

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Мухамбеталиева Алсу Сабиловна
Должность: Ведущий специалист отдела МКО
Дата подписания: 12.03.2026 02:34:40
Уникальный программный ключ:
a704735d5ef2b8c074250c71fd1f1631051f18e1

МИНИСТЕРСТВО СПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФИЗИЧЕСКОЙ КУЛЬТУРЫ, СПОРТА И ТУРИЗМА»

Кафедра информационных систем и диджитал спорта



УТВЕРЖДАЮ
Проректор по учебной работе и
цифровой трансформации
Панкова А.В.
«25» марта 2024 г.

Рабочая программа практики
Б2.О.01(У) Учебная практика: ознакомительная

основной профессиональной образовательной программы
по направлению подготовки
09.03.02 Информационные системы и технологии

Направленность (профиль) подготовки
Информационные системы и технологии

квалификация
бакалавр

Форма обучения очная

Авторы программы: к.ф.-м.н., доцент Мифтахов Р.Ф.

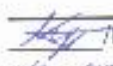
Программа обсуждена (или утверждена) на заседании кафедры «26» марта 2024 г.
Протокол № 8.

И.о. заведующего
кафедрой

Начальник
ИРЦ

Начальник управления по
образовательной деятельности

 Василев А.А.
«26» марта 2024 г.

 Кутенкова Л.М.
«26» марта 2024 г.

 Камалов Г.И.
«26» марта 2024 г.

Казань – 2024

1. ПРОХОЖДЕНИЕ ПРАКТИКИ НАПРАВЛЕНО НА ФОРМИРОВАНИЕ СЛЕДУЮЩИХ КОМПЕТЕНЦИЙ:

УК-1 - Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач;

ОПК-2 - Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

ОПК-5 - Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

РЕЗУЛЬТАТЫ ПРОХОЖДЕНИЯ ПРАКТИКИ

Формируемые компетенции	Индикаторы достижения компетенции	Формируемые знания, умения и навыки
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач;	УК-1.1. Знать: методики поиска, сбора и обработки информации; актуальные российские и зарубежные источники информации в сфере профессиональной деятельности; метод системного анализа. УК-1.2. Уметь: применять методики поиска, сбора и обработки информации; осуществлять критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач. УК-1.3 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации; методикой системного подхода для решения поставленных задач.	Знать: -способы определения задач в рамках поставленной цели, имеющихся ресурсов и ограничений; принципы, методы и средства решения, способы сбора, анализа и обработки данных, необходимых для решения поставленных задач с применением информационно-коммуникационных. Уметь: - решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий. Владеть: способностью обосновывать методологию решения задач рамках поставленной цели, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.	ОПК-2.1. Знает принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых для решения задач профессиональной деятельности. ОПК-2.2. Умеет выбирать для решения задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства, на основе понимания принципов их работы. ОПК-2.3. Владеет навыками использования современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	Знать: - принципы выбора программного обеспечения, в соответствии с целями и задачами профессиональной деятельности объекта практики. Уметь: - выбирать программные средства и информационные технологии для обеспечения работоспособности информационных систем объекта практики. Владеть: - Владеет навыками обеспечения функционирования информационных систем, используя необходимые программные средства.
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.	ОПК-5.1. Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем. ОПК-5.2. Умеет выполнять параметрическую настройку информационных и автоматизированных систем. ОПК-5.3. Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем.	Знать: - основные способы взаимодействия информационных систем; - инструменты организации взаимодействия информационных систем. Уметь: - создавать программно-техническую часть информационных систем, интегрированных в обеспечение деятельности объекта практики. Владеть: - программными средствами и информационными технологиями создания инфраструктуры информационных систем.

2. МЕСТО ПРАКТИКИ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ:

Б2.О.01(У) Учебная практика: ознакомительная практика относится к обязательной части Блока 2. Практика ОПОП ВО. В соответствии с учебным планом практика проходит на 1 курсе очной формы обучения (1, 2 семестры). Вид промежуточной аттестации: зачет (1 семестр), зачет (2 семестр).

3. ОБЪЁМ ПРАКТИКИ:

очная форма обучения

Название практики		Всего часов	Семестры							
			1	2	3	4	5	6	7	8
Выполнение индивидуальных заданий студентом		216	108	108						
Промежуточная аттестация (зачет/экзамен)		Зачет	Зачет	Зачет						
Общая трудоемкость	Часы	216								
	Зачетные единицы	6	3	3						

4.1. СОДЕРЖАНИЕ ПРАКТИКИ 1 СЕМЕСТР

№ п/п	Содержание раздела	Всего часов
1	<u>Организационное собрание, инструктаж по технике безопасности.</u> Разъяснение целей и задач практики, получение задания на проектно-технологическую практику, демонстрация формы отчетности. <u>Составление индивидуального плана практики.</u> Уточнение и согласование индивидуального задания на практику. Инструктаж по поиску информации в соответствии с целями и задачами практики на объекте практики. Задание определяется руководителем практики с учетом специфики объекта практики.	6
2	<u>Теоретическое представление проблемы исследования.</u> Подбор литературы по проблеме исследования. <u>Исследовательский этап</u> Изучение и анализ цифровой инфраструктуры объекта практики, функционирование отдельных элементов этой инфраструктуры. Виды профессиональной деятельности объекта практики, реализованные с помощью цифровых технологий. Особенности распределения задач и определения объемов работ. Сбор необходимой информации. Изучение перечня программного обеспечения используемого в организации и основных нормативно-правовых документов регламентирующих работу организации.	54
3	<u>Организация и проведение собственного исследования.</u> Наблюдение, сбор информации. Анализ полученной информации. Систематизация информации. Выполнение индивидуального задания	24

4	<u>Подготовка отчета по практике</u> , отражающего выполнение индивидуального задания. Защита отчета по практике.	24
Итого:		108

4.2. СОДЕРЖАНИЕ ПРАКТИКИ 2 СЕМЕСТР

№ п/п	Содержание раздела	Всего часов
1	<u>Организационное собрание, инструктаж по технике безопасности.</u> Разъяснение целей и задач практики, получение задания на проектно-технологическую практику, демонстрация формы отчетности. <u>Составление индивидуального плана практики.</u> Уточнение и согласование индивидуального задания на практику. Инструктаж по поиску информации в соответствии с целями и задачами практики на объекте практики. Задание определяется руководителем практики с учетом специфики объекта практики.	6
2	<u>Теоретическое представление проблемы исследования.</u> Подбор литературы по проблеме исследования. <u>Исследовательский этап</u> Изучение методов и средств проектирования информационных систем. Инструменты создания динамических и статических веб-сайтов. Изучение основ верстки веб-страниц и сайтов.	54
3	<u>Организация и проведение собственного исследования.</u> Наблюдение, сбор информации. Анализ полученной информации. Систематизация информации. Выполнение индивидуального задания	24
4	<u>Подготовка отчета по практике</u> , отражающего выполнение индивидуального задания. Защита отчета по практике.	24
Итого:		108

Формы и содержание отчетных документов по практике

Формами отчетности практики являются следующие документы:

- индивидуальный план (см. Приложение 2);
- дневник практиканта (см. Приложение 3);
- отчет (титульный лист см. Приложение 1) содержит анализ цифровой инфраструктуры объекта практики (университета). Особенности использования элементов цифровой инфраструктуры, распределения задач и определения объемов работ. Приложения со списком программных средств, которые используются в объекте практики. Пояснительные записки и демонстрационные материалы к выполненным заданиям.
- отзыв-характеристика (см. Приложение 4).

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Формируемые компетенции	Индикаторы достижения компетенции	Формируемые знания, умения и навыки
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач;	УК-1.1. Знать: методики поиска, сбора и обработки информации; актуальные российские и зарубежные источники информации в сфере профессиональной деятельности; метод системного анализа. УК-1.2. Уметь: применять методики поиска, сбора и обработки информации; осуществлять критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач. УК-1.3 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации; методикой системного подхода для решения поставленных задач.	Знать: -способы определения задач в рамках поставленной цели, имеющихся ресурсов и ограничений; принципы, методы и средства решения, способы сбора, анализа и обработки данных, необходимых для решения поставленных задач с применением информационно-коммуникационных. Уметь: - решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий. Владеть: способностью обосновывать методологию решения задач рамках поставленной цели, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.	ОПК-2.1. Знает принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых для решения задач профессиональной деятельности. ОПК-2.2. Умеет выбирать для решения задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства, на основе понимания принципов их работы. ОПК-2.3. Владеет навыками	Знать: - принципы выбора программного обеспечения, в соответствии с целями и задачами профессиональной деятельности объекта практики. Уметь: - выбирать программные средства и информационные технологии для обеспечения работоспособности информационных систем объекта практики. Владеть: - Владеет навыками

	использования современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	обеспечения функционирования информационных систем, используя необходимые программные средства.
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.	ОПК-5.1. Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем. ОПК-5.2. Умеет выполнять параметрическую настройку информационных и автоматизированных систем. ОПК-5.3. Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем.	Знать: - основные способы взаимодействия информационных систем; - инструменты организации взаимодействия информационных систем. Уметь: - создавать программно-техническую часть информационных систем, интегрированных в обеспечение деятельности объекта практики. Владеть: - программными средствами и информационными технологиями создания инфраструктуры информационных систем.

Учебно-методическое обеспечение самостоятельной работы студентов на учебной практике осуществляется свободным доступом студентов к библиотечным и электронным ресурсам университета по содержанию соответствующей программы практики, а также доступом к необходимой компьютерной технике, имеющейся в распоряжении кафедры и в лабораториях.

Для более глубокого изучения и анализа общих вопросов каждому студенту выдается индивидуальное задание в соответствии с содержанием практики и с учетом специфики будущей профессиональной деятельности.

ТИПОВЫЕ ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ 1 СЕМЕСТР

Задание 1. Создание электронного ресурса (модуля) в системе дистанционного образования университета (CMS Moodle). Ресурс должен содержать теоретическую часть, практическую часть (банк заданий), оценочные материалы в форме теста из 10 вопросов (5 теоретических вопросов и 5 практических вопросов).

Примерные темы:

1. Системы счисления.

Правильный вариант ответа.

Структура модуля:

1. Теоретическая часть

1. Понятие системы счисления:

Система счисления - это способ представления и записи чисел с помощью ограниченного набора символов (цифр). Система счисления определяет правила, по которым осуществляется представление и выполнение арифметических операций над

числами.

2. Основные виды систем счисления:

- Двоичная система счисления (основание 2) - использует только два символа: 0 и 1.
- Десятичная система счисления (основание 10) - использует 10 символов: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Шестнадцатеричная система счисления (основание 16) - использует 16 символов: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

3. Правила перевода чисел между системами счисления:

- Перевод из десятичной системы в другую:
- Делить число на основание новой системы, записывая остатки в обратном порядке.
- Перевод из другой системы в десятичную:
- Умножать каждую цифру на соответствующую степень основания и складывать результаты.

4. Арифметические операции в различных системах счисления:

- Сложение, вычитание, умножение и деление выполняются по тем же правилам, что и в десятичной системе, но с учетом основания системы.
- Например, в двоичной системе: - $1011 + 101 = 1100$ - $1101 - 10 = 1011$ - $101 \times 11 = 1111$ - $1010 \div 10 = 101$

2. Практическая часть (банк заданий)

1. Задания на перевод чисел из одной системы счисления в другую:

а) Перевести число 101101 из двоичной системы в десятичную:

Решение: $(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 32 + 0 + 8 + 4 + 0 + 1 = 45$

б) Перевести число 3AF16 из шестнадцатеричной системы в двоичную:

Решение: 3 = 0011, A = 1010, F = 1111 Таким образом, 3AF16 = 0011 1010 1111 2

2. Задания на выполнение арифметических операций в различных системах счисления:

а) Сложить числа 101011 и 11001 в двоичной системе:

Решение: $101011 + 11001 = 1000100$ б) Умножить числа 2A16 и 1F16 в шестнадцатеричной системе:

Решение: $2A \times 1F = 378$

3. Задания на сравнение чисел в разных системах счисления:

а) Сравнить числа 101010₂ и 52₁₀:

Решение: $101010_2 = 42_{10}$ $52_{10} > 42_{10}$, следовательно, $52_{10} > 101010_2$

б) Сравнить числа 7F16 и 12710:

Решение: $7F_{16} = 127_{10}$ $7F_{16} = 127_{10}$, следовательно, $7F_{16} = 127_{10}$

3. Оценочные материалы (10 вопросов)

Теоретические вопросы:

- Что такое система счисления?

Правильный вариант ответа.

Это способ представления чисел с помощью цифр. Она определяет правила записи и выполнения арифметических действий над числами.

- Назовите основные виды систем счисления.

Правильный вариант ответа.

Двоичная система счисления (основание 2, цифры 0 и 1) - Десятичная система счисления (основание 10, цифры от 0 до 9) - Шестнадцатеричная система счисления (основание 16, цифры от 0 до 9 и A-F)

- Как правильно выполнять перевод чисел между системами счисления?

Правильный вариант ответа.

Из десятичной системы в двоичную или шестнадцатеричную: - Деление числа на основание системы, запись остатков в обратном порядке. - Из двоичной или шестнадцатеричной системы в десятичную: - Умножение каждой цифры на соответствующую степень основания, сложение результатов.

- Какие арифметические операции можно выполнять в различных системах счисления?

Правильный вариант ответа.

- Сложение, вычитание, умножение и деление выполняются по тем же правилам, что и в десятичной системе.
- Необходимо учитывать "переносы" между разрядами в соответствии с основанием системы.

- Чем отличаются позиционные и непозиционные системы счисления?

Правильный вариант ответа.

Позиционные системы счисления (двоичная, десятичная, шестнадцатеричная) - значение цифры зависит от ее позиции в числе.

- Непозиционные системы счисления (римская)
 - значение цифры не зависит от ее позиции.
- В позиционных системах счисления можно выполнять арифметические операции, в непозиционных - только обозначение количества.

Практические вопросы:

- Переведите число 101011 из двоичной системы в десятичную.

Правильный вариант ответа.

Переведем число 101011 из двоичной системы в десятичную: $101011_2 = 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 32 + 0 + 8 + 0 + 2 + 1 = 43_{10}$

- Выполните сложение чисел 10101 (в двоичной) и 1101 (в двоичной).

Правильный вариант ответа.

Выполним сложение чисел 10101 (2) и 1101 (2): $10101 + 1101 = 100010_2$

- Представьте число 256 в шестнадцатеричной системе счисления.

Правильный вариант ответа.

Представим число 256 в шестнадцатеричной системе: $256_{10} = 100_{16}$

- Сравните числа 3A16 и 6D16, определите большее.

Правильный вариант ответа.

Сравним числа 3A16 и 6D16: $3A_{16} = 3 * 16^1 + 10 * 16^0 = 48 + 10 = 58_{10}$ $6D_{16} = 6 * 16^1 + 13 * 16^0 = 96 + 13 = 109_{10}$ Таким образом, $6D_{16} > 3A_{16}$.

- Вычтите число 25310 из числа 10001010 (в двоичной системе).

Правильный вариант ответа.

Вычтем число 25310 из числа 10001010 (2): $10001010_2 = 256 + 2 = 258_{10}$ $258_{10} - 253_{10} = 5_{10}$

2. Логические выражения.**Правильный вариант ответа.**

Теоретическая часть:

1. Введение в логические выражения:

Логические выражения - это выражения, которые могут иметь только два значения: истина (true) или ложь (false). Они используются для представления и обработки логических условий в программировании, математической логике и других областях.

2. Основные логические операции:

- Конъюнкция
 - логическое "и", истинна, если оба операнда истинны.
- Дизъюнкция - логическое "или", истинна, если хотя бы один операнд истинен.
- Отрицание - логическое "не", инвертирует значение операнда.
- Импликация - логическое "если..., то...", истинна, если левый операнд ложен или правый операнд истинен.
- Эквиваленция - логическое "равносильно", истинна, если оба операнда имеют одинаковое значение.

3. Приоритет логических операций:

1. Отрицание ($\neg$)
2. Конъюнкция ($\wedge$)

3. Дизъюнкция ($\vee$)
4. Импликация ($\rightarrow$)
5. Эквиваленция

4. Построение и анализ логических выражений:

Логические выражения строятся с использованием переменных, констант и логических операций. Необходимо следить за приоритетом операций и правильно расставлять скобки. Анализ логических выражений включает в себя определение их значений при различных комбинациях значений входных переменных.

5. Практические примеры использования логических выражений:

- Проверка условий в управляющих структурах (if-else, while, for)
- Реализация алгоритмов принятия решений
- Фильтрация и поиск данных
- Валидация вводимых пользователем значений
- Моделирование логических схем в электронике и информатике

Практическая часть (банк заданий):

1. Задания на построение простых логических выражений:

Пример: Постройте логическое выражение, которое будет истинно, если число x больше 5 и меньше 10.

Решение: $(x > 5) \wedge (x < 10)$

2. Задания на преобразование логических выражений:

Пример: Преобразуйте логическое выражение $(A \vee B) \wedge \neg C$ в эквивалентную форму.

Решение: $(A \vee B) \wedge \neg C \equiv A \wedge \neg C \vee B \wedge \neg C$

3. Задания на вычисление значений логических выражений:

Пример: Вычислите значение выражения $(A \wedge B) \vee (\neg A \wedge C)$, если $A = \text{true}$, $B = \text{false}$, $C = \text{true}$. Решение: $(A \wedge B) \vee (\neg A \wedge C) = (\text{true} \wedge \text{false}) \vee (\neg \text{true} \wedge \text{true}) = \text{false} \vee \text{true} = \text{true}$

4. Задания на анализ истинности логических выражений:

Пример: Определите, при каких значениях переменных A , B , C логическое выражение $(A \vee B) \wedge \neg C$ будет истинным.

Решение: $(A \vee B) \wedge \neg C$ будет истинно, когда: - $A = \text{true}$, $B = \text{true}$, $C = \text{false}$ - $A = \text{true}$, $B = \text{false}$, $C = \text{false}$ - $A = \text{false}$, $B = \text{true}$, $C = \text{false}$

5. Задания на применение логических выражений в различных контекстах:

Пример: Напишите логическое выражение, которое проверяет, что число x делится на 3 и 5 одновременно.

Решение: $(x \% 3 == 0) \wedge (x \% 5 == 0)$

Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Что такое логическое выражение?

Правильный вариант ответа.

Логическое выражение - это запись, состоящая из логических операторов и логических переменных (операндов), которая может принимать значения истина (true) или ложь (false).

- Перечислите основные логические операции.

Правильный вариант ответа.

Основные логические операции:

- Конъюнкция (AND, $\wedge$) - логическое "И"
- Дизъюнкция (OR, $\vee$) - логическое "ИЛИ"
- Отрицание (NOT, $\neg$) - логическое "НЕ"
- Импликация ($\rightarrow$) - логическая "если-то" связь
- Эквиваленция - логическое "равнозначность"
- Какова иерархия приоритета логических операций?

Правильный вариант ответа.

Иерархия приоритета логических операций:

1. Отрицание ($\neg$)
2. Конъюнкция ($\wedge$)
3. Дизъюнкция ($\vee$)
4. Импликация ($\rightarrow$)
5. Эквиваленция

- Как можно упростить сложное логическое выражение?

Правильный вариант ответа.

Для упрощения сложного логического выражения можно использовать различные законы алгебры логики, такие как:

- Законы поглощения
- Законы дистрибутивности
- Законы ассоциативности
- Другие законы и тождества
- Какие правила применяются при построении логических выражений?

