможность, решать поставленную задачу, называют:

- понятной;
- актуальной;
- достоверной;
- полезной;
- полной

Информацию, достаточную для решения поставленной задачи, называют:

- полезной;
- актуальной;
- полной;
- достоверной;
- понятной

Информацию, изложенную на доступном для получателя языке, называют:

- полной;
- полезной;
- актуальной;
- достоверной;
- понятной

По способу восприятия информации человеком различают следующие виды информации:

- текстовую, числовую, символьную, графическую, табличную и пр.;
- научную, социальную, политическую, экономическую, религиозную пр.;
- быденную, производственную, техническую, управленческую;
- визуальную, звуковую, тактильную, обонятельную, вкусовую;
- математическую, биологическую, медицинскую, психологическую и пр.

По форме представления информации можно условно разделить на следующие виды:

- социальную, политическую, экономическую, техническую, религиозную и пр.;
- техническую, числовую, символьную, графическую, табличную пр.;
- быденную, научную, производственную, управленческую;
- визуальную звуковую, тактильную, обонятельную, вкусовую;
- математическую, биологическую, медицинскую, психологическую.

Укажите лишний объект с точки зрения способа представления информации:

- школьный учебник;
- фотография;
- телефонный разговор;
- картина;
- чертеж

По области применения информацию можно условно разделить на:

- текстовую и числовую;
- визуальную и звуковую;
- графическую и табличную;
- научную и техническую;
- тактильную и вкусовую

Какое из высказываний ложно?

- получение и обработка информации являются необходимыми условиями жизнедеятельности любого организма.
- для обмена информацией между людьми служат языки.
- информацию условно можно разделить на виды в зависимости от формы представления.
- процесс обработки информации техническими устройствами носит осмысленный характер.

- процессы управления – это яркий пример информационных процессов, протекающих в природе, обществе, технике.

Каждая знаковая система строится на основе:

- естественных языков, широко используемых человеком для представления информации;
- двоичной знаковой системы, используемой в процессах хранения, обработки и передачи информации в компьютере;
- определенного алфавита (набора знаков) и правил выполнения операций над знаками;
- правил синтаксиса алфавита.

Выбери из списка все языки, которые можно считать формальными языками:

- двоичная система счисления
- языки программирования
- кириллица
- китайский язык
- музыкальные ноты
- русский язык
- дорожные знаки
- код азбуки Морзе.

Производится бросание симметричной восьмигранной пирамидки. Какое количество информации мы получаем в зрительном сообщении о ее падении на одну из граней?

- 1 бит
- 1 байт
- 3 бит
- 3 бита.

Какое количество информации (с точки зрения алфавитного подхода) содержит двоичное число 10101001?

- 1 байта
- 2 байта
- 3 байта
- 3 бита.

Что из нижеперечисленного не является основой формирования информационной культуры?

- знания о законах функционирования информационной среды
- принцип узкой специализации
- знания об информационной среде
- умение ориентироваться в информационных потоках

Установите соответствие:

А Полнота	1 Язык понятен получателю
-----------	---------------------------

Б Достоверность	2 Достаточность для понимания, принятия решения
В Актуальность	3 Важность, значимость
Г Понятность	4 Неискажение истинного положения дел
Д Релевантность	5 Вовремя, в нужный срок

### Ключ к тесту

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Ответ	г	в	б	б	г	в	д	г	б	в	г	г	в	абджз	в	а	б	а2 б4 в3 г1 д3

## 1. Модельные примеры оценочных средств для проведения промежуточного и рубежного контроля по прикладному модулю 2 «Аналитика и визуализация данных на Python»

Выполнение контрольной работы предусмотрено после изучения темы 2.3.  
Работа со списками и словарями

1. Определите, что будет напечатано в результате выполнения программы:

```
z = 30
for n in range(10):
    if n < 0:
        z = z - 2 * n
    else:
        z = n - z
print(z)
```

Ответ: -35

2. Определите, что будет напечатано в результате выполнения программы:

```
a = 23
b = 4
while a > b:
    if a % 2 == 0:
        b = b + a
    else:
```

```
a = a - 2 * b + 1
print(b)
```

Ответ: 20

3. Определите, что будет напечатано в результате выполнения программы:

```
s = 0
m = 123
while m > 0:
    d = m % 10
    s = s + d
    m = m // 10
print(s)
```

Ответ: 6

4. Дан список из чисел и индекс элемента в списке  $k$ . Удалите из списка элемент с индексом  $k$ , сдвинув влево все элементы, стоящие правее элемента с индексом  $k$ . Программа получает на вход список, затем число  $k$ . Программа сдвигает все элементы, а после этого удаляет последний элемент списка при помощи метода `pop()` без параметров.

Программа должна осуществлять сдвиг непосредственно в списке, а не делать это при выводе элементов. Также нельзя использовать дополнительный список. Также не следует использовать метод `pop(k)` с параметром.

Решение:

```
a = [int(s) for s in input().split()]
k = int(input())
for i in range(k, len(a) - 1):
    a[i] = a[i + 1]
a.pop()
print(' '.join([str(i) for i in a]))
```

5. Предприятие производит оптовую закупку некоторых изделий А и В, на которую выделена определённая сумма денег. У поставщика есть в наличии партии этих изделий различных модификаций по различной цене. На выделенные деньги необходимо приобрести как можно больше изделий А независимо от модификации. Если у поставщика закончатся изделия А, то на оставшиеся деньги необходимо приобрести как можно больше изделий В. Известны выделенная для закупки сумма, а также количество и цена различных модификаций данных изделий у поставщика. Необходимо определить, сколько будет закуплено изделий В и какая сумма останется неиспользованной.

Входные данные.

Первая строка входного файла содержит два целых числа:  $N$  — общее количество партий изделий у поставщика и  $M$  — сумма выделенных на закупку денег (в рублях). Каждая из следующих  $N$  строк описывает одну партию и содержит два



целых числа (цена одного изделия в рублях и количество изделий в партии) и один символ (латинская буква А или В), определяющий тип изделия. Все данные в строках входного файла отделены одним пробелом.

В ответе запишите два целых числа: сначала количество закупленных изделий типа В, затем оставшуюся неиспользованной сумму денег.

Пример входного файла:

```
4 1000
30 8 A
50 12 B
40 14 A
30 60 B
```

В данном случае сначала нужно купить изделия А: 8 изделий по 30 рублей и 14 изделий по 40 рублей. На это будет потрачено 800 рублей. На оставшиеся 200 рублей можно купить 6 изделий В по 30 рублей. Таким образом, всего будет куплено 6 изделий В и останется 20 рублей. В ответе надо записать числа 6 и 20.

Решение

```
f = open('26.txt')
x, y = f.readline().split()
y = int(y)
for_B_price = []
for_B_kol = []
for i in f:
    if 'A' in i:
        a, b, c = i.split()
        y -= int(a) * int(b)
    else:
        a2, b2, c2 = i.split()
        for_B_price.append(int(a2))
        for_B_kol.append(int(b2))
mini = min(for_B_price)
index_mini = 0
for i in range(len(for_B_price)):
    if mini == for_B_price[i]:
        index_mini = i
kol_B = 0
while y > for_B_price[index_mini]:
    y -= for_B_price[index_mini]
    for_B_kol[index_mini] -= 1
    kol_B += 1
if for_B_kol[index_mini] == 0:
    for_B_price[index_mini] = 1000000000
    mini = min(for_B_price)
```

```

for i in range(len(for_B_price)):
    if mini == for_B_price[i]:
        index_mini = i
print(kol_B, y)

```

<b>Отметка</b>	<b>Число баллов, необходимое для получения отметки</b>
«3» (удов.)	Выполнены три задания (№ 1, 2, 3)
«4» (хорошо)	Выполнены четыре задания (№ 1, 2, 3, 4) Выполнены два задания (№ 4, 5)
«5» (отлично)	Выполнены все пять заданий

## **2. Модельные примеры оценочных средств для проведения промежуточного и рубежного контроля по прикладному модулю 7 «Введение в веб-разработку на языке JavaScript»**

### **Автоматизированная часть проектной работы**

1. Перейдите по адресу [https://node-server.online/r?id=\\_#student|37](https://node-server.online/r?id=_#student|37) и в последнем-нижнем поле напишите код функции `task(x)`, возвращающий `true`, только если в качестве аргумента `x` передана строка, которая начинается и кончается заглавными латинскими буквами, а между ними находятся только цифры в количестве от 4 до 8, иначе `false`.

2. Перейдите по <https://node-server.online/r?id=x#y|1> и в последнем-нижнем поле напишите код функции `task(x)`, возвращающей массив из числа `x`, переданного ей в качестве аргумента, и ещё 9 подряд идущих за `x` чисел, т.е. `[x, x + 1, ..., x + 9]`

3. Перейдите по адресу [https://node-server.online/r?id=\\_ - student|2](https://node-server.online/r?id=_ - student|2) и в последнем-нижнем поле напишите код функции `task(x)`, возвращающей сумму элементов массива, переданного ей в качестве аргумента `x`

4. Перейдите по адресу <https://node-server.online/r?id=x#student|27> и убедившись что в выпадающем списке выбрано `obj-001`, в последнем-нижнем поле напишите код функции `task(x)`, возвращающей JSON в виде правильной строки; у исходного объекта должно быть свойство `love` со строковым значением `javascript` и свойство `year` с числовым значением равным текущему году (4 цифры) и свойство `arg` со значением `x`.