Правильный вариант ответа.

Основные правила построения логических выражений:

- Использование скобок для задания приоритета операций
- Соблюдение иерархии приоритета логических операций
- Корректное использование логических операторов
- Учет значений логических переменных (true/false)

Практические вопросы (5 вопросов):

- Определите значение логического выражения "(A AND B) OR (NOT C)".

Правильный вариант ответа.

Значение этого выражения будет зависеть от значений переменных A, B и C: - Если A AND B является истинным, а C ложным, то выражение будет истинным. - Если A AND B ложные или C истинно, то выражение будет ложным.

- Постройте логическое выражение, которое истинно, если оба условия A и B верны.

Правильный вариант ответа.

Логическое выражение, которое истинно, если оба условия A и B верны, будет иметь вид: A AND B.

- Упростите логическое выражение "(A OR B) AND (NOT A) AND (NOT B)".

Правильный вариант ответа.

Упрощенное выражение будет ложным, так как конъюнкция (A OR B) и (NOT A) и (NOT B) всегда будет ложной.

- Определите тип логической операции, используемой в выражении "A -> B".

Правильный вариант ответа.

Логическая операция, используемая в выражении "A -> B", является импликацией. Импликация истинна, если либо A ложно, либо B истинно.

- Применив логические операции, сформулируйте условие, при котором студент допускается к экзамену.

Правильный вариант ответа.

Условие, при котором студент допускается к экзамену, можно сформулировать следующим логическим выражением: Посещаемость $\geq 80\%$ AND Успеваемость $\geq 60\%$

3. Решение задания №5 ЕГЭ по информатике.

Правильный вариант ответа.

1. Теоретическая часть:

1. Введение в задание №5 ЕГЭ по информатике:

Задание №5 ЕГЭ по информатике является практической частью экзамена, где от участника требуется написать короткий программный код, решающий поставленную

задачу. Это проверяет не только теоретические знания, но и практические навыки программирования.

2. Основные типы задач в задании №5:

- Работа с массивами (поиск, сортировка, обработка элементов)
- Обработка строк (поиск, преобразование, анализ)
- Логические задачи (использование логических операций и условных конструкций)
- Алгоритмические задачи (разработка алгоритмов решения задач)

3. Алгоритмы решения типовых задач:

Для каждого типа задач существуют устоявшиеся алгоритмы решения, которые необходимо знать и уметь применять.

Например:

- Сортировка массива: алгоритмы сортировки (пузырьковая, вставками, быстрая и др.)
- Поиск в строке: алгоритм Кнута-Морриса-Пракса, алгоритм Бойера-Мура
- Логические задачи: построение таблиц истинности, использование логических операций

4. Методы оптимизации решений:

Для повышения эффективности решений можно применять различные методы оптимизации:

- Использование эффективных алгоритмов
- Сокращение количества итераций и операций
- Применение дополнительных структур данных (словари, очереди и др.)
- Преобразование логических выражений в более компактные формы

5. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет: Задача: Дан массив целых чисел. Найти максимальную длину подмассива, в котором элементы строго возрастают. Решение:

Python

```
def max_len_increasing_subarray(arr):
    max_len = 0
    current_len = 1
    for i in range(1, len(arr)):
        if arr[i] > arr[i-1]:
            current_len += 1
        else:
            current_len = 1
        max_len = max(max_len, current_len)
    return max_len
```

2. Практическая часть (банк заданий):

1. Задачи на работу с массивами:

а) Поиск минимального/максимального элемента:

Задача: Найти минимальный элемент в массиве.

Решение (Python): `def find_min(arr): min_val = arr[0] for i in range(1, len(arr)): if arr[i] < min_val: min_val = arr[i] return min_val`

б) Сортировка массива:

Задача: Отсортировать массив в порядке возрастания.

Решение (Python): `def bubble_sort(arr): n = len(arr) for i in range(n): for j in range(0, n-i-1): if arr[j] > arr[j+1]: arr[j], arr[j+1] = arr[j+1], arr[j] return arr`

в) Обработка двумерных массивов:

Задача: Найти сумму элементов главной диагонали двумерного массива.

Решение (Python): `def sum_main_diagonal(matrix): n = len(matrix) total = 0 for i in range(n): total += matrix[i][i] return total`

2. Задачи на обработку строк:

а) Поиск подстроки:

Задача: Найти индекс первого вхождения подстроки в строке.

Решение (Python): `def find_substring(text, substring): for i in range(len(text) - len(substring) + 1): if text[i:i+len(substring)] == substring: return i return -1`

б) Замена символов:

Задача: Заменить все вхождения символа 'a' на символ 'b' в строке.

Решение (Python): `def replace_chars(text, old_char, new_char): return text.replace(old_char, new_char)` в) Преобразование регистра:

Задача: Перевести все символы строки в верхний регистр.

Решение (Python): `def to_upper(text): return text.upper()`

3. Логические задачи:

а) Применение логических операций:

Задача: Определить, является ли число простым.

Решение (Python): `def is_prime(n): if n <= 1: return False for i in range(2, int(n ** 0.5) + 1): if n % i == 0: return False return True`

б) Поиск истинных/ложных значений:

Задача: Найти количество истинных значений в списке логических переменных.

Решение (Python): `def count_true(values): count = 0 for value in values: if value: count += 1 return count`

в) Построение таблиц истинности:

Задача: Построить таблицу истинности для логического выражения $(A \wedge B)$

Решение:

A	B	$A \wedge B$
0	0	0
1	0	0
0	1	0
1	1	1

4. Алгоритмические задачи:

а) Работа с циклами и ветвлениями:

Задача: Вывести все числа Фибоначчи, меньшие заданного числа N.

Решение (Python): `def fibonacci_less_than(N): a, b = 0, 1 while a < N: print(a) a, b = b, a + b`

б) Рекурсивные алгоритмы:

Задача: Реализовать алгоритм Евклида для нахождения наибольшего общего делителя двух чисел.

Решение (Python): `def gcd(a, b): if b == 0: return a return gcd(b, a % b)`

в) Анализ и оптимизация алгоритмов:

Задача: Найти наиболее эффективный алгоритм поиска минимального/максимального элемента в массиве.

Решение (Python): `def find_min(arr): return min(arr)` # Встроенная функция min() имеет сложность $O(n)$

3. *Оценочные материалы (10 вопросов):*

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №5 ЕГЭ по информатике?

Правильный вариант ответа.

В задании №5 ЕГЭ по информатике могут встречаться следующие типы задач:

- Задачи на обработку массивов (поиск, сортировка, фильтрация, агрегирование данных)
- Задачи на обработку строк (поиск, замена, обработка символов)
- Задачи на разработку алгоритмов (в том числе рекурсивных)
- Задачи на анализ и преобразование логических выражений
- Назовите основные алгоритмы решения задач с массивами.

Правильный вариант ответа.

Основные алгоритмы решения задач с массивами:

- Линейный поиск
- Бинарный поиск
- Сортировка (например, сортировка пузырьком, сортировка выбором, быстрая сортировка)
- Фильтрация (поиск элементов, удовлетворяющих заданным условиям)
- Агрегирование данных (нахождение суммы, среднего, минимума, максимума)
- Какие методы оптимизации решений существуют?

Правильный вариант ответа.

Методы оптимизации решений:

- Использование эффективных алгоритмов (например, бинарный поиск вместо линейного)
- Сокращение объема обрабатываемых данных (фильтрация, предварительная обработка)
- Применение рекурсивных алгоритмов
- Использование дополнительных структур данных (например, словари, деревья)
- Параллельная обработка данных
- Что такое рекурсивный алгоритм и в каких случаях он применяется?

Правильный вариант ответа.

Рекурсивный алгоритм - это алгоритм, в котором одна или несколько из его операций вызывают сам алгоритм. Рекурсивные алгоритмы применяются, когда задача может быть разбита на более простые подзадачи, решение которых ведет к решению исходной задачи. Примеры: вычисление факториала, обход дерева, поиск в глубину.

- Как построить таблицу истинности для логического выражения?

Правильный вариант ответа.

Для построения таблицы истинности логического выражения необходимо:

1. Определить все входные логические переменные, участвующие в выражении.
2. Перебрать все возможные комбинации значений входных переменных.
3. Для каждой комбинации значений вычислить значение всего логического выражения.
4. Занести полученные значения в таблицу истинности.

Практические вопросы (5 вопросов):

- Напишите алгоритм поиска минимального элемента в массиве.

Правильный вариант ответа.

Алгоритм поиска минимального элемента в массиве:

1. Присвоить переменной min первый элемент массива.
2. Пройти по всем оставшимся элементам массива.
3. Если текущий элемент меньше min, то присвоить min этот элемент.
4. По окончании цикла min будет содержать минимальный элемент массива.

- Как заменить все вхождения подстроки в строке?

Правильный вариант ответа.

Для замены всех вхождений подстроки в строке можно использовать встроенные строковые функции языка программирования. Например, на Python: text = "Hello, world! Hello, Python!"

```
new_text = text.replace("Hello", "Hi")
```

```
print(new_text) # Выведет: "Hi, world! Hi, Python!"
```

- Определите значение логического выражения "(A OR B) AND (NOT C)".

Правильный вариант ответа.

Значение этого логического выражения будет зависеть от значений переменных A, B и C: - Если A или B истинны, а C ложно, то выражение будет истинным. - Во всех остальных случаях выражение будет ложным.

- Разработайте алгоритм, который находит сумму всех элементов массива, кратных 3.

Правильный вариант ответа.

Алгоритм нахождения суммы элементов массива, кратных 3:

1. Инициализировать переменную $sum = 0$.
 2. Пройти по всем элементам массива.
 3. Если текущий элемент кратен 3, то добавить его к sum .
 4. По окончании цикла sum будет содержать сумму всех элементов массива, кратных 3.
- Что будет выведено в результате работы следующего алгоритма: "for i = 1 to 10 do print i"?

Правильный вариант ответа.

В результате работы этого алгоритма будут выведены числа от 1 до 10 (включительно):
1 2 3 4 5 6 7 8 9 10

4. Решение задания №6 ЕГЭ по информатике.

Правильный вариант ответа.

1. Теоретическая часть:

1. Введение в задание №6 ЕГЭ по информатике:

Задание №6 ЕГЭ по информатике проверяет умение работать с информацией и данными, а также применять практические навыки использования различных инструментов и технологий для решения поставленных задач.

2. Основные типы задач в задании №6:

- Работа с файлами (чтение, запись, обработка данных)
- Обработка таблиц данных (поиск, фильтрация, сортировка, вычисления)
- Поиск информации в сети Интернет (использование поисковых систем, анализ результатов)
- Создание и редактирование электронных таблиц (формулы, функции, построение диаграмм)

3. Алгоритмы решения типовых задач:

Для каждого типа задач существуют устоявшиеся алгоритмы решения, которые необходимо знать и уметь применять.

Например:

- Чтение и запись данных в файлы (использование соответствующих библиотек и функций)
- Поиск, фильтрация и сортировка таблиц данных (SQL-запросы, методы сортировки)
- Использование поисковых машин и анализ результатов поиска (ключевые слова, коэффициенты релевантности)
- Применение формул и функций в электронных таблицах (математические, статистические, логические)

4. Методы обработки и анализа данных:

Для повышения эффективности решений можно применять различные методы обработки и анализа данных:

- Агрегирование и группировка данных
- Визуализация данных (построение диаграмм, графиков)
- Статистический анализ (вычисление средних, медиан, дисперсий)
- Выявление закономерностей и трендов в данных

5. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: В текстовом файле содержатся сведения о температуре воздуха за несколько дней. Необходимо найти среднюю температуру за этот период.

Решение (Python):

```
with open('temperature.txt', 'r') as file:
    temperatures = [float(line.strip())
for line in file]
avg_temp = sum(temperatures) / len(temperatures)
print(f'Средняя температура: {avg_temp:.2f} °C')
```

2. Практическая часть (банк заданий):

1. Задачи на работу с файлами:

а) Чтение и запись текстовых файлов:

Задача: Подсчитать количество слов в текстовом файле.

Решение (Python): `def count_words(file_path): with open(file_path, 'r') as file: text = file.read() words = text.split() return len(words)`

б) Обработка структурированных данных (CSV, XML):

Задача: Прочитать данные из CSV-файла и вывести сумму значений в заданном

столбце. Решение (Python): `import csv def sum_column(file_path, column_index): total = 0 with open(file_path, 'r') as file: reader = csv.reader(file) next(reader) # Пропустить заголовок for row in reader: total += float(row[column_index]) return total`

в) Анализ и преобразование файловой системы:

Задача: Найти все файлы с расширением ".txt" в указанной директории.

Решение (Python): `import os def find_txt_files(directory): txt_files = [] for filename in os.listdir(directory): if filename.endswith('.txt'): txt_files.append(os.path.join(directory, filename)) return txt_files`

2. Задачи на обработку таблиц данных:

а) Поиск, фильтрация и сортировка данных:

Задача: Найти всех сотрудников с заработной платой выше среднего.

Решение (Python): `import csv def find_high_earners(file_path): high_earners = [] with open(file_path, 'r') as file: reader = csv.DictReader(file) salaries = [float(row['Salary']) for row in reader] avg_salary = sum(salaries) / len(salaries) file.seek(0) # Вернуться к началу файла reader = csv.DictReader(file) for row in reader: if float(row['Salary']) > avg_salary: high_earners.append(row) return high_earners`

б) Вычисление статистических показателей:

Задача: Найти медианную зарплату сотрудников.

Решение (Python): `import csv def find_median_salary(file_path): salaries = [] with open(file_path, 'r') as file: reader = csv.DictReader(file) for row in reader: salaries.append(float(row['Salary'])) salaries.sort() n = len(salaries) if n % 2 == 0: return (salaries[n/2-1] + salaries[n/2]) / 2 else: return salaries[n/2]`

3. Задачи на поиск информации в сети Интернет:

а) Использование поисковых систем:

Задача: Найти 5 наиболее релевантных результатов поиска по заданному запросу.

Решение (Python): `from googlesearch import search def search_google(query, num_results=5): results = [] for url in search(query, num_results=num_results): results.append(url) return results`

б) Работа с API интернет-сервисов:

Задача: Получить текущий курс валюты с использованием API сервиса.

Решение (Python): `import requests def get_currency_rate(currency_code): url = f'https://api.exchangeratesapi.io/latest?base={currency_code}' response = requests.get(url) data = response.json() return data['rates']`

4. Задачи на создание и редактирование электронных таблиц:

а) Форматирование ячеек и таблиц:

Задача: Отформатировать ячейки таблицы для отображения денежных значений.

Решение (Excel):

1. Выделите ячейки, содержащие денежные значения.

2. На вкладке "Главная" в группе "Число" выберите формат "Денежный".

3. При необходимости настройте дополнительные параметры форматирования (количество десятичных знаков, символ валюты и т.д.).

б) Использование функций и формул:

Задача: Вычислить сумму продаж по каждому товару в таблице.

Решение (Excel):

1. Выделите столбец, содержащий значения продаж.

2. На вкладке "Формулы" нажмите кнопку "Автосумма" или введите формулу " =SUM(ячейки) ".
3. Скопируйте формулу вниз, чтобы получить общую сумму продаж для каждого товара.

в) Построение графиков и диаграмм:

Задача: Построить столбчатую диаграмму, отображающую продажи по месяцам.

Решение (Excel):

1. Выделите данные, которые будете использовать для построения диаграммы.
2. На вкладке "Вставка" выберите тип диаграммы "Столбчатая".
3. Настройте параметры диаграммы (заголовок, оси, легенду и т.д.) по своему усмотрению.

3. Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №6 ЕГЭ по информатике?

Правильный вариант ответа.

В задании №6 ЕГЭ по информатике могут встречаться следующие типы задач:

- Задачи на обработку данных из файлов (чтение, запись, поиск, фильтрация)
- Задачи на работу с электронными таблицами (поиск, фильтрация, применение формул)
- Задачи на использование API интернет-сервисов для получения и обработки данных
- Задачи на применение алгоритмов и методов обработки данных
- Какие методы обработки данных используются в задачах с файлами?

Правильный вариант ответа.

Основные методы обработки данных в задачах с файлами:

- Последовательное чтение и запись данных в файл
- Поиск данных в файле по заданным критериям
- Фильтрация данных из файла (отбор по условиям)
- Агрегирование данных (подсчет, суммирование, группировка)
- Сортировка данных из файла
- Как осуществляется поиск и фильтрация данных в электронных таблицах?

Правильный вариант ответа.

Поиск и фильтрация данных в электронных таблицах:

- Поиск по значениям ячеек с помощью функций ПОИСК(), НАЙТИ()
- Фильтрация данных с помощью автофильтра или расширенного фильтра
- Использование логических функций (И(), ИЛИ(), НЕ()) для сложных условий фильтрации
- Что такое API интернет-сервисов и как они применяются в задачах?

Правильный вариант ответа.

API (Application Programming Interface) интернет-сервисов - это стандартизированный интерфейс для взаимодействия с онлайн-сервисами. Они применяются в задачах для:

- Получения данных с интернет-ресурсов (погода, курсы валют, новости и т.д.)
- Отправки данных и запросов на серверы интернет-сервисов
- Интеграции функциональности сторонних сервисов в свои приложения
- Какие функции и формулы могут быть использованы в электронных таблицах?

Правильный вариант ответа.

Широкий спектр функций и формул, применимых в электронных таблицах:

- Математические и статистические функции (СУММ(), СРЗНАЧ(), МИН(), МАКС() и др.)
- Логические функции (ЕСЛИ(), И(), ИЛИ(), НЕ())
- Текстовые функции (ДЛИНА(), ЛЕВСИМВ(), ПРАВСИМВ(), СЦЕПИТЬ() и др.)
- Дата и время (СЕГОДНЯ(), ДЕНЬ(), МЕСЯЦ(), ГОД() и т.д.)
- Ссылочные функции (ПОИСКПОЗ(), ИНДЕКС(), ПОИСКПОЗ() и др.)

Практические вопросы (5 вопросов):

- Напишите алгоритм, который считывает содержимое текстового файла и подсчитывает количество строк.

Правильный вариант ответа.

Алгоритм подсчета количества строк в текстовом файле:

1. Открыть файл для чтения.
2. Инициализировать переменную count = 0.
3. Считывать по одной строке из файла, пока не достигнут конец файла.
4. Для каждой прочитанной строки увеличивать count на 1.
5. По окончании чтения файла count будет содержать количество строк.
6. Закрыть файл.

- Как отсортировать данные в таблице по возрастанию значений в столбце "Количество"?

Правильный вариант ответа.

Для сортировки данных в таблице по возрастанию значений в столбце "Количество" можно использовать функцию сортировки в электронной таблице.

Например, в Excel это можно сделать следующим образом:

1. Выделить весь диапазон данных.
2. Перейти во вкладку "Данные" и выбрать "Сортировка".
3. В окне сортировки выбрать столбец "Количество" и направление сортировки "По возрастанию".
4. Применить сортировку.

- Используя поисковую систему, найдите информацию о средней температуре в городе N за последний месяц.

Правильный вариант ответа.

Для поиска информации о средней температуре в городе N за последний месяц можно использовать поисковую систему, например Google. Ввести в строке поиска что-то вроде "средняя температура в городе N за [последний месяц]" и просмотреть результаты. Также можно обратиться к специализированным метеорологическим сайтам или приложениям, которые могут предоставлять такую информацию.

- Создайте диаграмму, отображающую распределение оценок студентов по предмету.

Правильный вариант ответа.