5. Перейдите по адресу <https://node-server.online/r?id=x - student|28> и убедившись что в выпадающем списке выбрано `obj-002`, в последнем-нижнем поле напишите код функции `task(x)`, возвращающей массив собственных (не

унаследованных) перечислимых ключей объекта, переданного ей в качестве аргумента `x`

6. Перейдите по адресу <https://node-server.online/r?id=x - student|12> и убедившись что в выпадающем списке выбрано `dom-001`, в последнем-нижнем поле напишите код функции `task(x)`, возвращающей

созданный с помощью метода `createElement` новый DOM-элемент типа `span` у которого значением атрибута `id` является `x`

## Творческая часть проектной работы

*Посмотрите видеоуроки по начальным шагам работы с серверным JavaScript*

**<https://node-server.online/r/assets/server1-readfile.mp4>**

**<https://node-server.online/r/assets/server2-http.mp4>**

Перейдите по адресу

[https://node-server.online/r?id=\\_#student|20](https://node-server.online/r?id=_#student|20) и убедившись что в выпадающем списке выбрано `http-001`, в последнем-нижнем поле напишите код функции `task(x)`, возвращающей экземпляр `http`-сервера, созданный с помощью встроенного `Node.js`-модуля `http` (он будет доступен в функции). Этот сервер не должен быть запущен / слушать какой-либо порт. Используйте только `res.end` (не нужны заголовки или теги `HTML`). Он должен обрабатывать два маршрута:

(1) `/challenge` – в ответ на такой запрос он должен возвращать `x`

(2) `/api/rv/abc`

где `abc` – произвольная строка длиной не менее 1 символа, состоящая только из строчных латинских букв.

В ответ на такой запрос сервер должен возвращать перевёрнутую строку.

Например: <https://kodaktor.ru/api/rv/abc>

Для других маршрутов предусмотрен ответ `No`

Чтобы создать основу приложения, выполните следующие шаги:

1. Создайте новый проект:

`mkdir $(date +%Y%m%d_%H%M%S) && cd $_ && yarn init -y` или

`mkdir $(date +%Y%m%d_%H%M%S) && cd $_ && npm init -y`

(<https://kodaktor.ru/g/init>).

2. Установите инструмент `nodemon` для автоматизации перезапуска сценария и `moment` для работы с датой и временем: `yarn add --dev nodemon` или `npm i -D nodemon` и `yarn add moment` или `npm i moment`

```
"scripts" : {  
  "start": "nodemon"  
},
```

3. Установите настройки линтера и создайте нужный файл .eslintrc.

4. Создайте в папке проекта файл index.js с содержимым:

```
1  const http = require('http');  
2  const moment = require('moment');  
3  
4  http.createServer((req, res) => {  
5    res.end(moment().format('DD.MM.YYYY HH:mm:ss'));  
6  }).listen(4321);
```

5. Запустите сценарий yarn start и выполните curl localhost:4321.

6. Убедитесь, что в консоли отображается текущая дата и время.

7. Добавьте к проекту поддержку выдачи данных в формате JSON с выдачей соответствующего заголовка и кодировки UTF-8: <http://kodaktor.ru/gitcheckout.gif>

```
1  const http = require('http');  
2  const moment = require('moment');  
3  
4  http.createServer((req, res) => {  
5    res.setHeader('Content-Type', 'application/json; charset=utf-8');  
6    res.end(JSON.stringify({ date: moment().format('DD.MM.YYYY HH:mm:ss') }));  
7  }).listen(4321);
```

8. Перейдите по адресу localhost:4321 в браузере и убедитесь, что выдаётся ответ в формате JSON.

9. Осуществите рефакторинг кода так, чтобы коллбэк, отвечающий на запросы, явным образом указывался для события request:

```
1  const http = require('http');  
2  const moment = require('moment');  
3  
4  const server = http.createServer();  
5  server.listen(4321);  
6  server.on('request', (req, res) => {  
7    res.setHeader('Content-Type', 'application/json; charset=utf-8');  
8    res.end(JSON.stringify({ date: moment().format('DD.MM.YYYY HH:mm:ss') }));  
9  });
```

---

## Типовая процедура защиты результата проектной работы

При подготовке выступления для защиты проекта следует руководствоваться следующей дорожной картой презентации:

- обзор по теме;
- демонстрация в live-режиме (slides.com, например <http://slides.com/elizabethanatskaya-1/deck-2#/12> и др.);

- **ВЫВОДЫ;**
- поддержка в репозитории (ссылки на слайды / ресурсы / ...).

**Критерии оценивания.** Задание считается выполненным, если программа (сценарий) разработана и соответствует заданию, не содержит синтаксических ошибок, а также сопровождается репозиторием (в том числе, возможно, отчётом в форме слайдов). Часть заданий проверяется (или дополнительно проверяется) с помощью средств автоматизированной проверки. Веб-сценарии должны быть размещены на ресурсе, допускающем обращение через Интернет (веб-портфолио, например на Яндекс.Диске или с помощью GitHub Pages)

### Шкала итоговой оценки

Шкала	Критерии
зачтено	<p><b>обучающийся должен:</b> продемонстрировать знания изучаемых аспектов JavaScript в полном объёме: дать корректные ответы на 60% тестовых заданий и более, распознавать операторы и структуры данных JavaScript, исправлять ошибки в языковых конструкциях, быть способным корректно сформулировать определения, воспроизвести по запросу информацию о функциональности всех изучаемых средств разработки и проектирования веб-приложений, должно быть выполнено 85% и более заданий самостоятельной работы, представленных в виде корректно функционирующего веб-портфолио.</p>
не зачтено	<p><b>обучающийся:</b> не знает значительной части программного материала (ответил менее чем на 60% тестовых заданий); не владеет понятийным аппаратом дисциплины; не способен продемонстрировать знание ни одной синтаксической конструкции JavaScript и не способен оценить корректность работы веб-сценария; выполнил менее 85% заданий самостоятельной работы, которые не отражены в веб-портфолио.</p>

## 7.4. Оценочные средств для проведения промежуточной аттестации

### Промежуточная аттестация

Форма аттестации	Знания	Умения	Практический опыт (владеть)	Личные качества студента	Примеры оценочных средств
Дифференцированный зачет	Контроль знания базовых положений в области информатики	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области профессиональной деятельности	Оценка навыков логического мышления при решении задач в области профессиональной деятельности	Оценка способности грамотно и четко излагать ход решения задач в области профессиональной деятельности и аргументировать результаты	Задачи прилагаются

#### 7.4.1. Примерные вопросы для проведения промежуточной аттестации (Дифференцированный зачет)

- 1) Понятие «информация» как фундаментальное понятие современной науки.
- 2) Кодирование информации
- 3) Информация и информационные процессы.
- 4) Принципы построения компьютеров.
- 5) Принцип открытой архитектуры.
- 6) Аппаратное устройство компьютера.
- 7) Архитектура ЭВМ 5 поколения.
- 8) Основные характеристики компьютеров.
- 9) Программное обеспечение: классификация и его назначение, сетевое программное обеспечение.
- 10) Компьютерные сети их классификация.
- 11) Топологии локальных сетей.
- 12) Обмен данными.
- 13) Глобальная сеть Интернет.
- 14) IP-адресация.
- 15) Правовые основы работы в сети Интернет.
- 16) Информационная безопасность.
- 17) Защита информации.
- 18) Вредоносные программы.
- 19) Антивирусные программы. Безопасность в Интернете (сетевые угрозы, мошенничество).
- 20) Представление о компьютерных моделях.

- 21) Виды моделей.
- 22) Основные этапы компьютерного моделирования.
- 23) Структура информации.
- 24) Списки, графы, деревья.
- 25) Алгоритм построения дерева решений.
- 26) Структурированные типы данных.
- 27) Задачи поиска элемента с заданными свойствами.
- 28) Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов.
- 29) Базы данных как модель предметной области.
- 30) Таблицы и реляционные базы данных
- 31) Системы автоматизированного проектирования: история, назначение, примеры.
- 32) Автоматизированных Систем.

#### 7.4.2. Примерные задачи на дифференцированный зачет

Предлагается некоторая операция над двумя произвольными трехзначными десятичными числами:

- 1) записывается результат сложения старших разрядов этих чисел;
- 2) к нему дописывается результат сложения средних разрядов по такому правилу: если он меньше первой суммы, то полученное число приписывается к первому слева, иначе — справа;
- 3) итоговое число получают приписыванием справа к числу, полученному после второго шага, суммы значений младших разрядов исходных чисел.

Какое из перечисленных чисел могло быть построено по этому правилу?

- 141310
- 102113
- 101421
- 101413

Определите значение целочисленных переменных  $x$ ,  $y$  и  $t$  после выполнения фрагмента программы:

```
x := 5;  
y := 7;  
t := x;  
x := y mod x;  
y := t;
```

Определите значение переменной  $s$  после выполнения следующего фрагмента программы:

```
a := 6;  
b := 15;  
a := b - a*2;  
if a > b  
then c := a + b  
else c := b - a;
```

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2
2. умножь на 3

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, утраивает его. Запишите порядок команд в программе получения из 0 числа 28, содержащей не более 6 команд, указывая лишь номера команд.

Например, программа 21211 — это программа:

```
умножь на 3  
прибавь 2  
умножь на 3  
прибавь 2  
прибавь 2  
которая преобразует число 1 в 19.
```

Ответ \_\_\_\_\_

Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```
var i, y: integer;  
begin  
y := 0;  
for i := 1 to 4 do  
begin  
y := y * 10;  
y := y + i;  
end  
end.
```

Ответ \_\_\_\_\_

Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```
var y : real; i : integer;  
begin  
y := 0;  
i := 1;  
repeat
```



```
i := 2*i;  
y := y + i  
until i > 5;  
end.
```

Ответ \_\_\_\_\_

Определите значение переменной y, которое будет получено в результате выполнения следующей программы:

```
var y : real; i : integer;  
begin  
y := 0;  
i := 5;  
while i > 2 do  
begin  
i := i - 1;  
y := y + i * i  
end;  
end.
```

Ответ \_\_\_\_\_

В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do  
A[i]:= i + 1;  
for i:=0 to 10 do  
A[i]:= A[10-i];
```

Чему будут равны элементы этого массива?

10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0

11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

11, 10, 9, 8, 7, 6, 7, 8, 9, 10, 11

10, 9, 8, 7, 6, 5, 6, 7, 8, 9, 10

Все элементы двумерного массива A размером 5x5 равны 0. Сколько элементов массива после выполнения фрагмента программы будут равны 1?