Для создания диаграммы, отображающей распределение оценок студентов по предмету, можно использовать возможности электронных таблиц, таких как Excel или Google Sheets. Основные шаги:

1. Подготовить данные об оценках студентов в виде таблицы.
2. Выделить диапазон данных.
3. Перейти во вкладку "Вставка" и выбрать подходящий тип диаграммы, например гистограмму или круговую диаграмму.
4. Настроить параметры диаграммы (заголовок, подписи осей, легенду и т.д.).
5. Разместить диаграмму на листе.

- Вычислите сумму значений в ячейках диапазона A1:A10 с помощью формулы в электронной таблице.

Правильный вариант ответа.

Для вычисления суммы значений в ячейках диапазона A1:A10 в электронной таблице можно использовать формулу: =SUM(A1:A10) Эта формула суммирует все значения в указанном диапазоне ячеек и выводит результат в ячейку, где она расположена.

5. Решение задания №8 ЕГЭ по информатике.

Рекомендации к плану темы.

1. Теоретическая часть:

1. Введение в задание №8 ЕГЭ по информатике:

Задание №8 ЕГЭ по информатике является наиболее сложной частью экзамена, которая проверяет глубокие знания и навыки в области информатики. Это комплексное задание, требующее применения различных подходов и методов для его решения.

2. Основные типы задач в задании №8:

- Обработка информации, представленной в табличной форме (поиск, фильтрация, агрегирование данных)
- Анализ и преобразование графических изображений (работа с пикселями, фильтрация, обработка)
- Решение комбинаторных задач (подсчет количества комбинаций, вероятностные расчеты)
- Работа с двоичными числами и системами счисления (перевод, арифметические операции)

3. Методы обработки табличных данных:

- Использование SQL-запросов (SELECT, WHERE, GROUP BY, ORDER BY и др.)
- Применение статистических функций (SUM, AVG, MAX, MIN и др.)
- Построение сводных таблиц и диаграмм

4. Алгоритмы преобразования графических объектов:

- Алгоритмы обработки пикселей (яркость, контрастность, фильтрация)
- Алгоритмы масштабирования и преобразования изображений
- Алгоритмы распознавания объектов на изображениях

5. Принципы решения комбинаторных задач:

- Использование формул для подсчета количества комбинаций (размещения, перестановки, сочетания)
- Применение правила умножения и дерева вариантов
- Решение задач на вероятность с использованием комбинаторики

6. Правила перевода между системами счисления:

- Переход от десятичной системы к двоичной, восьмеричной, шестнадцатеричной и обратно
- Выполнение арифметических операций в различных системах счисления

7. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Дана таблица, содержащая данные о продажах. Необходимо найти продукт, объем продаж которого максимален.

Решение (SQL): `SELECT Product, SUM(Quantity * Price) AS TotalSales FROM Sales GROUP BY Product ORDER BY TotalSales DESC LIMIT 1;`

2. Практическая часть (банк заданий):

1. Задачи на обработку табличных данных:

а) Поиск, фильтрация и сортировка данных:

Задача: Найти 3 самых продаваемых товара в таблице "Продажи".

Решение (SQL): `SELECT Product, SUM(Quantity) AS TotalSales FROM Sales GROUP BY Product ORDER BY TotalSales DESC LIMIT 3;`

б) Вычисление статистических показателей:

Задача: Найти среднюю, минимальную и максимальную цену товаров в таблице "Товары". Решение (SQL): `SELECT AVG(Price) AS AvgPrice, MIN(Price) AS MinPrice, MAX(Price) AS MaxPrice FROM Products;`

в) Построение диаграмм и графиков:

Задача: Построить столбчатую диаграмму, отображающую продажи по категориям товаров. Решение (Python + Matplotlib): `import matplotlib.pyplot as plt import pandas as pd # Загрузка данных из таблицы "Продажи" sales_data = pd.read_csv('sales.csv') # Группировка по категории и подсчет общих продаж sales_by_category = sales_data.groupby('Category')['Quantity'].sum().reset_index() # Построение столбчатой диаграммы plt.figure(figsize=(10, 6)) sales_by_category.plot(kind='bar', x='Category',`

```
y='Quantity') plt.title('Продажи по категориям товаров') plt.xlabel('Категория')  
plt.ylabel('Общий объем продаж') plt.show()
```

2. Задачи на преобразование графических изображений:

а) Масштабирование, отражение и поворот объектов:

Задача: Уменьшить изображение в 2 раза, отразить его по вертикали и повернуть на 90 градусов. Решение (Python + Pillow):

```
from PIL import Image  
def transform_image(image_path):  
    image = Image.open(image_path) # Уменьшение  
    # изображения в 2 раза  
    resized_image = image.resize((image.width // 2, image.height // 2))  
    # Отражение по вертикали  
    flipped_image = resized_image.transpose(Image.FLIP_LEFT_RIGHT)  
    # Поворот на 90 градусов  
    rotated_image = flipped_image.rotate(90)  
    return rotated_image
```

б) Кодирование и декодирование изображений:

Задача: Сжать изображение с использованием алгоритма сжатия без потерь.

Решение (Python + Pillow):

```
from PIL import Image  
def compress_image(image_path, output_path):  
    image = Image.open(image_path) # Сохранение изображения в формате  
    # PNG (сжатие без потерь)  
    image.save(output_path, optimize=True, quality=100)
```

3. Комбинаторные задачи:

а) Перебор всех возможных вариантов:

Задача: Найти все возможные комбинации из 3 букв, используя буквы "А", "Б" и "В".

Решение (Python):

```
def find_combinations(letters):  
    combinations = []  
    for a in letters:  
        for b in letters:  
            for c in letters:  
                combination = a + b + c  
                combinations.append(combination)  
    return combinations  
print(find_combinations(['А', 'Б', 'В'])) # Выводит все 27 комбинаций
```

б) Применение правил комбинаторики:

Задача: Сколько различных 4-значных чисел можно составить из цифр 1, 2, 3 и 4, если каждая цифра может использоваться только один раз?

Решение: Для решения данной задачи можно применить правило перестановок:
Количество различных 4-значных чисел = $4! = 4 * 3 * 2 * 1 = 24$.

Задачи на работу с двоичными числами:

а) Перевод между системами счисления:

Задача: Переведите число 1101011_2 в десятичную систему счисления.

Решение: $1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 64 + 32 + 0 + 8 + 0 + 2 + 1 = 107$

б) Выполнение арифметических операций в двоичной системе:

Задача: Сложите числа 10101_2 и 11011_2 .

Решение: $10101_2 + 11011_2 = 101000_2$

3. *Оценочные материалы (10 вопросов):*

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №8 ЕГЭ по информатике?

Правильный вариант ответа.

В задании №8 ЕГЭ по информатике могут встречаться следующие типы задач:

- Задачи на обработку табличных данных (сортировка, фильтрация, поиск, агрегирование)
- Задачи на преобразование графических объектов (масштабирование, поворот, отражение)
- Задачи на комбинаторику (перебор вариантов, применение правил комбинаторики)
- Задачи на перевод между системами счисления (двоичная, десятичная, шестнадцатеричная)
- Как можно отсортировать табличные данные по возрастанию/убыванию?

Правильный вариант ответа.

Для сортировки табличных данных по возрастанию или убыванию можно использовать следующие методы:

- В электронных таблицах

- использовать встроенные функции сортировки, выбрав нужное направление.
- При программировании - реализовать алгоритмы сортировки, такие как сортировка пузырьком, выбором, быстрая сортировка и др.
- Какие алгоритмы преобразования используются для работы с графическими объектами?

Правильный вариант ответа.

Основные алгоритмы преобразования графических объектов:

- Масштабирование (изменение размера) - использование коэффициентов масштабирования.
- Поворот - применение матриц поворота.
- Отражение - использование матриц отражения. –
- Сдвиг (перемещение) - применение матриц сдвига.
- Что такое комбинаторика и какие правила применяются при решении комбинаторных задач?

Правильный вариант ответа.

Комбинаторика - это раздел математики, изучающий методы подсчета количества возможных вариантов. Основные правила комбинаторики:

- Правило суммы (для последовательных выборов)
- Правило произведения (для параллельных выборов)
- Формулы для перестановок, сочетаний, размещений
- Как осуществляется перевод между различными системами счисления?

Правильный вариант ответа.

Перевод между системами счисления выполняется по следующим алгоритмам:

- Из двоичной в десятичную: перемножить каждую цифру на 2 в соответствующей степени и сложить.
- Из десятичной в двоичную: последовательно делить на 2, записывая остатки.
- Между десятичной и шестнадцатеричной: использовать таблицу соответствия цифр.
- Между любыми системами: сначала перевести в десятичную, а затем в нужную систему.

Практические вопросы (5 вопросов):

- Вычислите среднее значение в столбце "Количество" таблицы данных.

Правильный вариант ответа.

Для вычисления среднего значения в столбце "Количество" необходимо:

1. Сложить все значения в столбце "Количество".
2. Разделить полученную сумму на количество значений в столбце.

Например, в формуле электронной таблицы это будет выглядеть так:
 $\text{=СРЗНАЧ(Столбец_Количество)}$

- Определите, как изменится изображение после поворота на 90 градусов по часовой стрелке.

Правильный вариант ответа.

При повороте изображения на 90 градусов по часовой стрелке произойдут следующие изменения:

- Ширина и высота изображения поменяются местами (ширина станет высотой, а высота - шириной).
- Все элементы изображения повернутся на 90 градусов по часовой стрелке относительно их первоначального положения.
- Сколько различных последовательностей из 3 букв можно составить из букв А, Б и В?

Правильный вариант ответа.

Для решения данной комбинаторной задачи можно применить правило произведения:
 Количество букв = 3 Длина последовательности = 3. Следовательно, количество различных последовательностей из 3 букв = $3 * 3 * 3 = 27$.

- Переведите число 101101 из двоичной системы счисления в десятичную.

Правильный вариант ответа.

Для перевода числа 101101 из двоичной системы в десятичную необходимо:

1. Записать двоичное число в столбик, учитывая значение каждой позиции.
 2. Умножить каждую цифру на соответствующую степень 2.
 3. Сложить полученные произведения. Таким образом, 101101 в десятичной системе будет равно: $12^5 + 02^4 + 12^3 + 12^2 + 02^1 + 12^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45$.
- Выполните логическую операцию "И" для двоичных чисел 10010 и 11001.

Правильный вариант ответа.

Для выполнения логической операции "И" над двоичными числами 10010 и 11001 необходимо поразрядно сравнить каждую позицию: 10010 И 11001 = 10000
Результатом операции "И" будет двоичное число 10000.

6. Решение задания №10 ЕГЭ по информатике.

Рекомендации к плану темы.**1. Теоретическая часть:**

1. Введение в задание №10 ЕГЭ по информатике:

Задание №10 ЕГЭ по информатике является самым сложным и комплексным заданием в экзамене. Оно проверяет глубокие знания и практические навыки в области программирования, анализа алгоритмов и решения прикладных задач.

2. Основные типы задач в задании №10:

- Анализ алгоритмов (оценка сложности, поиск ошибок, внесение изменений)
- Программирование на языках высокого уровня (реализация алгоритмов, работа с данными)
- Работа с данными в различных форматах (чтение, обработка, вывод)
- Решение практических задач (оптимизация, адаптация к условиям)

3. Методы анализа и оптимизации алгоритмов:

- Определение временной и пространственной сложности алгоритмов
- Применение техник программирования для повышения эффективности (избегание дублирования, использование структур данных, параллелизация и т.д.)
- Анализ и модификация существующих алгоритмов для решения поставленных задач

4. Подходы к разработке программного кода:

- Использование структурированного программирования (модульность, функции, циклы, условные операторы)
- Применение объектно-ориентированного подхода (классы, наследование, инкапсуляция)
- Соблюдение принципов чистого кода (читаемость, именование, комментирование)

5. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Написать программу, которая считывает из текстового файла список чисел, находит их среднее арифметическое и выводит результат.

Решение (Python): `with open('numbers.txt', 'r') as file: numbers = [float(num) for num in file.read().split()] average = sum(numbers) / len(numbers) print(f'Среднее арифметическое: {average:.2f}')`

2. Практическая часть (банк заданий):

1. Анализ алгоритмов:

Задача: Дан алгоритм, вычисляющий факториал числа. Определите его временную сложность. Алгоритм: `def factorial(n): if n == 0: return 1 else: return n * factorial(n-1)`

Решение: Данный алгоритм использует рекурсивный вызов функции factorial, что приводит к экспоненциальной временной сложности $O(n)$. Это связано с тем, что для вычисления факториала n требуется n рекурсивных вызовов.

2. Программирование на языках высокого уровня:

Задача: Написать программу, которая считывает из файла список чисел, находит их

минимальное и максимальное значения, а затем выводит их.

Решение (Python): `with open('numbers.txt', 'r') as file: numbers = [int(num) for num in file.read().split()] min_value = min(numbers) max_value = max(numbers) print(f'Минимальное значение: {min_value}') print(f'Максимальное значение: {max_value}')`

3. Работа с данными в различных форматах:

Задача: Написать программу, которая считывает данные из CSV-файла, находит среднее значение в заданном столбце и выводит результат.

Решение (Python): `import csv def calculate_average(file_path, column_index): total = 0 count = 0 with open(file_path, 'r') as file: reader = csv.reader(file) next(reader) # Пропустить заголовок for row in reader: value = float(row[column_index]) total += value count += 1 average = total / count return average file_path = 'data.csv' column_index = 2 # Индекс столбца, для которого нужно вычислить среднее average_value = calculate_average(file_path, column_index) print(f'Среднее значение: {average_value:.2f}')`

4. Решение практических задач:

Задача: Написать программу, которая находит в текстовом файле все слова, начинающиеся с заданной буквы, и выводит их в алфавитном порядке.

Решение (Python): `def find_words(file_path, start_letter): words = [] with open(file_path, 'r') as file: text = file.read().lower().split() for word in text: if word.startswith(start_letter.lower()): words.append(word) words.sort() return words file_path = 'text.txt' start_letter = 'a' result_words = find_words(file_path, start_letter) for word in result_words: print(word)`

3. *Оценочные материалы (10 вопросов):*

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №10 ЕГЭ по информатике?

Правильный вариант ответа.

В задании №10 ЕГЭ по информатике могут встречаться следующие типы задач:

- Задачи на разработку и анализ алгоритмов
- Задачи на определение временной и пространственной сложности алгоритмов
- Задачи на применение управляющих конструкций в программировании
- Задачи на чтение и преобразование данных из различных форматов
- Задачи на интеграцию знаний из разных разделов информатики
- Что такое сложность алгоритма и как ее можно оценить?

Правильный вариант ответа.

Сложность алгоритма - это мера трудоемкости выполнения алгоритма в зависимости от размера входных данных. Различают две основные характеристики сложности:

- Временная сложность - оценивает количество операций, выполняемых алгоритмом.
- Пространственная сложность - оценивает объем памяти, необходимый для выполнения алгоритма.

Сложность алгоритма обычно обозначается с помощью "большого O"-нотации и может иметь значения $O(1)$, $O(\log n)$, $O(n)$, $O(n^2)$ и т.д.

- Какие основные управляющие конструкции используются в языках программирования высокого уровня?

Правильный вариант ответа.

Основные управляющие конструкции в языках программирования высокого уровня:

- Следование (последовательное выполнение команд)
- Ветвление (if-else, switch/case)
- Циклы (for, while, do-while)
- Подпрограммы (функции, процедуры)
- Обработка исключительных ситуаций (try-catch)
- Как происходит чтение и преобразование данных из различных форматов?

Правильный вариант ответа.

Чтение и преобразование данных из различных форматов может включать следующие шаги:

- Определение формата данных (текстовый, бинарный и т.д.)
- Использование соответствующих библиотек или функций для чтения данных
- Парсинг или разбор данных в нужную структуру (например, строки в числа)
- Преобразование данных из одного формата в другой
- Обработка ошибок, связанных с некорректными или неподдерживаемыми форматами
- Что такое интеграция знаний в информатике и зачем она применяется?

Правильный вариант ответа.

Интеграция знаний в информатике - это применение и комбинирование знаний из разных разделов информатики для решения более сложных задач. Это может включать:

- Использование алгоритмов и структур данных для обработки данных
 - Применение языков программирования для реализации алгоритмов
 - Интеграция программного обеспечения, баз данных и сетевых технологий
 - Использование методов искусственного интеллекта для решения прикладных задач
- Интеграция знаний позволяет создавать более эффективные и комплексные решения, объединяя различные аспекты информатики.

Практические вопросы (5 вопросов):

- Определите выходные данные для следующего алгоритма: "x = 5, y = 10, z = x + y, print(z)".

Правильный вариант ответа.

Выходные данные: 15
Объяснение: - Переменной x присваивается значение 5. - Переменной y присваивается значение 10. - Переменной z присваивается результат суммы $x + y$, то есть $5 + 10 = 15$. - Команда `print(z)` выводит значение переменной z, которое равно 15.

- Найдите ошибку в представленном алгоритме сортировки массива.

Правильный вариант ответа.

Ошибка в том, что в алгоритме сортировки не предусмотрен выход из внутреннего цикла, когда массив уже отсортирован. Таким образом, алгоритм будет выполнять лишние итерации, даже если массив уже отсортирован.

- Напишите программу на языке Python, которая вычисляет сумму элементов массива.

Правильный вариант ответа.

Программа на Python: `arr = [5, 2, 8, 1, 9] total = 0 for num in arr: total += num print(total)` # Выведет 25

- Как отобразить данные из CSV-файла в виде столбчатой диаграммы?

Правильный вариант ответа.

Для отображения данных из CSV-файла в виде столбчатой диаграммы можно использовать библиотеку Matplotlib в Python.

Например: `import pandas as pd import matplotlib.pyplot as plt # Чтение данных из CSV-файла data = pd.read_csv('data.csv') # Построение столбчатой диаграммы data.plot(kind='bar') plt.title('Диаграмма') plt.xlabel('Категория') plt.ylabel('Значение') plt.show()`

- Разработайте алгоритм и программу для расчета площади и периметра прямоугольника по заданным сторонам.

Правильный вариант ответа.

Алгоритм:

1. Ввести значения сторон прямоугольника (длина и ширина).
2. Вычислить площадь прямоугольника по формуле: $\text{площадь} = \text{длина} * \text{ширина}$.
3. Вычислить периметр прямоугольника по формуле: $\text{периметр} = 2 * (\text{длина} + \text{ширина})$.
4. Вывести результаты вычислений.

Программа на Python: `length = float(input("Введите длину прямоугольника: ")) width = float(input("Введите ширину прямоугольника: ")) area = length * width perimeter = 2 *`

```
(length + width) print("Площадь прямоугольника:", area) print("Периметр  
прямоугольника:", perimeter)
```

7. Решение задания №12 ЕГЭ по информатике.

Рекомендации к плану темы.

1. Теоретическая часть:

1. Введение в задание №12 ЕГЭ по информатике:

Задание №12 ЕГЭ по информатике является наиболее сложным в экзамене. Оно проверяет глубокие знания и практические навыки в области алгоритмизации, программирования, использования ИКТ-технологий и работы с базами данных.