```
for n:=1 to 5 do  
for m:=1 to 5 do  
A[n,m] := (m - n)*(m - n);
```

2

5

8

14

В программе описан одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

```
for i:=0 to 10 do
  A[i]:=i-1;
for i:=1 to 10 do
  A[i-1]:=A[i];
A[10]:=10;
```

Как изменятся элементы этого массива после выполнения фрагмента программы?

- все элементы, кроме последнего, окажутся равны между собой
- все элементы окажутся равны своим индексам
- все элементы, кроме последнего, будут сдвинуты на один элемент вправо
- все элементы, кроме последнего, уменьшатся на единицу

Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

Дан рекурсивный алгоритм:

```
procedure F(n: integer);
begin
  writeln(n);
  if n < 5 then begin
    F(n+3);
    F(n*3)
  end
end;
```

Найдите сумму чисел, которые будут выведены при вызове  $F(1)$ .

Ответ \_\_\_\_\_

Ниже записан рекурсивный алгоритм F:

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n-1)+F(n-2)+F(n-3)
  else
    F := n;
end;
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова  $F(6)$ ?

Ответ \_\_\_\_\_

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на четырех языках программирования

Алгоритмический язык	Паскаль	Python	C++
алг нач цел n, s s := 0 n := 95 нц пока s + n < 177 s := s + 10 n := n - 5 кц вывод n кон	<pre>var s, n: integer; begin   s := 0;   n := 95;   while s + n &lt; 177   do     begin       s := s + 10;       n := n - 5;     end;   writeln(n) end.</pre>	<pre>s = 0 n = 95 while s + n &lt; 177:   s = s + 10   n = n - 5 print(n)</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() {   int s = 0, n = 95;   while (s + n &lt; 177)   {     s = s + 10;     n = n - 5;   }   cout &lt;&lt; n &lt;&lt; endl;   return 0; }</pre>

Ответ \_\_\_\_\_

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на четырех языках программирования

Алгоритмический язык	Паскаль	Python	C++
алг нач цел n, s s := 0 n := 90 нц пока s + n < 145 s := s + 15 n := n - 5 кц вывод n кон	<pre>var s, n: integer; begin   s := 0;   n := 90;   while s + n &lt; 145   do     begin       s := s + 15;       n := n - 5;     end;   writeln(n) end.</pre>	<pre>s = 0 n = 90 while s + n &lt; 145:   s = s + 15   n = n - 5 print(n)</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() {   int s = 0, n = 90;   while (s + n &lt; 145)   {     s = s + 15;     n = n - 5;   }   cout &lt;&lt; n &lt;&lt; endl;   return 0; }</pre>

Ответ \_\_\_\_\_

Ниже приведена программа, записанная на пяти языках программирования

Алгоритмический язык	Паскаль	Python	C++
алг нач цел s, t, A ввод s ввод t ввод A если s > 10 или t > A то вывод "YES" иначе вывод "NO" все кон	<pre>var s, t, A: integer; begin   readln(s);   readln(t);   readln(A);   if (s &gt; 10) or (t &gt; A)   then     writeln("YES")   else writeln("NO") end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s &gt; 10) or (t &gt; A):   print("YES") else:   print("NO")</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main(){   int s, t, A;   cin &gt;&gt; s;   cin &gt;&gt; t;   cin &gt;&gt; A;   if (s &gt; 10) or (t &gt; A)   cout &lt;&lt; "YES" &lt;&lt;   endl;   else   cout &lt;&lt; "NO" &lt;&lt;   endl;   return 0; }</pre>

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел: (1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5). Укажите количество целых значений параметра A, при которых для указанных входных данных программа напечатает «YES» шесть раз.

Ответ \_\_\_\_\_

Ниже приведена программа, записанная на пяти языках программирования

Алгоритмический язык	Паскаль	Python	C++
алг нач цел s, t, A ввод s ввод t ввод A если s > 10 или t > A то вывод "YES" иначе вывод "NO"	<pre>var s, t, A: integer; begin   readln(s);   readln(t);   readln(A);   if (s &gt; 10) or (t &gt; A)   then     writeln("YES")   else writeln("NO") end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s &gt; 10) or (t &gt; A):   print("YES") else:   print("NO")</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main(){   int s, t, A;   cin &gt;&gt; s;   cin &gt;&gt; t;   cin &gt;&gt; A;   if (s &gt; 10) or (t &gt;</pre>

<p>все кон</p>			<p>A) cout &lt;&lt; "YES" &lt;&lt; endl; else cout &lt;&lt; "NO" &lt;&lt; endl; return 0; }</p>
--------------------	--	--	---

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел: (1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5). Укажите наименьшее целое значение параметра A, при котором для указанных входных данных программа напечатает «NO» семь раз.

Ответ \_\_\_\_\_

Ключ к тесту

## 8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Топология локальных сетей.

**Состав и конфигурация сетевой аппаратуры в зависимости от топологии сети.**

#### 1. Понятие топологии сети

*Общая схема соединения компьютеров в локальные сети называется топологией сети*

**Топология** - это физическая конфигурация сети в совокупности с ее логическими характеристиками. Топология - это стандартный термин, который используется при описании основной компоновки сети. Если понять, как используются различные топологии, то можно будет определить, какими возможностями обладают различные типы сетей.

Существует два основных типа топологий:

- физическая
- логическая

**Логическая топология** описывает правила взаимодействия сетевых станций при передаче данных.

**Физическая топология** определяет способ соединения носителей данных.

Термин "топология сети" характеризует физическое расположение компьютеров, кабелей и других компонентов сети. Топология физических связей может принимать разные «геометрические» формы, при этом существенным является не геометрическое расположение кабеля, а лишь наличие связи между узлами (замкнутость/незамкнутость, наличие центра и т.д.).

Топология сети обуславливает ее характеристики.

Выбор той или иной топологии влияет на:

- состав необходимого сетевого оборудования
- характеристики сетевого оборудования
- возможности расширения сети
- способ управления сетью

Конфигурация сети может быть или децентрализованной (когда кабель "обегает" каждую станцию в сети), или централизованной (когда каждая станция физически подключается к некоторому центральному устройству, распределяющему фреймы и пакеты между станциями). Примером централизованной конфигурации является звезда с рабочими станциями, располагающимися на концах ее лучей. Децентрализованная конфигурация похожа на цепочку альпинистов, где каждый имеет свое положение в связке, а все вместе соединены одной веревкой. Логические характеристики топологии сети определяют маршрут, проходимый пакетом при передаче по сети.

При выборке топологии нужно учитывать, чтобы она обеспечивала надежную и эффективную работу сети, удобное управление потоками сетевых данных. Желательно также, чтобы сеть по стоимости создания и сопровождения получилась недорогой, но в то же время оставались возможности для ее дальнейшего расширения и, желательно, для перехода к более высокоскоростным технологиям связи. Это непростая задача! Чтобы ее решить, необходимо знать, какие бывают сетевые топологии.

**По топологии связей различают:**

- сети с топологией "общая шина (шина)";

- сети с топологией "звезда";
- сети с топологией "кольцо"»;
- сети с древовидной топологией;
- сети со смешанной топологией

## **2. Базовые топологии сети**

Существует три базовые топологии, на основе которых строится большинство сетей.

- шина (bus)
- звезда (star)
- кольцо (ring)

"Шинной" называется топология, в которой компьютеры подключены вдоль одного кабеля.

"Звездой" называется топология, в которой компьютеры подключены к сегментам кабеля, исходящим из одной точки, или концентратора.

"Кольцом" называется топология, если кабель, к которому подключены компьютеры, замкнут в кольцо.

Хотя сами по себе базовые топологии несложны, в реальности часто встречаются довольно сложные комбинации, объединяющие свойства нескольких топологий.

### **2.1 Топология сети типа "шина" (bus)**

В этой топологии все компьютеры соединяются друг с другом одним кабелем. Каждый компьютер присоединяется к общему кабелю, на концах которого устанавливаются терминаторы. Сигнал проходит по сети через все компьютеры, отражаясь от конечных терминаторов.

Топология "шина" порождается линейной структурой связей между узлами. Аппаратно такая топология может быть реализована, например, путём установки на центральные компьютеры двух сетевых адаптеров. В целях предотвращения отражения сигнала на концах кабеля должны быть установлены терминаторы, поглощающие сигнал.

В сети с топологией "шина" компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов - аппаратных MAC-адресов. Чтобы понять процесс взаимодействия компьютеров по шине, нужно уяснить следующие понятия:

- передача сигнала
- отражение сигнала
- терминатор

#### **1. Передача сигнала**

Данные в виде электрических сигналов, передаются всем компьютерам сети; однако информацию принимает только тот, адрес которого соответствует адресу получателя, зашифрованному в этих сигналах. Причем в каждый момент времени только один компьютер может вести передачу. Так как данные в сеть передаются лишь одним компьютером, ее производительность зависит от количества компьютеров, подключенных к шине. Чем их больше, т.е. чем больше компьютеров, ожидающих передачи данных, тем медленнее сеть. Однако вывести прямую зависимость между пропускной способностью сети и количеством компьютеров в ней нельзя. Ибо, кроме числа компьютеров, на быстродействие сети влияет множество факторов, в том числе:

- характеристики аппаратного обеспечения компьютеров в сети

- частота, с которой компьютеры передают данные
- тип работающих сетевых приложений
- тип сетевого кабеля
- расстояние между компьютерами в сети

Шина - пассивная топология. Это значит, что компьютеры только "слушают" передаваемые по сети данные, но не перемещают их от отправителя к получателю. Поэтому, если один из компьютеров выйдет из строя, это не скажется на работе остальных. В активных топологиях компьютеры регенерируют сигналы и передают их по сети.

### 2. Отражение сигнала

Данные, или электрические сигналы, распространяются по всей сети - от одного конца кабеля к другому. Если не предпринимать никаких специальных действий, сигнал, достигая конца кабеля, будет отражаться и не позволит другим компьютерам осуществлять передачу. Поэтому, после того как данные достигнут адресата, электрические сигналы необходимо погасить.

### 3. Терминатор

Чтобы предотвратить отражение электрических сигналов, на каждом конце кабеля устанавливают заглушки (терминаторы, terminators), поглощающие эти сигналы. Все концы сетевого кабеля должны быть к чему-нибудь подключены, например к компьютеру или к баррел-коннектору - для увеличения длины кабеля. К любому свободному (неподключенному ни к чему) концу кабеля должен быть подсоединен терминатор, чтобы предотвратить отражение электрических сигналов.

#### Установка терминатора

Нарушение целостности сети может произойти, если разрыв сетевого кабеля происходит при его физическом разрыве или отсоединении одного из его концов. Возможна также ситуация, когда на одном или нескольких концах кабеля отсутствуют терминаторы, что приводит к отражению электрических сигналов в кабеле и прекращению функционирования сети. Сеть "падает". Сами по себе компьютеры в сети остаются полностью работоспособными, но до тех пор, пока сегмент разорван, они не могут взаимодействовать друг с другом.