2. Основные типы задач в задании №12:

- Разработка и анализ алгоритмов (оценка сложности, поиск ошибок, внесение изменений)
- Программирование на языках высокого уровня (реализация алгоритмов, работа с данными)
- Решение прикладных задач с использованием ИКТ (обработка данных, визуализация, оптимизация)
- Работа с базами данных (извлечение, фильтрация, агрегирование данных)

3. Методы разработки и анализа алгоритмов:

- Использование структурированного программирования (модульность, функции, циклы, условные операторы)
- Применение объектно-ориентированного подхода (классы, наследование, инкапсуляция)
- Определение временной и пространственной сложности алгоритмов
- Применение техник программирования для повышения эффективности (избегание дублирования, использование структур данных, параллелизация и т.д.)

4. Использование ИКТ для решения практических задач:

- Обработка и анализ данных (статистические расчеты, визуализация, построение моделей)
 - Автоматизация рутинных процессов (скрипты, макросы, интеграция систем)
 - Разработка и тестирование программного обеспечения
5. Основы работы с базами данных:
- Структура реляционных баз данных (таблицы, столбцы, строки)
 - Язык SQL для работы с данными
 - Оптимизация запросов для повышения производительности

6. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Написать программу, которая считывает данные о продажах товаров из базы данных, находит самый популярный товар и выводит его название.

Решение (Python + SQL):

```
import sqlite3  
def find_top_product(database_path):  
    connection =  
    sqlite3.connect(database_path)  
    cursor = connection.cursor() # Выполнение SQL-запроса для  
    поиска самого популярного товара  
    cursor.execute(""" SELECT Product, SUM(Quantity)  
    AS TotalSales FROM Sales GROUP BY Product ORDER BY TotalSales DESC LIMIT 1;  
    """)  
    result = cursor.fetchone()  
    top_product = result[0]  
    connection.close()  
    return top_product  
database_path = 'sales.db'  
top_product = find_top_product(database_path)  
print(f'Самый популярный товар: {top_product}')
```

2. Практическая часть (банк заданий):

1. Задачи на разработку и анализ алгоритмов:

а) Построение блок-схем и псевдокода:

Задача: Разработать алгоритм и представить его в виде блок-схемы, который находит наибольшее общее делитель (НОД) двух чисел.

Решение: Алгоритм Евклида для нахождения НОД:

1. Ввести два числа a и b .
2. Если $b = 0$, то вывести a как НОД и завершить алгоритм.
3. Присвоить $a = b$, $b = a \% b$.
4. Перейти к шагу 2.

б) Определение выходных данных для заданных алгоритмов:

Задача: Дан следующий алгоритм: Начало Ввести число N Если $N \leq 1$, то вывести "Ошибка" и завершить Иначе $K = 2$ Пока $K \leq \sqrt{N}$ Если $N \% K = 0$, то Вывести K $N = N / K$ Иначе $K = K + 1$ Конец Пока Вывести N Конец Если Конец Определите, какие числа будут выведены на экран, если ввести число 100.

Решение: Если ввести число 100, то алгоритм выведет следующие числа: 2 2 5

Объяснение: Алгоритм проверяет делимость числа 100 на различные простые числа, начиная с 2. Он находит все простые делители числа 100 и выводит их по порядку.

в) Оценка эффективности и сложности алгоритмов:

Задача: Дан следующий алгоритм: Функция Факториал(N): Если $N = 0$ или $N = 1$, то Вернуть 1 Иначе Вернуть $N * \text{Факториал}(N-1)$ Конец Функции Оцените временную сложность данного алгоритма.

Решение: Временная сложность данного рекурсивного алгоритма вычисления факториала равна $O(N)$. Это связано с тем, что для вычисления факториала числа N требуется выполнить N рекурсивных вызовов.

2. Задачи на программирование:

а) Написание программ на языках высокого уровня:

Задача: Написать программу на Python, которая считывает из текстового файла список чисел, находит их среднее арифметическое и выводит результат.

Решение: `with open('numbers.txt', 'r') as file: numbers = [float(num) for num in file.read().split()] average = sum(numbers) / len(numbers) print(f'Среднее арифметическое: {average:.2f}')`

б) Реализация структур данных и алгоритмических конструкций:

Задача: Написать программу на C++, которая реализует стек (Stack) с операциями push, pop и peek.

Решение: `#include <iostream> #include <vector> class Stack { private: std::vector<int> elements; public: void push(int value) { elements.push_back(value); } void pop() { if (!elements.empty()) { elements.pop_back(); } else { std::cout << "Стек пуст" << std::endl; } } int peek() { if (!elements.empty()) { return elements.back(); } else { std::cout << "Стек пуст" << std::endl; return -1; } } }; int main() { Stack myStack; myStack.push(10); myStack.push(20); std::cout << "Верхний элемент: " << myStack.peek() << std::endl; myStack.pop(); std::cout << "Верхний элемент: " << myStack.peek() << std::endl; return 0; }`

3. Прикладные задачи с использованием ИКТ:

а) Решение задач в области моделирования, обработки данных, визуализации:

Задача: Написать программу на Python, которая строит круговую диаграмму, отображающую продажи по категориям товаров.

Решение: `import matplotlib.pyplot as plt import pandas as pd # Загрузка данных из таблицы "Продажи" sales_data = pd.read_csv('sales.csv') # Группировка по категории и подсчет общих продаж sales_by_category = sales_data.groupby('Category')['Quantity'].sum().reset_index() # Построение круговой диаграммы plt.figure(figsize=(8, 6)) sales_by_category.plot(kind='pie', y='Quantity', labels=sales_by_category['Category'], autopct='%1.1f%%') plt.title('Продажи по категориям товаров') plt.axis('equal') plt.savefig('sales_pie_chart.png')`

б) Применение электронных таблиц, баз данных, мультимедийных технологий:

Задача: Создать простую базу данных, содержащую таблицу "Продукты", и написать SQL-запрос для выборки всех продуктов с ценой выше среднего.

Решение: `import sqlite3 # Создание и подключение к базе данных conn = sqlite3.connect('products.db') cursor = conn.cursor() # Создание таблицы "Продукты" cursor.execute(""" CREATE TABLE IF NOT EXISTS Products (id INTEGER PRIMARY KEY, name TEXT, price REAL) """) # Вставка данных в таблицу products = [('Яблоки', 50.0), ('Бананы', 30.0), ('Апельсины', 70.0), ('Груши', 40.0), ('Клубника', 80.0)] cursor.executemany("INSERT INTO Products (name, price) VALUES (?, ?)", products) # Выборка продуктов с ценой выше среднего cursor.execute("SELECT name, price FROM Products WHERE price > (SELECT AVG(price) FROM Products)") for row in cursor.fetchall(): print(f"{row[0]} - {row[1]}") # Сохранение изменений и закрытие соединения conn.commit() conn.close()`

3. Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №12 ЕГЭ по информатике?

Правильный вариант ответа.

Задачи на алгоритмизацию и программирование

- Задачи на работу с табличными данными (электронные таблицы)

- Задачи на анализ и обработку текстовой информации

- Что такое блок-схема и как она используется для представления алгоритмов?

Правильный вариант ответа.

Блок-схема - это графическое представление алгоритма, использующее стандартные геометрические фигуры для обозначения различных действий. Блок-схемы помогают наглядно продемонстрировать последовательность шагов алгоритма.

- Каковы основные принципы структурного и объектно-ориентированного программирования?

Правильный вариант ответа.

Структурное программирование:

- Модульность

- Использование базовых управляющих конструкций (следование, ветвление, цикл)

- Абстракция данных

Объектно-ориентированное программирование:

- Инкапсуляция

- Наследование

- Полиморфизм

- Абстракция

- Как ИКТ-инструменты могут быть применены для решения практических задач?

Правильный вариант ответа.

ИКТ-инструменты могут применяться для решения практических задач в различных областях, например:

- Офисные приложения (текстовые редакторы, электронные таблицы) для обработки данных

- Специализированное программное обеспечение для моделирования, расчетов, визуализации

- Интернет-сервисы для поиска, хранения и обмена информацией

- Системы управления базами данных для структурированного хранения и обработки данных

- Что такое база данных и какие операции можно выполнять с данными?

Правильный вариант ответа.

База данных - это организованная структура для хранения данных. Основные операции с базами данных:

- Создание, модификация и удаление таблиц

- Добавление, изменение и удаление записей

- Выборка данных с применением фильтров и сортировки

- Связывание таблиц, выполнение запросов
- Резервное копирование и восстановление данных

Практические вопросы (5 вопросов):

- Определите выходные данные для следующего алгоритма: "x = 5, y = 10, z = x + y, if z > 10 then print(z) else print(x)".

Правильный вариант ответа.

Выходные данные для алгоритма: x = 5 y = 10 z = 15 Поскольку z = 15 > 10, будет выведено значение z, то есть 15.

- Напишите программу на языке Python, которая вычисляет площадь круга по заданному радиусу.

Правильный вариант ответа.

Программа на Python для вычисления площади круга по заданному радиусу: `import math`

`radius = float(input("Введите радиус круга: "))`

`area = math.pi * radius ** 2`

`print("Площадь круга:", area)`

- Как можно использовать электронные таблицы для анализа и визуализации данных?

Правильный вариант ответа.

Использование электронных таблиц для анализа и визуализации данных:

- Организация данных в табличном формате
- Применение различных функций и формул для вычислений
- Построение графиков, диаграмм и других визуальных представлений данных
- Сортировка, фильтрация и группировка данных
- Создание сводных таблиц и отчетов
- Спроектируйте простую базу данных для учета успеваемости студентов.

Правильный вариант ответа.

Простая база данных для учета успеваемости студентов:

Таблица "Студенты": - ID_студента - Фамилия - Имя – Группа Таблица "Оценки": - ID_оценки - ID_студента - Предмет - Оценка - Дата Связь между таблицами: один студент может иметь множество оценок (отношение "один-ко-многим").

- Составьте SQL-запрос, который выбирает все записи из таблицы "Товары" с ценой, превышающей 1000 рублей.

Правильный вариант ответа.

SQL-запрос для выборки записей из таблицы "Товары" с ценой, превышающей 1000 рублей: `SELECT *`

`FROM Товары`

`WHERE Цена > 1000;`

8. Решение задания №16 ЕГЭ по информатике.

Рекомендации к плану темы.

1. Теоретическая часть:

1. Введение в задание №16 ЕГЭ по информатике:

Задание №16 ЕГЭ по информатике проверяет знания и умения учащихся в области кодирования и декодирования информации, разработки компьютерных моделей, обработки и анализа данных, а также использования электронных таблиц.

2. Основные типы задач в задании №16:

- Решение задач на кодирование и декодирование информации
- Реализация компьютерных моделей
- Применение методов обработки и анализа данных
- Выполнение вычислений с использованием электронных таблиц

3. Принципы кодирования и декодирования информации:

- Представление различных типов данных (текст, числа, изображения) в двоичном коде

- Применение алгоритмов сжатия и кодирования информации (например, алгоритм Хаффмана)

- Выполнение операций с двоичными числами (сложение, умножение, сдвиг)

4. Методы разработки и анализа компьютерных моделей:

- Построение моделей с использованием различных подходов (математические, имитационные, информационные)

- Анализ входных данных, параметров и выходных результатов моделей

- Оценка адекватности и эффективности компьютерных моделей

5. Алгоритмы обработки и анализа данных:

- Сортировка, фильтрация, агрегирование данных

- Поиск статистических характеристик (среднее, медиана, мода)

- Визуализация данных (диаграммы, графики)

6. Применение функций и формул в электронных таблицах:

- Использование различных типов функций (математические, логические, текстовые)

- Создание сложных формул с использованием ссылок на ячейки

- Организация вычислений с применением абсолютных и относительных ссылок

7. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Написать программу, которая считывает из текстового файла список чисел, находит их среднее арифметическое и выводит результат.

Решение (Python): `with open('numbers.txt', 'r') as file: numbers = [float(num) for num in file.read().split()] average = sum(numbers) / len(numbers) print(f'Среднее арифметическое: {average:.2f}')`

2. Практическая часть (банк заданий):

1. Задачи на кодирование и декодирование информации:

а) Кодирование и декодирование текстовой информации:

Задача: Написать программу, которая кодирует текст в соответствии с кодом Хаффмана и выводит полученный код.

Решение (Python): `from collections import Counter def huffman_encode(text): # Подсчет частоты символов freq = Counter(text) # Создание дерева Хаффмана nodes = [(freq[char], char) for char in freq] while len(nodes) > 1: # Объединение двух узлов с наименьшей частотой a, b = min(nodes[:2], key=lambda x: x[0]) nodes = nodes[2:] + [(a + b, [a[1], b[1]])] # Построение таблицы кодирования codes = {} def build_codes(node, code=""): if isinstance(node[1], str): codes[node[1]] = code else: build_codes(node[1][0], code + '0') build_codes(node[1][1], code + '1') build_codes(nodes[0]) # Кодирование текста encoded_text = ".join(codes[char] for char in text) return encoded_text # Пример использования text = "Hello, World!" encoded_text = huffman_encode(text) print(encoded_text)`

б) Решение задач на сжатие данных:

Задача: Реализовать алгоритм LZW для сжатия и распаковки текстовых данных.

Решение (Python): `def lzw_encode(text): # Создание словаря символов dictionary = {chr(i): i for i in range(256)} next_code = 256 encoded_text = [] current_code = "" for char in text: potential_code = current_code + char if potential_code in dictionary: current_code = potential_code else: encoded_text.append(dictionary[current_code]) dictionary[potential_code] = next_code next_code += 1 current_code = char if current_code: encoded_text.append(dictionary[current_code]) return encoded_text def lzw_decode(encoded_text): # Создание словаря символов dictionary = {i: chr(i) for i in range(256)} next_code = 256 decoded_text = "" prev_code = encoded_text[0] decoded_text += dictionary[prev_code] for code in encoded_text[1:]: if code in dictionary: entry = dictionary[code] else: entry = dictionary[prev_code] + dictionary[prev_code][0] decoded_text += entry dictionary[next_code] = dictionary[prev_code] + entry[0] next_code += 1 prev_code = code return decoded_text # Пример использования text = "Hello, World!" encoded_text =`

```
lzw_encode(text) decoded_text = lzw_decode(encoded_text) print(encoded_text)
print(decoded_text)
```

2. Задачи на разработку компьютерных моделей:

а) Моделирование различных объектов и процессов:

Задача: Разработать компьютерную модель движения тела под действием силы тяжести и силы сопротивления воздуха.

Решение (Python):

```
import numpy as np
import matplotlib.pyplot as plt

def simulate_motion(initial_velocity, angle, time_step, air_resistance):
    # Начальные условия
    x = 0
    y = 0
    vx = initial_velocity * np.cos(np.radians(angle))
    vy = initial_velocity * np.sin(np.radians(angle))
    # Параметры моделирования
    g = 9.8 # Ускорение свободного падения, м/с^2
    time = 0
    # Списки для хранения координат
    x_positions = [x]
    y_positions = [y]
    while y >= 0:
        # Вычисление ускорений
        ax = -air_resistance * vx
        ay = -g - air_resistance * vy
        # Вычисление новых скоростей и координат
        vx += ax * time_step
        vy += ay * time_step
        time += time_step
        x += vx * time_step
        y += vy * time_step
    # Обновление списков координат
    x_positions.append(x)
    y_positions.append(y)
    time += time_step
    return x_positions, y_positions

# Пример использования
initial_velocity = 50 # м/с
angle = 45 # Угол в градусах
time_step = 0.01 # Шаг времени, с
air_resistance = 0.1 # Коэффициент сопротивления воздуха
x_positions, y_positions = simulate_motion(initial_velocity, angle, time_step, air_resistance)
# Визуализация результатов
plt.figure(figsize=(8, 6))
plt.plot(x_positions, y_positions)
plt.xlabel('Дистанция, м')
plt.ylabel('Высота, м')
plt.title('Движение тела под действием силы тяжести и сопротивления воздуха')
plt.grid(True)
plt.show()
```

3. Задачи на обработку и анализ данных:

а) Поиск, фильтрация и сортировка данных:

Задача: Написать программу, которая считывает данные о продажах из CSV-файла, фильтрует данные по заданному критерию и выводит отфильтрованные записи.

Решение (Python):

```
import csv

def filter_sales_data(file_path, min_quantity):
    filtered_sales = []
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            quantity = int(row['Quantity'])
            if quantity >= min_quantity:
                filtered_sales.append(row)
    return filtered_sales

# Пример использования
sales_file = 'sales.csv'
min_quantity = 10
filtered_data = filter_sales_data(sales_file, min_quantity)
for sale in filtered_data:
    print(sale)
```

б) Вычисление статистических показателей:

Задача: Написать программу, которая считывает данные о продажах из CSV-файла, вычисляет среднее, медиану и моду продаж, а затем выводит эти показатели.

Решение (Python):

```
import csv
from collections import Counter

def calculate_sales_statistics(file_path):
    sales_data = []
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            sales_data.append(int(row['Quantity']))
    # Вычисление среднего
    average = sum(sales_data) / len(sales_data)
    # Вычисление медианы
    sorted_sales = sorted(sales_data)
    length = len(sorted_sales)
    if length % 2 == 0:
        median = (sorted_sales[length // 2 - 1] + sorted_sales[length // 2]) / 2
    else:
        median = sorted_sales[length // 2]
    # Вычисление моды
    mode = max(Counter(sales_data).items(), key=lambda x: x[1])[0]
    return average, median, mode

# Пример использования
sales_file = 'sales.csv'
average, median, mode = calculate_sales_statistics(sales_file)
print(f"Среднее: {average:.2f}")
print(f"Медиана: {median}")
print(f"Мода: {mode}")
```

4. Задачи на использование электронных таблиц:

а) Работа с формулами и функциями:

Задача: Написать формулу в ячейке электронной таблицы, которая вычисляет сумму продаж за каждый квартал года.

Решение: Предположим, что данные о продажах расположены в диапазоне ячеек A2:B13, где A - Квартал, B - Продажи. Формула для вычисления суммы продаж за каждый квартал: =СУММ(ЕСЛИ(A2:A13=A2;B2:B13)) Эта формула использует

функцию ЕСЛИ для проверки, что значение в столбце А (Квартал) равно текущему кварталу, и суммирует значения в столбце В (Продажи) для соответствующих строк.

б) Построение графиков и диаграмм:

Задача: Построить круговую диаграмму, отображающую долю продаж каждой категории товаров в электронной таблице.

Решение

1. Подготовьте данные в электронной таблице, где столбцы содержат информацию о Категории и Продажах.
2. Выделите диапазон ячеек с данными.
3. На вкладке "Вставка" выберите "Круговая диаграмма".
4. Excel автоматически построит круговую диаграмму, отображающую долю продаж для каждой категории товаров.
5. Можно настроить диаграмму, добавив заголовок, легенду, изменив оформление и т.д.

3. Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №16 ЕГЭ по информатике?

Правильный вариант ответа.

Задачи на кодирование и декодирование информации

- Задачи на анализ и преобразование данных
- Задачи на разработку и использование компьютерных моделей
- Задачи на применение алгоритмов обработки данных

- Как происходит кодирование и декодирование текстовой информации?

Правильный вариант ответа.

Представление символов в компьютере с помощью кодовых таблиц (например, ASCII, Unicode)

- Алгоритмы кодирования (например, шифрование) и декодирования текста
- Определение информационного объема текста в зависимости от используемой кодировки

- Что такое компьютерная модель и какие методы используются при ее разработке?

Правильный вариант ответа.

Формализация - определение основных объектов, свойств и связей

- Алгоритмизация - разработка алгоритмов для имитации поведения модели
- Программирование - реализация модели в виде компьютерной программы
- Тестирование и отладка - проверка адекватности модели реальному объекту

- Какие алгоритмы применяются для обработки и анализа данных?

Правильный вариант ответа.

- Алгоритмы сортировки (например, быстрая сортировка, сортировка слиянием)
- Алгоритмы поиска (например, линейный поиск, бинарный поиск)
- Алгоритмы кластеризации (например, k-means, иерархическая кластеризация)
- Алгоритмы анализа данных (например, регрессионный анализ, машинное обучение)

- Какие функции и формулы могут быть использованы в электронных таблицах?

Правильный вариант ответа.

- Математические и статистические функции (SUM, AVERAGE, MIN, MAX)
- Логические функции (IF, AND, OR, NOT)
- Текстовые функции (CONCAT, LEFT, RIGHT, TRIM)
- Функции даты и времени (TODAY, NOW, DATEVALUE, TIMEVALUE)
- Ссылки на ячейки и диапазоны
- Формулы с использованием операторов (+, -, *, /, ^)

Практические вопросы (5 вопросов):

- Определите, как будет закодирован текст "Информатика" в кодировке ASCII.

Правильный вариант ответа.

Кодирование текста "Информатика" в кодировке ASCII: И - 73 н - 110 ф - 102 о - 111 р - 114 м - 109 а - 97 т - 116 и - 105 к - 107 а - 97

- Разработайте компьютерную модель, имитирующую движение планеты вокруг Солнца.

Правильный вариант ответа.

- Определить основные объекты: Солнце, планета
- Задать параметры планеты: масса, радиус орбиты, начальная скорость, угол наклона
- Разработать алгоритм расчета положения планеты с учетом гравитационного притяжения Солнца
- Реализовать модель в виде компьютерной программы
- Визуализировать движение планеты вокруг Солнца
- Как отсортировать данные в таблице по убыванию значений в столбце "Цена"?

Правильный вариант ответа.

1. Выделить диапазон данных, включая заголовок столбца "Цена"
 2. На вкладке "Данные" выбрать опцию "Сортировка"
 3. В окне сортировки выбрать столбец "Цена" и установить порядок сортировки "По убыванию"
 4. Нажать "ОК" для применения сортировки
- Вычислите среднее арифметическое значений в диапазоне ячеек A1:A10 с помощью формулы.

Правильный вариант ответа.

Вычисление среднего арифметического значений в диапазоне A1:A10:
=AVERAGE(A1:A10)

- Постройте диаграмму, отображающую распределение оценок студентов по предмету.

Правильный вариант ответа.

1. Выделить данные об оценках в таблице (столбец с оценками)
2. На вкладке "Вставка" выбрать тип диаграммы, например, гистограмму или круговую диаграмму
3. Отформатировать диаграмму, добавить заголовок, оси, легенду
4. Разместить диаграмму на листе электронной таблицы

9. Решение задания №18 ЕГЭ по информатике.

Рекомендации к плану темы.

1. Теоретическая часть:

1. Введение в задание №18 ЕГЭ по информатике:

Задание №18 ЕГЭ по информатике является наиболее сложным в экзамене. Оно проверяет глубокие знания и практические навыки в области алгоритмизации, программирования, обработки и анализа данных.

2. Основные типы задач в задании №18:

- Реализация алгоритмов с использованием рекурсии
- Разработка программ для решения практических задач
- Применение методов обработки и анализа данных
- Выполнение вычислений с использованием программирования

3. Принципы рекурсивного программирования:

- Понимание базовых рекурсивных алгоритмов (факториал, Фибоначчи, обход дерева и т.д.)
- Навыки разработки рекурсивных функций и программ
- Умение анализировать и оценивать эффективность рекурсивных алгоритмов

4. Методы разработки алгоритмов и программ для решения прикладных задач:

- Использование структурированного программирования (модульность, функции, циклы, условные операторы)
- Применение объектно-ориентированного подхода (классы, наследование,

инкапсуляция)

- Разработка алгоритмов с использованием различных парадигм (итеративная, рекурсивная, динамическое программирование)

5. Алгоритмы обработки и анализа данных:

- Сортировка, фильтрация, агрегирование данных
- Поиск статистических характеристик (среднее, медиана, мода)
- Визуализация данных (диаграммы, графики)

6. Применение программирования для выполнения вычислений:

- Реализация математических и вычислительных алгоритмов
- Численные методы решения задач (приближенное вычисление, интегрирование, дифференцирование)
- Оптимизация вычислительных процессов

7. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Написать рекурсивную функцию, которая вычисляет n-е число Фибоначчи.

Решение (Python): `def fibonacci(n): if n <= 1: return n else: return (fibonacci(n-1) + fibonacci(n-2))` # Пример использования `print(fibonacci(10))` # Результат: 55

2. Практическая часть (банк заданий):

1. Задачи на рекурсивное программирование:

а) Разработка рекурсивных функций и алгоритмов:

Задача: Написать рекурсивную функцию, которая находит наибольший общий делитель (НОД) двух чисел.

Решение (Python): `def gcd(a, b): if b == 0: return a else: return gcd(b, a % b)` # Пример использования `print(gcd(24, 18))` # Результат: 6

б) Вычисление рекурсивных последовательностей:

Задача: Написать рекурсивную функцию, которая вычисляет n-е число Фибоначчи.

Решение (Python): `def fibonacci(n): if n <= 1: return n else: return (fibonacci(n-1) + fibonacci(n-2))` # Пример использования `print(fibonacci(10))` # Результат: 55

2. Задачи на разработку программ для решения практических задач:

а) Написание программ на языках высокого уровня:

Задача: Написать программу на Python, которая считывает из текстового файла список чисел, находит их среднее арифметическое и выводит результат.

Решение (Python): `with open('numbers.txt', 'r') as file: numbers = [float(num) for num in file.read().split()] average = sum(numbers) / len(numbers) print(f'Среднее арифметическое: {average:.2f}')`

б) Реализация алгоритмов обработки данных:

Задача: Написать программу на C++, которая реализует сортировку массива методом быстрой сортировки.

Решение (C++): `#include <iostream> #include <vector> void quicksort(std::vector<int>& arr, int left, int right) { if (left < right) { int pivot = arr[(left + right) / 2]; int i = left, j = right; while (i <= j) { while (arr[i] < pivot) i++; while (arr[j] > pivot) j--; if (i <= j) { std::swap(arr[i], arr[j]); i++; j--; } } quicksort(arr, left, j); quicksort(arr, i, right); } } int main() { std::vector<int> numbers = {5, 2, 8, 1, 9, 3}; quicksort(numbers, 0, numbers.size() - 1); for (int num : numbers) { std::cout << num << " "; } return 0; }`

3. Задачи на применение методов обработки и анализа данных:

а) Поиск, фильтрация и сортировка данных:

Задача: Написать программу, которая считывает данные о продажах из CSV-файла, фильтрует данные по заданному критерию и выводит отфильтрованные записи.

Решение (Python): `import csv def filter_sales_data(file_path, min_quantity): filtered_sales = [] with open(file_path, 'r') as file: reader = csv.DictReader(file) for row in reader: quantity = int(row['Quantity']) if quantity >= min_quantity: filtered_sales.append(row) return filtered_sales`

```
filtered_sales # Пример использования sales_file = 'sales.csv' min_quantity = 10
filtered_data = filter_sales_data(sales_file, min_quantity) for sale in filtered_data: print(sale)
```

б) Вычисление статистических показателей:

Задача: Написать программу, которая считывает данные о продажах из CSV-файла, вычисляет среднее, медиану и моду продаж, а затем выводит эти показатели.

Решение (Python): import csv from collections import Counter def

```
calculate_sales_statistics(file_path): sales_data = [] with open(file_path, 'r') as file: reader =
csv.DictReader(file) for row in reader: sales_data.append(int(row['Quantity'])) # Вычисление
среднего average = sum(sales_data) / len(sales_data) # Вычисление медианы sorted_sales =
sorted(sales_data) length = len(sorted_sales) if length % 2 == 0: median =
(sorted_sales[length // 2 - 1] + sorted_sales[length // 2]) / 2 else: median = sorted_sales[length
// 2]
```

```
# Вычисление моды mode = max(Counter(sales_data).items(), key=lambda x: x[1])[0] return
average, median, mode # Пример использования sales_file = 'sales.csv' average, median,
mode = calculate_sales_statistics(sales_file) print(f"Среднее: {average:.2f}")
print(f"Медиана: {median}") print(f"Мода: {mode}")
```

4. Задачи на выполнение вычислений с использованием программирования:

а) Реализация математических вычислений:

Задача: Написать программу на Python, которая вычисляет приближенное значение числа π с заданной точностью с помощью ряда Лейбница.

Решение (Python): def calculate_pi(precision): pi = 0 sign = 1 denominator = 1 while abs(pi - 3.14159) > precision: pi += sign * 4 / denominator sign *= -1 denominator += 2 return pi #

Пример использования print(f"Значение π с точностью до 4 знаков после запятой: {calculate_pi(0.0001):.4f}")

б) Решение уравнений и систем уравнений:

Задача: Написать программу на C++, которая решает систему линейных уравнений методом Гаусса.

```
Решение (C++): #include <iostream> #include <vector> std::vector<double>
solveLinearSystem(const std::vector<std::vector<double>>& coefficients, const
std::vector<double>& constants) { int n = coefficients.size(); std::vector<double> solution(n,
0.0); // Прямой ход (преобразование в треугольную форму) for (int i = 0; i < n; i++) { //
Найти ведущий элемент int maxRow = i; for (int j = i + 1; j < n; j++) { if
(std::abs(coefficients[j][i]) > std::abs(coefficients[maxRow][i])) { maxRow = j; } }
std::swap(coefficients[i], coefficients[maxRow]); std::swap(constants[i],
constants[maxRow]); // Прямой ход for (int j = i + 1; j < n; j++) { double factor =
coefficients[j][i] / coefficients[i][i]; for (int k = i; k < n; k++) { coefficients[j][k] -= factor *
coefficients[i][k]; } constants[j] -= factor * constants[i]; } } // Обратный ход (вычисление
решений) for (int i = n - 1; i >= 0; i--) { double sum = constants[i]; for (int j = i + 1; j < n;
j++) { sum -= coefficients[i][j] * solution[j]; } solution[i] = sum / coefficients[i][i]; } return
solution; } int main() { std::vector<std::vector<double>> coefficients = {{2, 1, -1}, {-3, -1,
2}, {-2, 1, 1}}; std::vector<double> constants = {8, -11, -3}; std::vector<double> solution =
solveLinearSystem(coefficients, constants); for (double x : solution) { std::cout << x << " "; }
return 0; }
```

3. Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №18 ЕГЭ по информатике?

Правильный вариант ответа.

Задачи на анализ и преобразование данных

- Задачи на разработку алгоритмов и программ для решения прикладных задач
- Задачи на применение методов программирования (в том числе рекурсивного)
- Задачи на использование программирования для выполнения вычислений
- Что такое рекурсивное программирование и каковы его основные принципы?

Правильный вариант ответа.

Основной принцип

- функция или алгоритм вызывает сам себя для решения подзадачи
- Рекурсивные вызовы последовательно уменьшают размер задачи, приближаясь к базовому случаю
- Рекурсия позволяет компактно описывать сложные алгоритмы, обработку древовидных структур данных

- Какие методы используются при разработке алгоритмов и программ для решения прикладных задач?

Правильный вариант ответа.

- Декомпозиция - разбиение задачи на более простые подзадачи
- Абстракция - выделение ключевых объектов и операций, скрывающих детали
- Модульность - разбиение программы на независимые функциональные блоки
- Итеративная разработка - пошаговое уточнение алгоритма и программы
- Тестирование - проверка работоспособности на различных входных данных
- Какие алгоритмы применяются для обработки и анализа данных?

Правильный вариант ответа.

Алгоритмы обработки и анализа данных:

- Алгоритмы сортировки (например, быстрая, слияние)
- Алгоритмы поиска (линейный, бинарный, хэширование)
- Алгоритмы кластеризации (k-means, иерархическая)
- Алгоритмы статистического анализа (регрессия, корреляция)
- Алгоритмы машинного обучения (классификация, прогнозирование)
- Как программирование может быть использовано для выполнения вычислений?

Правильный вариант ответа.

- Реализация математических формул и функций (арифметические, тригонометрические, логические)
- Численные методы (интегрирование, дифференцирование, решение уравнений)
- Имитационное моделирование физических, биологических или экономических процессов
- Обработка и визуализация данных (построение графиков, диаграмм, отчетов)

Практические вопросы (5 вопросов):

- Напишите рекурсивную функцию, которая вычисляет факториал заданного числа.

Правильный вариант ответа.

Рекурсивная функция для вычисления факториала: `def factorial(n):`

```
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)
```

- Разработайте программу на языке Python, которая находит максимальный элемент в списке.

Правильный вариант ответа.

Программа на Python, находящая максимальный элемент в списке: `def find_max(lst):`

```
    if len(lst) == 1:
        return lst[0]
    else:
        return max(lst[0], find_max(lst[1:]))
```

```
numbers = [5, 2, 8, 1, 9]
```

```
print("Максимальный элемент:", find_max(numbers))
```

- Как отсортировать данные в таблице по убыванию значений в столбце "Цена"?

Правильный вариант ответа.

Сортировка данных в таблице по убыванию значений в столбце "Цена":

1. Выделить диапазон данных, включая заголовок столбца "Цена"
 2. На вкладке "Данные" выбрать опцию "Сортировка"
 3. В окне сортировки выбрать столбец "Цена" и установить порядок сортировки "По убыванию"
 4. Нажать "ОК" для применения сортировки
- Вычислите значение выражения 2^{10} с помощью программы на языке высокого уровня.

Правильный вариант ответа.

Вычисление 2^{10} с помощью программы на языке высокого уровня: `print(2 ** 10)` #
Результат: 1024

- Постройте диаграмму, отображающую распределение оценок студентов по предмету.

Правильный вариант ответа.

Построение диаграммы, отображающей распределение оценок студентов:

1. Выделить данные об оценках в таблице (столбец с оценками)
2. На вкладке "Вставка" выбрать тип диаграммы, например, гистограмму или круговую диаграмму
3. Отформатировать диаграмму, добавить заголовок, оси, легенду
4. Разместить диаграмму на листе электронной таблицы

10. Решение задания №22 ЕГЭ по информатике.

Правильный вариант ответа.

1. Теоретическая часть:

1. Введение в задание №22 ЕГЭ по информатике:

Задание №22 ЕГЭ по информатике является практико-ориентированным и проверяет навыки учащихся в области разработки и анализа алгоритмов, программирования, работы с базами данных и обработки табличных данных.

2. Основные типы задач в задании №22:

- Разработка и анализ алгоритмов для обработки данных
- Программирование на языках высокого уровня
- Решение задач с использованием баз данных
- Применение методов обработки табличных данных

3. Методы разработки и анализа алгоритмов для обработки данных:

- Использование структурированного программирования (модульность, функции, циклы, условные операторы)
- Разработка алгоритмов с использованием различных парадигм (итеративная, рекурсивная, динамическое программирование)
- Анализ эффективности алгоритмов (оценка временной и пространственной сложности)

4. Принципы структурного и объектно-ориентированного программирования:

- Организация программного кода в виде модулей, функций, классов
- Инкапсуляция, наследование, полиморфизм в объектно-ориентированном программировании
- Использование стандартных библиотек и расширение функциональности языка программирования

5. Основы работы с базами данных:

- Проектирование структуры баз данных (таблицы, поля, связи)
- Выполнение операций CRUD (Create, Read, Update, Delete) с данными
- Написание SQL-запросов для выборки, фильтрации, сортировки, агрегирования данных

6. Методы обработки и анализа табличных данных:

- Работа с электронными таблицами
- Применение формул и функций для вычислений, фильтрации, сортировки данных
- Визуализация данных с помощью диаграмм, графиков, сводных таблиц

7. Примеры решения задач из прошлых лет:

Рассмотрим пример решения задачи из ЕГЭ прошлых лет:

Задача: Написать программу на Python, которая считывает данные о продажах из CSV-файла, фильтрует записи по критерию минимального количества проданных товаров и выводит отфильтрованные данные.

Решение (Python):

```
import csv
def filter_sales_data(file_path, min_quantity):
    filtered_sales = []
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            quantity = int(row['Quantity'])
            if quantity >= min_quantity:
                filtered_sales.append(row)
    return filtered_sales
# Пример использования
sales_file = 'sales.csv'
min_quantity = 10
filtered_data = filter_sales_data(sales_file, min_quantity)
for sale in filtered_data:
    print(sale)
```

2. Практическая часть (банк заданий):

1. Задачи на разработку и анализ алгоритмов для обработки данных:

а) Определение выходных данных для заданных алгоритмов:

Задача: Определить выходные данные алгоритма, который находит наибольший общий делитель (НОД) двух целых чисел.

Решение: Входные данные: a, b - целые числа. Выходные данные: НОД (a, b)

б) Оценка эффективности и сложности алгоритмов:

Задача: Оценить временную сложность алгоритма сортировки пузырьком.

Решение: Временная сложность алгоритма сортировки пузырьком имеет порядок $O(n^2)$, где n - размер массива. Это означает, что время выполнения алгоритма растет пропорционально квадрату размера входных данных.

2. Задачи на программирование:

а) Написание программ на языках высокого уровня:

Задача: Написать программу на Python, которая считывает данные о продажах из CSV-файла, вычисляет средние продажи по каждому продукту и выводит результат.

Решение (Python):

```
import csv
def calculate_average_sales(file_path):
    sales_data = {}
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            product = row['Product']
            quantity = int(row['Quantity'])
            if product in sales_data:
                sales_data[product].append(quantity)
            else:
                sales_data[product] = [quantity]
    for product, quantities in sales_data.items():
        average = sum(quantities) / len(quantities)
        print(f"Средние продажи {product}: {average:.2f}")
# Пример использования
sales_file = 'sales.csv'
calculate_average_sales(sales_file)
```

б) Реализация структур данных и алгоритмических конструкций:

Задача: Написать программу на C++, которая реализует стек с операциями push, pop и peek. Решение (C++):

```
#include <iostream>
#include <vector>
class Stack {
private:
    std::vector<int> elements;
public:
    void push(int value) { elements.push_back(value); }
    void pop() { if (elements.empty()) { std::cout << "Стек пуст, операция pop невозможна" << std::endl; } else { elements.pop_back(); } }
    int peek() { if (elements.empty()) { std::cout << "Стек пуст, операция peek невозможна" << std::endl; return -1; } else { return elements.back(); } }
};
int main() {
    Stack stack;
    stack.push(10);
    stack.push(20);
    std::cout << "Верхний элемент стека: " << stack.peek() << std::endl;
    stack.pop();
    std::cout << "Верхний элемент стека: " << stack.peek() << std::endl;
    return 0;
}
```

3. Задачи с использованием баз данных:

а) Выполнение операций с данными (CRUD):

Задача: Написать SQL-запрос, который создает новую запись в таблице "Заказы" с заданными параметрами.

Решение (SQL):

```
INSERT INTO Заказы (КодЗаказа, КодКлиента, КодТовара,
Количество, ДатаЗаказа) VALUES (101, 1, 5, 2, '2023-05-15');
```

б) Составление запросов на выборку и обработку информации:

Задача: Написать SQL-запрос, который выводит список всех товаров, отсортированных по цене в порядке убывания.

Решение (SQL):

```
SELECT * FROM Товары ORDER BY Цена DESC;
```

4. Задачи на обработку табличных данных:

а) Поиск, фильтрация и сортировка данных:

Задача: Написать программу на Python, которая считывает данные о продажах из CSV-файла, фильтрует записи по критерию минимального количества проданных товаров и выводит отфильтрованные данные.