У такой топологии сети есть достоинства и недостатки.

#### **Достоинства** топологии "шина":

- небольшое время установки сети
- дешевизна (требуется меньше кабеля и сетевых устройств)
- простота настройки
- выход из строя рабочей станции не отражается на работе сети

#### **Недостатки** топологии "шина":

- такие сети трудно расширять (увеличивать число компьютеров в сети и количество сегментов - отдельных отрезков кабеля, их соединяющих).
- поскольку шина используется совместно, в каждый момент времени передачу может вести только один из компьютеров.
- "шина" является пассивной топологией - компьютеры только "слушают" кабель и не могут восстанавливать затухающие при передаче по сети сигналы.
- надежность сети с топологией "шина" невысока. Когда электрический сигнал достигает конца кабеля, он (если не приняты специальные меры) отражается, нарушая работу всего сегмента сети.

Проблемы, характерные для топологии "шина", привели к тому, что эти сети сейчас уже практически не используются.



Топология сети типа "шина" известна как логическая топология Ethernet 10 Мбит/с.

## **2.2 Базовая топология сети типа "звезда" (star)**

При топологии "звезда" все компьютеры подключаются к центральному компоненту, именуемому концентратором (hub). Каждый компьютер подсоединяется к сети при помощи отдельного соединительного кабеля. Сигналы от передающего компьютера поступают через концентратор ко всем остальным.

В «звезде» всегда есть центр, через который проходит любой сигнал в сети. Функции центрального звена выполняют специальные сетевые устройства, причём передача сигнала в них может идти по-разному: в одних случаях устройство направляет данные всем узлам, кроме узла-отправителя, в других устройство анализирует, какому узлу предназначаются данные и направляет их только ему.

Эта топология возникла на заре вычислительной техники, когда компьютеры были подключены к центральному, главному, компьютеру.

**Достоинства** типологии "звезда":

- выход из строя одной рабочей станции не отражается на работе всей сети в целом
- хорошая масштабируемость сети
- лёгкий поиск неисправностей и обрывов в сети
- высокая производительность сети (при условии правильного проектирования)
- гибкие возможности администрирования

**Недостатки** типологии "звезда":

- выход из строя центрального концентратора обернётся неработоспособностью сети (или сегмента сети) в целом
- для прокладки сети зачастую требуется больше кабеля, чем для большинства других топологий
- конечное число рабочих станций в сети (или сегменте сети) ограничено количеством портов в центральном концентраторе.

Одна из наиболее распространённых топологий, поскольку проста в обслуживании. В основном используется в сетях, где носителем выступает кабель витая пара. UTP категория 3 или 5. (Категории кабеля «витая пара», которые нумеруются от 1 до 7 и определяют эффективный пропускаемый частотный диапазон. Кабель более высокой категории обычно содержит больше пар проводов и каждая пара имеет больше витков на единицу длины).

Топология типа "звезда" нашла свое отражение в технологии Fast Ethernet6.

## **2.3 Базовая топология сети типа "кольцо" (ring)**

При топологии "кольцо" компьютеры подключаются к кабелю, замкнутому в кольцо. Поэтому у кабеля просто не может быть свободного конца, к которому надо подключать терминатор. Сигналы передаются по кольцу в одном направлении и проходят через каждый компьютер. В отличие от пассивной топологии "шина", здесь каждый компьютер выступает в роли репитера (повторителя), усиливая сигналы и передавая их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть.

Функционирование замкнутой топологии «кольцо» основано на передаче маркера.

Маркер – пакет данных, разрешающий компьютеру передавать данные в сеть.

Маркер последовательно, от одного компьютера к другому, передается до тех пор, пока его не получит тот, который "хочет" передать данные. Компьютер, желающий начать передачу, «захватывает» маркер, изменяет его, помещает адрес получателя в данные и посылает их по кольцу получателю.

Данные проходят через каждый компьютер, пока не окажутся у того, чей адрес совпадает с адресом получателя, указанным в данных. После этого принимающий компьютер посылает передающему сообщение, где подтверждает факт приёма данных. Получив подтверждение, передающий компьютер создаёт новый маркер и возвращает его в сеть.

На первый взгляд, кажется, что передача маркера отнимает много времени, однако на самом деле маркер передвигается практически со скоростью света. В кольце диаметром 200 метров маркер может циркулировать с частотой 10 000 оборотов в секунду.

**Достоинства** топологии "кольцо":

- простота установки
- практически полное отсутствие дополнительного оборудования
- возможность устойчивой работы без существенного падения скорости передачи данных при интенсивной загрузке сети, поскольку использование маркера исключает возможность возникновения коллизий.

**Недостатки** топологии "кольцо":

- выход из строя одной рабочей станции, и другие неполадки (обрыв кабеля), отражаются на работоспособности всей сети
- сложность конфигурирования и настройки
- сложность поиска неисправностей

Наиболее широкое применение получила в оптоволоконных сетях. Используется в стандартах FDDI8, Token ring9.

### **3. Другие возможные сетевые топологии**

Реальные компьютерные сети постоянно расширяются и модернизируются. Поэтому почти всегда такая сеть является гибридной, т.е. ее топология представляет собой комбинацию нескольких базовых топологий. Легко представить себе гибридные топологии, являющиеся комбинацией "звезды" и "шины", либо "кольца" и "звезды".