Решение (Python):

```
import csv
def filter_sales_data(file_path, min_quantity):
    filtered_sales = []
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            quantity = int(row['Quantity'])
            if quantity >= min_quantity:
                filtered_sales.append(row)
    return filtered_sales

# Пример использования
sales_file = 'sales.csv'
min_quantity = 10
filtered_data = filter_sales_data(sales_file, min_quantity)
for sale in filtered_data:
    print(sale)
```

б) Вычисление статистических показателей:

Задача: Написать программу на Python, которая считывает данные о продажах из CSV-файла, вычисляет среднее, медиану и моду продаж, а затем выводит эти показатели.

Решение (Python):

```
import csv
from collections import Counter

def calculate_sales_statistics(file_path):
    sales_data = []
    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            sales_data.append(int(row['Quantity']))

    # Вычисление среднего
    average = sum(sales_data) / len(sales_data)

    # Вычисление медианы
    sorted_sales = sorted(sales_data)
    length = len(sorted_sales)
    if length % 2 == 0:
        median = (sorted_sales[length // 2 - 1] + sorted_sales[length // 2]) / 2
    else:
        median = sorted_sales[length // 2]

    # Вычисление моды
    mode = max(Counter(sales_data).items(), key=lambda x: x[1])[0]

    return average, median, mode

# Пример использования
sales_file = 'sales.csv'
average, median, mode = calculate_sales_statistics(sales_file)
print(f"Среднее: {average:.2f}")
print(f"Медиана: {median}")
print(f"Мода: {mode}")
```

3. *Оценочные материалы (10 вопросов):*

Теоретические вопросы (5 вопросов):

- Какие типы задач встречаются в задании №22 ЕГЭ по информатике?

Правильный вариант ответа.

- Задачи на разработку и анализ алгоритмов
- Задачи на программирование и реализацию алгоритмов
- Задачи на работу с базами данных
- Задачи на обработку и анализ табличных данных
- Что такое блок-схема и как она используется для представления алгоритмов?

Правильный вариант ответа.

- Графическое представление алгоритма с использованием стандартных геометрических фигур
- Основные элементы: операция, ветвление, цикл, ввод/вывод
- Применяется для наглядного описания последовательности действий в алгоритме
- Каковы основные принципы структурного и объектно-ориентированного программирования?

Правильный вариант ответа.

Структурное:

- Разбиение программы на модули (функции, процедуры)
- Использование структур данных (массивы, списки, деревья)
- Управление потоком выполнения (ветвление, циклы)

Объектно-ориентированное:

- Объекты как сущности, объединяющие данные и методы
- Наследование, полиморфизм, инкапсуляция
- Абстракция, модульность, повторное использование кода
- Что такое база данных и какие операции можно выполнять с данными?

Правильный вариант ответа.

Организованная коллекция взаимосвязанных данных

- Основные операции: создание, чтение, обновление, удаление
- Языки запросов (SQL) для выборки, фильтрации, группировки данных

- Какие методы применяются для обработки и анализа табличных данных?

Правильный вариант ответа.

Методы обработки и анализа табличных данных:

- Сортировка по одному или нескольким столбцам
- Фильтрация и поиск данных по заданным критериям
- Агрегирование данных с помощью функций (сумма, среднее, максимум, минимум)
- Сводные таблицы для анализа данных по различным измерениям
- Визуализация данных с помощью диаграмм, графиков, отчетов

Практические вопросы (5 вопросов):

- Определите выходные данные для следующего алгоритма: "x = 5, y = 10, z = x + y, if z > 10 then print(z) else print(x)".

Правильный вариант ответа.

Выходные данные алгоритма: $x = 5$ $y = 10$ $z = x + y = 5 + 10 = 15$ Так как $z > 10$, будет выведено значение z, то есть 15.

- Напишите программу на языке Python, которая вычисляет площадь круга по заданному радиусу.

Правильный вариант ответа.

Программа на Python, вычисляющая площадь круга: `import math`

`radius = float(input("Введите радиус круга: "))`

`area = math.pi * radius ** 2`

`print("Площадь круга:", area)`

- Спроектируйте простую базу данных для учета успеваемости студентов.

Правильный вариант ответа.

Проект простой базы данных для учета успеваемости студентов:

Таблицы:

- Студенты (id, ФИО, группа)
- Успеваемость (id_студента, предмет, оценка, дата)

Связь "один-ко-многим" между таблицами "Студенты" и "Успеваемость".

- Составьте SQL-запрос, который выбирает все записи из таблицы "Товары" с ценой, превышающей 1000 рублей.

Правильный вариант ответа.

SQL-запрос, выбирающий записи из таблицы "Товары" с ценой > 1000: `SELECT * FROM Товары WHERE Цена > 1000;`

- Как можно отсортировать данные в таблице по возрастанию значений в столбце "Количество"?

Правильный вариант ответа.

1. Выделить диапазон данных, включая заголовок столбца "Количество"
2. На вкладке "Данные" выбрать опцию "Сортировка"
3. В окне сортировки выбрать столбец "Количество" и установить порядок сортировки "По возрастанию"
4. Нажать "ОК" для применения сортировки

11. Графическое представление алгоритмов.

Рекомендации к плану темы.

1. Теоретическая часть:

1. Введение в графическое представление алгоритмов:

Графическое представление алгоритмов является важным инструментом для наглядного и структурированного описания вычислительных процессов. Одним из наиболее распространенных видов графического представления является блок-схема.

2. Основные элементы блок-схем:

- а) Начало и конец алгоритма
- б) Процессы и операции

в) Ветвления и циклы

г) Ввод и вывод данных

3. Правила построения блок-схем:

- Использование стандартизированных геометрических фигур для обозначения различных элементов алгоритма
- Соблюдение направления потока выполнения алгоритма (сверху вниз, слева направо)
- Подключение элементов блок-схемы с помощью ориентированных линий
- Минимизация пересечений линий связи между элементами

4. Методы описания алгоритмов:

а) Естественнo-языковoe описание

- Использование обычного человеческого языка для описания шагов алгоритма
- Удобно для понимания, но менее формализованно

б) Псевдокод

- Промежуточный вариант между естественным языком и формальным языком программирования
- Использует ключевые слова и синтаксис, приближенные к языкам высокого уровня

в) Схемы данных

- Визуальное представление структуры данных, используемых в алгоритме
- Позволяет наглядно показать взаимосвязи между различными элементами данных

5. Примеры графического представления различных алгоритмов: Рассмотрим примеры блок-схем для следующих алгоритмов:

- Алгоритм поиска максимального элемента в массиве
- Алгоритм вычисления факториала числа
- Алгоритм сортировки массива методом пузырьковой сортировки

6. Области применения блок-схем:

- Описание и документирование программных алгоритмов
- Обучение основам программирования и алгоритмизации
- Проектирование и анализ вычислительных процессов
- Визуализация и представление сложных вычислительных задач

2. Практическая часть (банк заданий):

1. Перевод из естественнo-языковoe описания в блок-схему:

Задача: Дано естественнo-языковoe описание алгоритма: "Ввести два целых числа. Если первое число больше второго, то вывести первое число, иначе вывести второе число."

Решение: Для перевода данного описания в блок-схему, нам необходимо использовать стандартные элементы блок-схем, такие как начало/конец, ввод/вывод данных, ветвление. [

2. Перевод из псевдокода в блок-схему:

Задача: Дан псевдокод алгоритма вычисления максимального из двух чисел: `function max(a, b): if a > b then return a else return b end if end function`

Решение: Для перевода данного псевдокода в блок-схему, мы можем использовать блоки для обозначения функции, условных операторов и возврата значения.

3. Перевод из схемы данных в блок-схему:

Задача: Дана схема данных для алгоритма сортировки массива методом пузырьковой сортировки. Требуется построить блок-схему для данного алгоритма.

Решение: Для перевода схемы данных в блок-схему, необходимо проанализировать взаимосвязи между элементами данных и представить последовательность операций в виде блоков.

Оценочные материалы (10 вопросов):

Теоретические вопросы (5 вопросов):

- Что такое блок-схема и для чего она используется?

Правильный вариант ответа.

Блок-схема:

- Графическое представление алгоритма с использованием стандартных геометрических фигур
- Применяется для наглядного описания последовательности действий в алгоритме
- Перечислите основные элементы, используемые в блок-схемах.

Правильный вариант ответа.

Основные элементы блок-схем:

- Операция (прямоугольник)
- Ввод/вывод данных (параллелограмм)
- Ветвление (ромб)
- Цикл (овал или шестигранник)
- Начало/конец (овал)
- Комментарий (параллелограмм с закругленными углами)
- Какие правила необходимо соблюдать при построении блок-схем?

Правильный вариант ответа.

Правила построения блок-схем:

- Блоки должны быть расположены сверху вниз в порядке выполнения действий
- Линии связи должны идти слева направо и сверху вниз
- Использовать только стандартные фигуры
- Надписи в блоках должны быть краткими и понятными
- Чем отличаются естественно-языковое описание, псевдокод и схемы данных от блок-схем?

Правильный вариант ответа.

- Естественно-языковое описание - словесное описание алгоритма
- Псевдокод - промежуточный язык, похожий на программный код
- Схемы данных - графическое представление структур данных
- В каких областях применяются блок-схемы для представления алгоритмов?

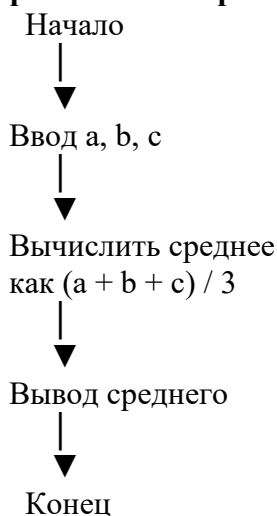
Правильный вариант ответа.

- Программирование - представление алгоритмов и программ
- Бизнес-процессы - описание рабочих процедур и операций
- Анализ данных - визуализация алгоритмов обработки данных
- Образование - наглядное объяснение и изучение алгоритмов

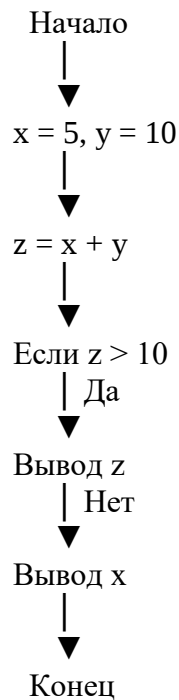
Практические вопросы (5 вопросов):

- Постройте блок-схему для алгоритма, вычисляющего среднее арифметическое трех чисел.

Правильный вариант ответа.



- Переведите следующий псевдокод в блок-схему: "x = 5, y = 10, z = x + y, if z > 10 then print(z) else print(x)".



- Определите выходные данные для блок-схемы, изображающей алгоритм поиска максимального числа в массиве.

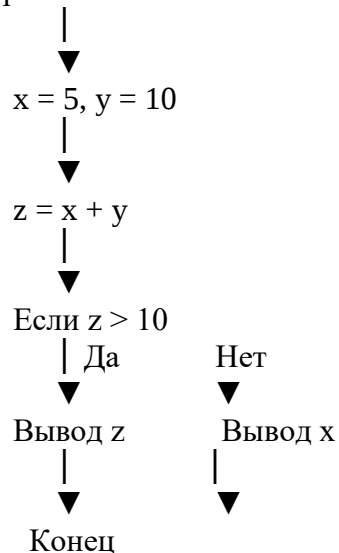
Правильный вариант ответа.

Выходные данные для блок-схемы поиска максимального числа в массиве: Предположим, что массив содержит числа {5, 2, 8, 1, 9}. Тогда выходным данным будет число 9, так как это максимальное число в массиве.

- Найдите ошибку в представленной блок-схеме и предложите исправленный вариант.

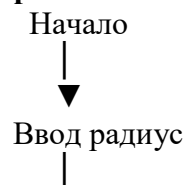
Правильный вариант ответа.

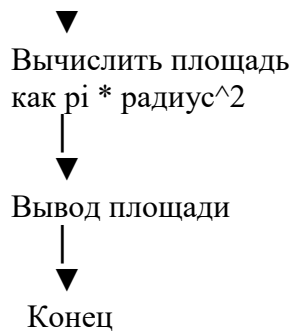
Ошибка в блок-схеме и исправленный вариант: Ошибка: Отсутствует операция сравнения после вычисления z. Исправленная блок-схема: Начало



- Разработайте блок-схему для алгоритма, который рассчитывает площадь круга по заданному радиусу.

Правильный вариант ответа.





12. Псевдокод.

Правильный вариант ответа.

1. Теоретическая часть:

1. Введение в псевдокод:

Псевдокод - это промежуточная форма записи алгоритмов, которая находится между естественно-языковым описанием и формальным языком программирования. Он позволяет более точно и структурированно представить шаги алгоритма, не привязываясь к конкретному языку.

2. Основные структуры псевдокода:

- а) Присваивание: переменной присваивается значение
- б) Ветвление (если-то-иначе): выполнение действий в зависимости от условия
- в) Циклы (для, пока, до-тех-пор): повторение действий определенное количество раз или до выполнения условия
- г) Ввод и вывод данных: получение данных от пользователя и вывод результатов

3. Правила записи псевдокода:

- Использование понятных и кратких обозначений для операций и структур
- Соблюдение отступов и иерархии для обозначения вложенности
- Комментирование сложных или нетривиальных фрагментов
- Использование понятных имен для переменных и функций

4. Преимущества использования псевдокода:

- Более высокий уровень абстракции по сравнению с языками программирования
- Удобство для быстрого описания и обсуждения алгоритмов
- Возможность представления алгоритмов независимо от конкретной реализации
- Простота перевода в программный код на различных языках

5. Сравнение псевдокода с другими формами представления алгоритмов:

- а) Блок-схемы: наглядное графическое представление алгоритма, но менее формализованное
- б) Естественно-языковое описание: понятно для человека, но менее структурированно
- в) Программирование на языках высокого уровня: более формальное и детальное, но привязано к конкретному языку

6. Примеры псевдокода для различных алгоритмов:

- а) Алгоритм поиска максимального элемента в массиве: `function findMax(array): max = array[0] for i = 1 to length(array) - 1: if array[i] > max: max = array[i] return max end function`
- б) Алгоритм сортировки методом пузырька: `function bubbleSort(array): for i = 0 to length(array) - 2: for j = 0 to length(array) - i - 2: if array[j] > array[j+1]: swap(array[j], array[j+1]) return array end function`

2. Практическая часть (банк заданий):

1. Задачи на запись простых алгоритмов в виде псевдокода:

а) Линейные алгоритмы:

Задача: Записать псевдокод для алгоритма, который вычисляет площадь круга по заданному радиусу.

Решение: `function calculateCircleArea(radius): pi = 3.14159 area = pi * radius^2 return area end function`

б) Алгоритмы с ветвлением:

Задача: Записать псевдокод для алгоритма, который определяет, является ли число четным или нечетным.

Решение: `function isEven(number): if number % 2 = 0 then return "Число является четным" else return "Число является нечетным" end if end function`

в) Циклические алгоритмы:

Задача: Записать псевдокод для алгоритма, который находит сумму всех чисел в диапазоне от 1 до n.

Решение: `function sumUpToN(n): sum = 0 for i = 1 to n: sum = sum + i return sum end function`

2. Задачи на перевод алгоритмов из одной формы в псевдокод:

а) Из блок-схем в псевдокод:

Задача: Перевести блок-схему вычисления максимума из двух чисел в псевдокод.

Решение: `function max(a, b): if a > b then return a else return b end if end function`

б) Из естественно-языкового описания в псевдокод:

Задача. Естественно-языковое описание: "Ввести два числа. Если первое число больше второго, вывести первое число, иначе вывести второе число."

Решение: `function compareNumbers(a, b): if a > b then return a else return b end if end function`

в) Из программного кода в псевдокод:

Задача. Код на Python: `def factorial(n): if n == 0: return 1 else: return n * factorial(n-1)`

Решение (псевдокод): `function factorial(n): if n = 0 then return 1 else return n * factorial(n-1) end if end function`

3. Задачи на анализ и модификацию псевдокода:

а) Определение выходных данных для заданного псевдокода:

Задача: Определить, какие данные будут выведены в результате выполнения данного псевдокода: `function calculateAverage(numbers): sum = 0 for i = 1 to length(numbers): sum = sum + numbers[i] average = sum / length(numbers) return average end function`

Решение: Выходными данными будет средневзвешенное значение элементов массива numbers. б) Поиск ошибок и внесение исправлений в псевдокод:

Задача: Найти и исправить ошибки в псевдокоде, представляющем алгоритм вычисления факториала числа.

Решение (исправленный псевдокод): `function factorial(n): if n = 0 then return 1 else return n * factorial(n-1) end if end function`

4. Задачи на практическое применение псевдокода:

а) Разработка алгоритмов для решения прикладных задач:

Задача: Записать псевдокод для алгоритма, который определяет, является ли год високосным. Решение: `function isLeapYear(year): if year % 4 = 0 and (year % 100 <> 0 or year % 400 = 0) then return "Год является високосным" else return "Год не является високосным" end if end function`

б) Использование псевдокода при программировании:

Задача: Записать псевдокод для алгоритма сортировки массива методом быстрой сортировки. Решение: `function quickSort(array, left, right): if left < right then pivot = partition(array, left, right) quickSort(array, left, pivot-1) quickSort(array, pivot+1, right) end if return array end function function partition(array, left, right): pivot = array[right] i = left - 1 for j = left to right-1: if array[j] < pivot then i = i + 1 swap(array[i], array[j]) end if end for swap(array[i+1], array[right]) return i+1 end function`

3. *Оценочные материалы (10 вопросов):*

Теоретические вопросы (5 вопросов):

- Что такое псевдокод и для чего он используется?

Правильный вариант ответа.

Псевдокод

- это неформальное, но структурированное описание алгоритма на естественном языке, которое используется для представления основных шагов решения задачи без детализации синтаксиса конкретного языка программирования.

Псевдокод используется для:

- быстрой и наглядной разработки и описания алгоритмов;
- проверки и отладки идей алгоритма до написания кода;
- объяснения алгоритмов другим людям.
- Перечислите основные структуры, используемые в псевдокоде.

Правильный вариант ответа.

Основные структуры псевдокода:

- последовательность действий;
- ветвления (if-then-else, switch);
- циклы (while, for, repeat-until);
- ввод/вывод данных;
- вызов функций/подпрограмм;
- комментарии.
- Каковы правила записи псевдокода?

Правильный вариант ответа.

Основные правила записи псевдокода:

- использование простых, понятных слов и фраз на естественном языке;
- четкая структуризация и отступы для обозначения блоков кода;
- использование общепринятых обозначений и сокращений;
- отсутствие детализированного синтаксиса языка программирования;
- возможность легкого преобразования в реальный код.
- В чем заключаются преимущества использования псевдокода по сравнению с другими формами представления алгоритмов?

Правильный вариант ответа.

Преимущества псевдокода:

- более понятное и читаемое представление алгоритма по сравнению с блок-схемами или программным кодом;
- гибкость и быстрота создания и изменения алгоритмов;
- возможность на высоком уровне сосредоточиться на логике решения, абстрагируясь от синтаксиса;
- удобство для коммуникации и объяснения алгоритмов другим людям.
- Как псевдокод соотносится с блок-схемами, естественно-языковым описанием и программированием на языках высокого уровня?

Правильный вариант ответа.

Блок-схемы - более формальное и наглядное графическое представление алгоритма.

Псевдокод более гибок и текстуален.

- Естественно-языковое описание- менее структурировано и детализировано, чем псевдокод.
- Программирование на языках высокого уровня - это реализация алгоритма в виде исполняемого кода.

Псевдокод является промежуточным этапом между алгоритмом и программой.

Практические вопросы (5 вопросов):

- Запишите псевдокод для алгоритма, вычисляющего среднее арифметическое трех чисел.

Правильный вариант ответа.

Псевдокод для алгоритма, вычисляющего среднее арифметическое трех чисел: Начало Ввести число1, число2, число3 Вычислить среднее = (число1 + число2 + число3) / 3

Вывести среднее Конец

- Переведите следующую блок-схему в псевдокод: "x = 5, y = 10, z = x + y, if z > 10 then print(z) else print(x)".

Правильный вариант ответа.

Перевод блок-схемы в псевдокод: Начало x = 5 y = 10 z = x + y Если z > 10 Тогда Вывести z Иначе Вывести x Конец

- Определите выходные данные для следующего псевдокода: "x = 5, y = 10, z = x + y, if z > 10 then print(z) else print(x)".

Выходные данные для представленного псевдокода: x = 5 y = 10 z = x + y = 5 + 10 = 15
Так как z > 10, будет выведено значение z = 15.

- Найдите ошибку в представленном псевдокоде и предложите исправленный вариант.

Правильный вариант ответа.

Ошибка в представленном псевдокоде и исправленный вариант: Ошибки нет, псевдокод записан верно.

- Разработайте псевдокод для алгоритма, который рассчитывает площадь круга по заданному радиусу.

Правильный вариант ответа.

Псевдокод для алгоритма, рассчитывающего площадь круга по заданному радиусу:
Начало Ввести радиус Вычислить площадь = $\pi * \text{радиус}^2$ Вывести площадь Конец

ТИПОВЫЕ ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ 2 СЕМЕСТР

Задание 1. Разработка веб-сайта, с использованием технологий HTML, CSS, jQuery. Сайт должен содержать от 5 до 10 страниц, иметь логическую структуру, содержать мультимедийные и интерактивные элементы.

Примерные темы:

1. История развития микроконтроллеров.

Правильный вариант ответа.

Структура сайта:

1. Главная страница
 - Краткое введение в тему микроконтроллеров
 - Навигационное меню
2. История развития микроконтроллеров
 - Первые микропроцессоры и микроконтроллеры
 - Основные вехи развития микроконтроллерной техники
3. Поколения микроконтроллеров
 - Микроконтроллеры 1-го, 2-го, 3-го и 4-го поколений
 - Особенности и характеристики каждого поколения
4. Современные микроконтроллеры
 - Популярные микроконтроллеры 8-битные, 16-битные, 32-битные
 - Области применения современных микроконтроллеров
5. Перспективы развития микроконтроллерной техники
 - Тенденции и прогнозы
 - Возможные направления дальнейшего совершенствования
6. Интерактивная временная шкала
 - Интерактивная визуализация основных этапов развития микроконтроллеров
 - Возможность масштабирования и навигации по шкале
7. Страница с контактами
 - Контактная информация разработчиков
 - Форма обратной связи

Использование технологий:

HTML:

- Semantic markup для структуризации контента
- Мультимедийные элементы (изображения, видео, аудио)
- Интерактивные элементы (формы, всплывающие окна и т.п.)

CSS:

- Адаптивная верстка и отзывчивый дизайн
- Стильное оформление элементов интерфейса
- Анимации и переходы jQuery:
- Реализация интерактивной временной шкалы
- Обработка событий (клики, наведение и т.д.)
- Плавная прокрутка между разделами сайта

Дополнительно:

- Использование библиотек и фреймворков
- Возможность реализации адаптивного меню и мобильной версии

2. История развития процессора.

Правильный вариант ответа.

1. Главная страница (index.html):

- Краткое введение и приветствие пользователей
- Навигационное меню с ссылками на другие страницы сайта
- Аттрактивный баннер или слайдер, демонстрирующий эволюцию процессоров
- Ключевые вехи в истории развития процессорной техники

2. Страница "Первые процессоры" (first-processors.html):

- Информация о первых микропроцессорах, таких как Intel 4004 и Intel 8080
- Описание характеристик и возможностей этих процессоров
- Интерактивная шкала времени, демонстрирующая появление ранних процессоров
- Фотографии и иллюстрации первых микросхем

3. Страница "Развитие x86 архитектуры" (x86-architecture.html):

- Детальное описание x86 архитектуры, ее основные особенности
- Эволюция процессоров Intel и AMD, использующих x86 архитектуру
- Сравнение характеристик и достижений различных моделей процессоров
- Мультимедийный элемент (например, интерактивная диаграмма) для визуализации производительности

производительности

4. Страница "Современные процессоры" (modern-processors.html):

- Обзор последних достижений в области процессорной техники
- Характеристики и возможности современных многоядерных и высокопроизводительных процессоров
- Сравнение различных архитектур, таких как x86, ARM, RISC-V
- Использование jQuery для реализации интерактивных элементов, например, сравнения характеристик процессоров

5. Страница "Тенденции развития" (development-trends.html):

- Прогнозы и перспективы развития процессорных технологий
- Обсуждение новейших разработок, таких как квантовые и оптические процессоры
- Инфографика, демонстрирующая тенденции роста производительности, энергоэффективности и миниатюризации
- Использование jQuery для создания интерактивных элементов, например, слайдера с прогнозами

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, анимации, эффекты прокрутки и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

3. Нейронные ускорители.

Правильный вариант ответа.

1. Главная страница (index.html):

- Введение в тему нейронных ускорителей
- Краткое описание основных концепций и применения
- Интерактивный раздел с демонстрацией работы нейронного ускорителя (с помощью jQuery)
- Ссылки на другие страницы сайта

2. Страница "Что такое нейронный ускоритель?" (what-is-neural-accelerator.html):

- Подробное объяснение принципов работы нейронных ускорителей
- Сравнение с традиционными процессорами и графическими ускорителями
- Информация о различных типах нейронных ускорителей (ASIC, FPGA, специализированные чипы)
- Использование мультимедийных элементов (изображения, диаграммы, видео) для иллюстрации концепций

3. Страница "Применение нейронных ускорителей" (neural-accelerator-applications.html):

- Области применения нейронных ускорителей (машинное обучение, компьютерное зрение, обработка естественного языка и т.д.)
- Примеры реальных продуктов и решений, использующих нейронные ускорители
- Интерактивные элементы (с помощью jQuery), демонстрирующие использование нейронных ускорителей в различных приложениях
- Ссылки на дополнительные ресурсы и статьи по теме

4. Страница "Производители нейронных ускорителей" (neural-accelerator-manufacturers.html):

- Обзор ведущих компаний, разрабатывающих нейронные ускорители
- Сравнение характеристик и особенностей решений различных производителей
- Использование jQuery для создания интерактивной таблицы или слайдера, позволяющего сравнить характеристики ускорителей
- Ссылки на официальные сайты производителей

5. Страница "Будущее нейронных ускорителей" (future-of-neural-accelerators.html):

- Прогнозы и тенденции развития технологий нейронных ускорителей
- Обсуждение новых архитектур, материалов и алгоритмов
- Инфографика, демонстрирующая возможные сценарии развития
- Использование jQuery для создания интерактивных элементов, таких как слайдер или таймлайн

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, анимации, эффекты прокрутки и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

4. Государственные информационные системы в России.

Правильный вариант ответа.

1. Главная страница (index.html):

- Введение в тему государственных информационных систем
- Краткое описание основных систем и их назначения
- Интерактивная карта России, отображающая расположение ключевых систем (реализовано с помощью jQuery)
- Ссылки на другие страницы сайта

2. Страница "Основные государственные информационные системы" (key-government-systems.html):

- Подробное описание наиболее важных государственных информационных систем (ЕГРН, ГИС ЖКХ, ЕСИА, ЕГАИС и др.)
- Информация о целях, задачах и функциях каждой системы
- Мультимедийные элементы (изображения, схемы, видео) для наглядного представления систем
- Использование jQuery для создания интерактивных элементов, например, аккордеон с информацией о каждой системе

3. Страница "Развитие и интеграция систем" (development-and-integration.html):

- Обзор процесса развития и совершенствования государственных информационных систем
- Информация об интеграции систем и обмене данными между ними
- Использование jQuery для создания интерактивной временной шкалы, демонстрирующей этапы развития систем
- Ссылки на официальные источники и дополнительные ресурсы

4. Страница "Нормативно-правовая база" (legal-framework.html):

- Обзор законодательства, регулирующего государственные информационные системы
- Информация о ключевых нормативных актах и их требованиях
- Использование jQuery для создания интерактивной таблицы или слайдера, позволяющего сравнить различные правовые акты
- Ссылки на официальные тексты нормативных документов

5. Страница "Доступ и взаимодействие" (access-and-interaction.html):

- Информация о способах доступа и взаимодействия граждан и организаций с

государственными информационными системами

- Описание порталов и интерфейсов, предоставляющих доступ к системам
- Использование jQuery для создания интерактивных элементов, демонстрирующих процесс взаимодействия с системами
- Ссылки на официальные порталы и руководства по работе с системами

6. Страница "О проекте" (about-project.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме государственных информационных систем

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как карты, слайдеры, аккордеоны, временные шкалы и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

5. Промпт-инжиниринг.

Правильный вариант ответа.

1. Главная страница (index.html):

- Вводная информация о промпт-инжиниринге и его значении
- Интерактивный баннер или слайдер, демонстрирующий примеры применения промпт-инжиниринга
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое промпт-инжиниринг?" (what-is-prompt-engineering.html):

- Подробное определение и объяснение концепции промпт-инжиниринга
- Информация о различных типах промптов и их использовании
- Мультимедийные элементы (изображения, диаграммы, видео) для иллюстрации промпт-инжиниринга
- Использование jQuery для создания интерактивных примеров промптов

3. Страница "Области применения" (applications-of-prompt-engineering.html):

- Обзор основных областей, где используется промпт-инжиниринг (машинное обучение, разработка ПО, научные исследования и т.д.)
- Описание конкретных примеров использования промпт-инжиниринга в различных сферах
- Интерактивная система фильтрации или сортировки областей применения (реализовано с помощью jQuery)
- Ссылки на дополнительные ресурсы по тематике

4. Страница "Инструменты и технологии" (tools-and-technologies.html):

- Обзор инструментов и технологий, используемых в промпт-инжиниринге
- Информация об основных языках программирования, фреймворках и библиотеках
- Сравнение характеристик и возможностей различных инструментов с помощью интерактивной таблицы (jQuery)
- Ссылки на официальные сайты и документацию инструментов

5. Страница "Лучшие практики" (best-practices.html):

- Рекомендации и советы по эффективному использованию промпт-инжиниринга
- Примеры успешных кейсов и их анализ
- Интерактивные элементы (jQuery), позволяющие пользователям оценить свои знания и навыки в промпт-инжиниринге
- Ссылки на обучающие ресурсы и сообщества

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме промпт-инжиниринга

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, фильтры, таблицы сравнения и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

6. Как участвовать в хакатонах?

Правильный вариант ответа.

1. Главная страница (index.html):

- Вводная информация о хакатонах и их значении
- Интерактивный баннер или слайдер с отображением различных хакатонов
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое хакатон?" (what-is-a-hackathon.html):

- Подробное определение и объяснение концепции хакатона
- Информация о различных форматах и типах хакатонов
- Мультимедийные элементы (изображения, видео) для иллюстрации хакатонов
- Использование jQuery для создания интерактивных таймеров или счетчиков, демонстрирующих продолжительность хакатонов

3. Страница "Как подготовиться к хакатону?" (how-to-prepare-for-a-hackathon.html):

- Советы и рекомендации по подготовке к участию в хакатоне
- Информация о необходимых навыках, инструментах и ресурсах
- Использование jQuery для создания интерактивных чек-листов или пошаговых руководств
- Ссылки на дополнительные материалы по подготовке к хакатам

4. Страница "Регистрация и участие" (registration-and-participation.html):

- Инструкции по регистрации и подаче заявки на участие в хакатонах
- Информация о требованиях, критериях отбора и правилах участия
- Интерактивная форма регистрации с использованием jQuery
- Ссылки на официальные сайты и платформы для поиска хакатонов

5. Страница "Успешное выступление" (successful-hackathon-performance.html):

- Рекомендации по эффективному участию в хакатоне
- Примеры успешных кейсов и разбор стратегий победителей

- Использование jQuery для создания интерактивных симуляторов или мини-игр, моделирующих хакатонные условия
- Ссылки на обучающие ресурсы и сообщества по хакатонам

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме хакатонов

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, таймеры, симуляторы и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

7. Современная робототехника.

Правильный вариант ответа.

1. Главная страница (index.html):

- Вводная информация о современной робототехнике и ее развитии
- Интерактивный баннер или слайдер с демонстрацией различных робототехнических решений
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое современная робототехника?" (what-is-modern-robotics.html):

- Подробное определение и объяснение концепции современной робототехники
- Информация о различных типах роботов, их классификация и характеристики
- Использование мультимедийных элементов (изображения, видео) для иллюстрации различных робототехнических систем
- Интерактивные элементы (с помощью jQuery), позволяющие пользователям изучить основные компоненты роботов

3. Страница "Области применения" (applications-of-modern-robotics.html):

- Обзор основных областей, где используется современная робототехника (промышленность, медицина, сельское хозяйство, сфера услуг и т.д.)
- Описание конкретных примеров внедрения робототехнических решений в различных сферах
- Интерактивная система фильтрации или сортировки областей применения (реализовано с помощью jQuery)
- Ссылки на дополнительные ресурсы по тематике

4. Страница "Технологии и инновации" (technologies-and-innovations.html):

- Обзор новейших технологий, используемых в современной робототехнике
- Информация об интеллектуальных системах, сенсорах, приводах, методах управления и т.д.
- Использование jQuery для создания интерактивных элементов, демонстрирующих принцип работы технологий
- Ссылки на официальные источники и документацию по технологиям

5. Страница "Будущее робототехники" (future-of-robotics.html):

- Прогнозы и тенденции развития современной робототехники
- Информация об ожидаемых инновациях и перспективных разработках
- Использование jQuery для создания интерактивных таймлайнов или инфографики, демонстрирующих возможное развитие робототехники
- Ссылки на ведущие исследовательские центры и сообщества

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме современной робототехники

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, фильтры, анимированные демонстрации и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, инфографика) для визуализации информации

8. Современные мультимедийные технологии.

Правильный вариант ответа.

1. Главная страница (index.html):

- Вводная информация о современных мультимедийных технологиях и их развитии
- Интерактивный баннер или слайдер, демонстрирующий различные примеры применения мультимедиа
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое современные мультимедийные технологии?" (what-are-modern-multimedia-technologies.html):

- Подробное определение и объяснение концепции современных мультимедийных технологий
- Информация о различных типах мультимедийных форматов, устройств и программных решений
- Использование мультимедийных элементов (изображения, аудио, видео) для иллюстрации технологий
- Интерактивные элементы (с помощью jQuery), позволяющие пользователям изучить и протестировать различные мультимедийные форматы

3. Страница "Области применения" (applications-of-modern-multimedia.html):

- Обзор основных областей, где используются современные мультимедийные технологии (развлечения, образование, бизнес, маркетинг и т.д.)
- Описание конкретных примеров внедрения мультимедийных решений в различных сферах
- Интерактивная система фильтрации или сортировки областей применения (реализовано с помощью jQuery)
- Ссылки на дополнительные ресурсы по тематике

4. Страница "Технологии и инновации" (technologies-and-innovations.html):

- Обзор новейших мультимедийных технологий, таких как виртуальная и

дополненная реальность, 3D-графика, передовые методы визуализации и т.д.

- Информация об аппаратных и программных решениях, используемых в современных мультимедийных системах
- Использование jQuery для создания интерактивных демонстраций и симуляторов, иллюстрирующих работу технологий
- Ссылки на официальные источники и документацию по технологиям

5. Страница "Тенденции и перспективы" (trends-and-prospects.html):

- Прогнозы и тенденции развития современных мультимедийных технологий
- Информация об ожидаемых инновациях и перспективных разработках
- Использование jQuery для создания интерактивных таймлайнов или инфографики, демонстрирующих возможное развитие мультимедийных технологий
- Ссылки на ведущие исследовательские центры и сообщества

6. Страница "О нас" (about-us.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме современных мультимедийных технологий

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, фильтры, демонстрационные симуляторы и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, аудио, видео, 3D-графика) для визуализации информации

9. Языковые модели искусственного интеллекта.

Правильный вариант ответа.

1. Главная страница (index.html):

- Введение в тему языковых моделей искусственного интеллекта
- Интерактивный баннер или слайдер, демонстрирующий примеры применения языковых моделей
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое языковые модели ИИ?" (what-are-ai-language-models.html):

- Подробное определение и объяснение концепции языковых моделей искусственного интеллекта
- Информация о различных типах языковых моделей, таких как GPT, BERT, T5 и других
- Использование мультимедийных элементов (изображения, видео) для иллюстрации работы языковых моделей
- Интерактивные элементы (с помощью jQuery), позволяющие пользователям протестировать возможности языковых моделей

3. Страница "Области применения" (applications-of-ai-language-models.html):

- Обзор основных областей, где используются языковые модели искусственного интеллекта (обработка естественного языка, генерация текста, перевод, вопросно-ответные системы и т.д.)
- Описание конкретных примеров внедрения языковых моделей в различные сферы

- Интерактивная система фильтрации или сортировки областей применения (реализовано с помощью jQuery)
- Ссылки на дополнительные ресурсы по тематике

4. Страница "Технологии и архитектуры" (technologies-and-architectures.html):

- Обзор ключевых технологий, лежащих в основе языковых моделей ИИ (нейронные сети, трансформеры, самоконтролируемое обучение и т.д.)
- Информация об архитектурах и алгоритмах, используемых в различных языковых моделях
- Использование jQuery для создания интерактивных визуализаций и демонстраций, объясняющих принципы работы технологий
- Ссылки на официальные источники и документацию

5. Страница "Этические аспекты" (ethical-considerations.html):

- Обсуждение возможных этических проблем и рисков, связанных с использованием языковых моделей ИИ
- Информация о существующих подходах, стандартах и рекомендациях по этичному применению технологий
- Использование jQuery для создания интерактивных опросов или дискуссионных форумов, позволяющих пользователям высказать свое мнение
- Ссылки на соответствующие исследования и публикации

6. Страница "О проекте" (about-project.html):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме языковых моделей ИИ

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, демонстрационные инструменты, опросы и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, интерактивные визуализации) для наглядного представления информации

10. Компьютерное зрение.

Правильный вариант ответа.