#### **3.1 Топология сети типа "дерево" (tree)**

Топологию "дерево" (tree), можно рассматривать как объединение нескольких "звезд". Именно эта топология сегодня является наиболее популярной при построении локальных сетей.

Схема топологии сети типа "дерево"

В древовидной топологии есть корень дерева, от которого произрастают ветви и листья.

Дерево может быть активным или истинным и пассивным. При активном дереве в центрах объединения нескольких линий связи находятся центральные компьютеры, а при пассивном - концентраторы (хабы).

#### **3.2 Комбинированные топологии сети**

Довольно часто применяются комбинированные топологии, среди них наиболее распространены звездно-шинная и звездно-кольцевая.

В звездно-шинной (star-bus) топологии используется комбинация шины и пассивной звезды.

К концентратору подключаются как отдельные компьютеры, так и целые шинные сегменты. На самом деле реализуется физическая топология шина, включающая все компьютеры сети. В данной топологии может использоваться и несколько концентраторов, соединенных между собой и образующих так называемую магистральную, опорную шину. К каждому из концентраторов при этом подключаются отдельные компьютеры или шинные сегменты. В результате получается звездно-шинное дерево. Таким образом, пользователь может гибко комбинировать преимущества шинной и звездной топологий, а также легко изменять количество компьютеров, подключенных к сети. С точки зрения распространения информации данная топология равноценна классической шине.

В случае звездно-кольцевой (star-ring) топологии в кольцо объединяются не сами компьютеры, а специальные концентраторы, к которым в свою очередь подключаются компьютеры с помощью звездообразных двойных линий связи.

В действительности все компьютеры сети включаются в замкнутое кольцо, так как внутри концентраторов линии связи образуют замкнутый контур (как показано на рисунке 9). Данная топология дает возможность комбинировать преимущества звездной и кольцевой топологий. Например, концентраторы позволяют собрать в одно место все точки подключения кабелей сети. Если говорить о распространении информации, данная топология равноценна классическому кольцу.

### **3.3 "Сеточная" топология сети**

Наконец, следует упомянуть о сетчатой, или сеточной (mesh) топологии, в которой все либо многие компьютеры и другие устройства соединены друг с другом напрямую (рисунок 10).

Такая топология исключительно надежна - при обрыве любого канала передача данных не прекращается, поскольку возможно несколько маршрутов доставки информации. Сеточные топологии (чаще всего не полные, а частичные) используются там, где требуется обеспечить максимальную отказоустойчивость сети, например, при объединении нескольких участков сети крупного предприятия или при подключении к Интернету, хотя за это, конечно, приходится платить: существенно увеличивается расход кабеля, усложняется сетевое оборудование и его настройка.

В настоящее время, подавляющее большинство современных сетей используют топологию "звезда" или гибридную топологию, представляющую собой объединение нескольких "звезд" (например, топологию типа "дерево"), и метод доступа к среде передачи CSMA/CD (множественный доступ с контролем несущей и обнаружением столкновений).

## РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины УД.02 Информатика  
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины УД.02 Информатика соответствует требованиям ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины УД.02 «Информатика», примерной рабочей программы общеобразовательной дисциплины «Информатика» для профессиональных образовательных организаций, утвержденной на заседании Совета по оценке содержания и качества примерных рабочих программ общеобразовательного и социального-гуманитарного циклов среднего профессионального образования (протокол №14 от 30 ноября 2022 г.), и федеральному государственному образовательному стандарту среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (технологический профиль), утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936.

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертизы можно сделать заключение, что рабочая программа учебной дисциплины УД.02 Информатика по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Профессор кафедры математики,  
информатики, естественнонаучных  
и общетехнических дисциплин,  
доктор технических наук, профессор  
« » 20 г.



А.А. Маслак



## РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины УД.02 Информатика  
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины УД.02 Информатика соответствует требованиям ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины УД.02 «Информатика», примерной рабочей программы общеобразовательной дисциплины «Информатика» для профессиональных образовательных организаций, утвержденной на заседании Совета по оценке содержания и качества примерных рабочих программ общеобразовательного и социального-гуманитарного циклов среднего профессионального образования (протокол №14 от 30 ноября 2022 г.), и федеральному государственному образовательному стандарту среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (технологический профиль), утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936.

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертиза можно сделать заключение, что рабочая программа учебной дисциплины УД.02 Информатика по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Директор МБОУ ООШ № 31 имени  
полного кавалера ордена Славы В.Л. Храпко  
станции Петровской муниципального  
образования Славянский район



В.В. Васильев

« »

20 г.