1. Главная страница (index.html):

- Вводная информация о компьютерном зрении и его значении
- Интерактивный баннер или слайдер, демонстрирующий примеры применения технологий компьютерного зрения
- Краткое описание основных разделов сайта
- Ссылки на другие страницы

2. Страница "Что такое компьютерное зрение?" (what-is-computer-vision.html):

- Подробное определение и объяснение концепции компьютерного зрения
- Информация о различных алгоритмах и методах, используемых в компьютерном зрении
- Использование мультимедийных элементов (изображения, видео) для иллюстрации работы систем компьютерного зрения
- Интерактивные элементы (с помощью jQuery), позволяющие пользователям

протестировать отдельные компоненты систем компьютерного зрения

3. Страница "Области применения" ([applications-of-computer-vision.html](#)):

- Обзор основных областей, где используется компьютерное зрение (распознавание объектов, обработка медицинских изображений, беспилотные транспортные средства и т.д.)
- Описание конкретных примеров внедрения технологий компьютерного зрения в различные сферы
- Интерактивная система фильтрации или сортировки областей применения (реализовано с помощью jQuery)
- Ссылки на дополнительные ресурсы по тематике

4. Страница "Технологии и инновации" ([technologies-and-innovations.html](#)):

- Обзор новейших технологий и методов, используемых в компьютерном зрении (глубокое обучение, нейронные сети, компьютерное зрение на основе сенсоров и т.д.)
- Информация об аппаратных и программных решениях, необходимых для реализации систем компьютерного зрения
- Использование jQuery для создания интерактивных демонстраций и симуляторов, иллюстрирующих работу технологий
- Ссылки на официальные источники и документацию по технологиям

5. Страница "Будущее компьютерного зрения" ([future-of-computer-vision.html](#)):

- Прогнозы и тенденции развития технологий компьютерного зрения
- Информация об ожидаемых инновациях и перспективных разработках
- Использование jQuery для создания интерактивных таймлайнов или инфографики, демонстрирующих возможное развитие компьютерного зрения
- Ссылки на ведущие исследовательские центры и сообщества

6. Страница "О нас" ([about-us.html](#)):

- Информация об авторах и разработчиках сайта
- Контактные данные для обратной связи
- Ссылки на другие ресурсы по теме компьютерного зрения

Общая структура сайта:

- Использование HTML для создания основной структуры и содержания страниц
- Применение CSS для оформления и стилизации элементов
- Внедрение jQuery для реализации интерактивных и мультимедийных компонентов, таких как слайдеры, фильтры, демонстрационные симуляторы и т.д.
- Обеспечение логической навигации между страницами с помощью ссылок и меню
- Внедрение мультимедийных элементов (изображения, видео, интерактивные визуализации) для наглядного представления информации

Задание 2. Решение задач из школьного курса математики и физики с использованием онлайн-системы GeoGebra Classic.

Правильный вариант ответа.

1. Решение алгебраических уравнений:

Задача: Решить уравнение $x^3 - 2x^2 - 3x + 6 = 0$.

Решение в GeoGebra:

- Создайте функцию $f(x) = x^3 - 2x^2 - 3x + 6$
- Используйте инструмент "Найти корни" для нахождения корней уравнения
- Результат: $x = -1$, $x = 2$, $x = 1$

2. Решение систем уравнений:

Задача: Решить систему уравнений: $2x + 3y = 12$, $x - y = 4$.

Решение в GeoGebra:

- Создайте две прямые, соответствующие уравнениям
- Используйте инструмент "Пересечение" для нахождения точки пересечения
- Результат: $x = 4$, $y = 2$

3. Решение уравнений с параметром:

Задача: Решить уравнение $x^2 - 2ax + a = 0$ при $a = 2$.

Решение в GeoGebra:

- Создайте функцию $f(x,a) = x^2 - 2ax + a$
- Подставьте $a = 2$ и найдите корни уравнения
- Результат: $x = 1$ или $x = 3$

4. Графический метод решения уравнений с модулем:

Задача: Решить уравнение $|x - 2| = |x + 1|$ графически.

Решение в GeoGebra:

- Создайте две функции: $f(x) = |x - 2|$ и $g(x) = |x + 1|$
- Постройте графики функций на одном рисунке
- Определите точки пересечения графиков - они являются решениями уравнения
- Результат: $x = -1/2$ или $x = 3/2$

5. Решение задач на свободное падение:

Задача: Тело брошено вертикально вверх со скоростью 20 м/с. Определить, через какое время тело достигнет максимальной высоты и какова эта высота.

Решение в GeoGebra:

- Создайте динамический график, отображающий зависимость высоты от времени
- Используйте формулу $s = v_0t + 1/2at^2$ для вычисления высоты
- Определите время достижения максимальной высоты и саму высоту
- Результат: время достижения максимальной высоты = 2 с, максимальная высота = 20 м

6. Решение задач на движение тела, брошенного под углом к горизонту:

Задача: Тело брошено под углом 30° к горизонту со скоростью 40 м/с. Определите максимальную высоту подъема и дальность полета.

Решение в GeoGebra:

- Создайте динамический график, отображающий траекторию движения тела
- Используйте формулы для вычисления высоты и дальности полета
- Определите максимальную высоту подъема и дальность полета
- Результат: максимальная высота подъема = 40 м, дальность полета = 80 м

7. Решение геометрических задач на построение:

Задача: Построить биссектрису угла, используя циркуль и линейку.

Решение в GeoGebra:

- Создайте угол
- Используйте инструмент "Биссектриса" для построения биссектрисы угла
- Проверьте, что построенная биссектриса делит угол пополам

Перечень вопросов

1. Цели, задачи, цифровая инфраструктура организации, где обучающийся проходил практику.

Правильный вариант ответа.

Цели:

- Повышение эффективности образовательного процесса за счет внедрения современных цифровых технологий.
- Формирование у студентов компетенций в области цифровых технологий, необходимых для их успешной профессиональной деятельности.

- Создание цифровой среды, способствующей непрерывному обучению, научным исследованиям и инновациям.
- Развитие "умного" кампуса с широким использованием технологий Интернета вещей, искусственного интеллекта и больших данных.

Задачи:

- Модернизация ИТ-инфраструктуры университета для обеспечения надежной и высокопроизводительной цифровой среды.
- Внедрение современных образовательных технологий, таких как онлайн-обучение, виртуальные лаборатории, адаптивные обучающие системы.
- Развитие центра компетенций в области искусственного интеллекта, больших данных, Интернета вещей.
- Создание цифровых сервисов для студентов, преподавателей и сотрудников (личные кабинеты, мобильные приложения, электронный документооборот).
- Обучение и повышение цифровой грамотности всех участников образовательного процесса.

Цифровая инфраструктура организации:

- Современная сетевая инфраструктура с высокоскоростным доступом в Интернет.
- Центр обработки данных с резервированием и отказоустойчивыми системами.
- Единая информационная система управления университетом, включающая модули для автоматизации основных процессов.
- Парк вычислительной техники, мультимедийное оборудование и интерактивные средства обучения.
- Специализированные лаборатории и центры компетенций в области новых технологий.
- Система видеоконференцсвязи и средства для организации онлайн-обучения.

2. Перечень программного обеспечения используемого в организации.

Правильный вариант ответа.

- Операционные системы
 - Офисные приложения
 - Системы управления обучением (LMS)
 - Средства видеоконференцсвязи
 - Системы управления проектами
 - СУБД
 - Среды разработки
 - Инструменты анализа и визуализации данных

3. Основные нормативно-правовые документы регламентирующие работу организации

Правильный вариант ответа.

- Федеральные законы об образовании и науке
 - Устав университета и локальные нормативные акты
 - Государственные образовательные стандарты
 - Лицензия на образовательную деятельность, аккредитация

4. Какими факторами определяется сферы интересов организации?

Правильный вариант ответа.

- Стратегические приоритеты и миссия университета
 - Потребности рынка труда и заказчиков образовательных услуг
 - Научно-исследовательские компетенции и материально-техническая база
 - Региональные и национальные программы развития образования и науки

5. Приоритеты в деятельности организации?

Правильный вариант ответа.

- Качество образовательного процесса
- Развитие научно-исследовательской деятельности

- Внедрение инноваций и коммерциализация разработок
 - Интернационализация и развитие международных связей
6. Какие методы и подходы используются в работе организации

Правильный вариант ответа.

- Компетентностный подход в образовании
- Проектное и проблемное обучение
- Применение информационных технологий
- Интеграция науки, образования и практики

7. Какое место в организационной иерархии занимает подразделение, в котором обучающийся проходил практику?

Правильный вариант ответа.

- Подразделение (факультет, кафедра, лаборатория) может находиться в подчинении заведующего кафедрой, директора института

8. Какие программные средства могли быть использованы в данном подразделении для оптимизации его работы

Правильный вариант ответа.

- Системы управления учебным процессом (расписание, успеваемость, документооборот)
- Программы для организации научных исследований (сбор данных, обработка, визуализация)
- Средства коммуникации и совместной работы (видеоконференции, совместный доступ к файлам)

9. В чем преимущества в организации работы с указанными программными продуктами

Правильный вариант ответа.

- Автоматизация рутинных операций
- Повышение эффективности и производительности
- Улучшение качества и доступности информации
- Поддержка совместной работы и коммуникаций

10. Рекомендации по совершенствованию деятельности подразделения, организации в целом

Правильный вариант ответа.

- Оптимизация бизнес-процессов и документооборота
- Совершенствование информационных систем и их интеграция
- Развитие компетенций сотрудников в области ИТ
- Улучшение материально-технической базы подразделений
- Повышение эффективности взаимодействия с другими подразделениями

ВОПРОСЫ ПРИ ЗАЩИТЕ ОТЧЕТА

Перечень вопросов
1. Опишите основное направление деятельности организации Правильный вариант ответа. - Университет является ведущим образовательным и научно-исследовательским учреждением. - Основные направления деятельности: - Образовательная деятельность: реализация программ высшего образования (бакалавриат, магистратура, аспирантура), дополнительного профессионального образования. - Научно-исследовательская деятельность: проведение фундаментальных и прикладных научных исследований, разработка инновационных технологий. - Воспитательная и внеучебная работа со студентами.

- Взаимодействие с работодателями, органами власти, социальными партнерами.

2. Каков спектр прикладных программ, применяемых при данной деятельности

Правильный вариант ответа.

- Системы управления обучением (LMS)
- Офисные пакеты
- Средства видеоконференцсвязи
- Системы управления проектами
- Программы для анализа данных и визуализации
- Среды разработки программного обеспечения
- Специализированное ПО для проведения научных исследований, моделирования

3. Какими из перечисленных программных продуктов вы овладели при прохождении практики

Правильный вариант ответа.

- Системы управления обучением для работы с учебными курсами, заданиями, оценками
- Офисные приложения для подготовки документов, презентаций, электронных таблиц
- Средства видеоконференцсвязи для участия в онлайн-мероприятиях
- Программы для организации совместной работы при выполнении групповых проектов
- Инструменты для анализа данных и визуализации при работе над исследовательскими задачами

4. Можно ли использовать альтернативные программы и в чем их преимущества

Правильный вариант ответа.

- Бесплатное/открытое программное обеспечение
- Кроссплатформенность и возможность использования на различных ОС
- Более гибкая настройка и адаптация под специфические потребности
- Современный дизайн и удобный пользовательский интерфейс
- Интеграция с другими инструментами и сервисами
- Выбор альтернативных программных продуктов зависит от конкретных задач и оценки их соответствия потребностям университета

5. Оцените возможные пути внедрения современных прикладных программ в деятельность предприятия

Правильный вариант ответа.

Важно анализировать функциональные возможности, стоимость, совместимость с существующими системами, качество поддержки и отзывы пользователей. Также стоит изучить, предлагает ли выбранное ПО возможность кастомизации.

6. Формирование стратегии цифровой трансформации университета с учетом технологических тенденций

Правильный вариант ответа.

Необходимо провести аудит существующих технологий, программного обеспечения и цифровых сервисов, доступных в университете. Оценить уровень цифровой зрелости как сотрудников, так и студентов, определить, какие ресурсы уже используются и насколько они эффективны.

7. Проведение пилотных проектов по апробации новых программных продуктов

Правильный вариант ответа.

- Установить системы сбора данных, позволяющие отслеживать использование программы, успехи пользователей, а также возможные ошибки и недостатки.
- Включить опросы пользователей для получения обратной связи о функционировании системы. Проведение пилотных проектов является неотъемлемой частью внедрения новых программных продуктов в организации. Это позволяет минимизировать риски, протестировать функциональность и сбалансировать инвестиции в новые технологии с реальной потребностью бизнеса. Через тщательное планирование, мониторинг и анализ пилотных проектов организации могут принимать обоснованные решения о дальнейшем внедрении новых решений.

8. Обучение сотрудников и студентов работе с современными прикладными программами

Правильный вариант ответа.

Обучение сотрудников и студентов работе с современными прикладными программами — это стратегически важный процесс, требующий планирования, организации и постоянной оценки. Основной целью является создание среды, где пользователи чувствуют себя уверенными и способны эффективно использовать новые технологии для достижения поставленных целей. Правильное обучение не только увеличивает производительность, но и способствует культурным изменениям в организации, поддерживая инновации и адаптацию к изменениям.

9. Обеспечение необходимой IT-инфраструктуры и техническая поддержка внедряемых решений

Правильный вариант ответа.

1. Обеспечение необходимой IT-инфраструктуры:

- Развертывание и поддержка серверной инфраструктуры (серверы, системы хранения данных, сетевое оборудование)
- Организация корпоративной вычислительной сети с выделенными сегментами (для студентов, преподавателей, административного персонала)
- Внедрение системы управления доступом к информационным ресурсам (идентификация, авторизация, управление правами доступа)
- Внедрение и администрирование систем электронного документооборота, корпоративных информационных систем
- Обеспечение бесперебойной работы учебных и административных компьютерных классов

2. Техническая поддержка внедряемых решений:

- Организация службы технической поддержки пользователей (helpdesk, call-center)
- Оперативное устранение неисправностей и решение проблем пользователей
- Проведение регулярного профилактического обслуживания оборудования
- Обновление и патчинг программного обеспечения (операционные системы, офисные приложения, антивирусы)
- Обучение и консультирование пользователей по работе с внедряемыми информационными системами

- Анализ инцидентов, выявление узких мест и разработка корректирующих мероприятий

3. Обеспечение информационной безопасности:

- Реализация мер защиты периметра сети (межсетевые экраны, системы предотвращения вторжений)
- Внедрение систем антивирусной защиты и управления уязвимостями
- Организация резервного копирования и восстановления данных
- Разработка и внедрение политик информационной безопасности
- Обучение пользователей основам информационной безопасности

10. Интеграция прикладных программ с существующими информационными системами университета

Правильный вариант ответа.

Основные аспекты интеграции:

1. Анализ существующих информационных систем:

- Выявление используемых в университете корпоративных информационных систем
- Изучение архитектуры, интерфейсов и механизмов интеграции этих систем

2. Определение точек интеграции:

- Идентификация данных и бизнес-процессов, подлежащих интеграции
- Согласование форматов данных, протоколов обмена информацией

3. Разработка интеграционных решений:

- Проектирование архитектуры интеграции (шина данных, API, адаптеры)
- Реализация механизмов синхронизации, репликации, обмена данными

- Обеспечение безопасности передаваемой информации
- 4. Тестирование и внедрение интеграционных решений:
 - Проведение всесторонних тестов интеграционных компонентов
 - Поэтапное внедрение интеграционных решений, мониторинг их работы
 - Обучение пользователей работе с интегрированными системами
- 5. Сопровождение и развитие интеграционных решений:
 - Оперативное устранение возникающих ошибок и сбоев
 - Модернизация интеграционных компонентов по мере развития систем
 - Расширение интеграционных возможностей в соответствии с новыми потребностями
- 11. Разработка регламентов и стандартов использования программного обеспечения.

Правильный вариант ответа.

Регламенты должны охватывать следующие области:

- Правила установки и настройки ПО, включая разрешенные версии и конфигурации.
- Процедуры безопасности: требования к паролям, шифрованию, а также управление доступом.
- Инструкции по обновлению и обслуживанию ПО, включая регулярные проверки и обновления.
- Методы мониторинга использования ПО, включая инструменты для отслеживания и отчетности.

Пакет документов для отчета студента о прохождении практики

1. Индивидуальный план.
2. Дневник практиканта.
3. Отзыв – характеристика.
4. Отчет практиканта.

Рекомендации по оцениванию результатов достижения компетенций

Формой промежуточной аттестации обучающихся по итогам практики является **зачет**, который проводится в форме презентации результатов (защита отчета).

Оценивание результатов по итогам прохождения практики на промежуточной аттестации

Ответы обучающегося на зачете оцениваются по 100-балльной шкале, а итоговая оценка выставляется по пятибалльной системе.

Балльно-рейтинговая система оценки практики включает в себя 3 критерия оценки: формальный, содержательный и презентационный.

Критерии оценки по формальному признаку (не более 30 баллов):

24-30 баллов – обучающийся в установленные сроки представил отчетную документацию по итогам прохождения учебной практики: технически грамотно оформленную, четко структурированную, качественно оформленную с наличием подписей руководителя практики;

18-23 баллов - обучающийся в установленные сроки представил отчетную документацию по итогам прохождения практики: с незначительными замечаниями по оформлению, структурированную, оформленную с наличием подписей руководителя практики;

10-17 баллов - обучающийся в установленные сроки представил отчетную документацию по итогам прохождения практики, со значительными замечаниями по

оформлению, структурированную, оформленную с наличием подписей руководителя практики;

1-9 баллов - обучающийся не в установленные сроки представил отчетную документацию по итогам прохождения практики.

Критерии оценки по содержательному признаку (не более 50 баллов):

45-50 баллов – все задания выполнены верно в установленные сроки.

20-44 балла – часть заданий выполнены верно в установленные сроки.

1-19 баллов – все задания выполнялись с нарушением сроков (не соответствует индивидуальному плану практики).

Критерии оценки по презентационному признаку (не более 20 баллов):

15-20 баллов – защита отчета проведена с использованием мультимедийных средств, на заданные вопросы обучающихся представил четкие и полные ответы;

10-14 баллов – защита отчета проведена с использованием мультимедийных средств, на заданные вопросы обучающихся представил полные ответы, однако отмечены погрешности в ответе, скорректированные при собеседовании;

5-9 баллов – защита отчета проведена с использованием мультимедийных средств, на заданные вопросы обучающийся представил не полные ответы, которые невозможно скорректировать;

1-4 балла – защита отчета проведена без использования мультимедийных средств, на заданные вопросы обучающихся представил не полные ответы;

0 баллов – защита отчета не проведена (в том числе отсутствие без уважительной причины студента).

При промежуточной аттестации на зачете для перевода оценки из 100-балльной в 5-балльную необходимо пользоваться таблицей перевода.

Критерии перевода оценки из 100-балльной в 5-балльную

Рейтинговая оценка	Традиционная оценка	Уровень сформированной компетенции
50 и менее	Неудовлетворительно	Не аттестован
51-65	Удовлетворительно	Низкий
66-84	Хорошо	Средний
85-100	Отлично	Высокий

6. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ПРАКТИКИ

Основная литература:

1. Богатырев, В. А. Информационные системы и технологии. Теория надежности : учебное пособие для вузов / В. А. Богатырев. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 366 с. — (Высшее образование). — ISBN 978-5-534-15951-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/510320> (дата обращения: 15.02.2024).

2. Грекул, В. И. Проектирование информационных систем : учебник и практикум для вузов / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 423 с. — (Высшее образование). — ISBN 978-5-534-17841-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/536901> (дата обращения: 15.02.2024).

Дополнительная литература:

1. Гостев, И. М. Операционные системы : учебник и практикум для вузов / И. М. Гостев. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 164 с. —

(Высшее образование). — ISBN 978-5-534-04520-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537133> (дата обращения: 15.02.2024).

2. Толстобров, А. П. Архитектура ЭВМ : учебное пособие для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 162 с. — (Высшее образование). — ISBN 978-5-534-16839-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/543005> (дата обращения: 15.02.2024).

7. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-КОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ПРОХОЖДЕНИЯ ПРАКТИКИ

1. Государственный комитет Республики Татарстан по туризму : официальный сайт. - Казань. - URL: <https://tourism.tatarstan.ru> (дата обращения: 20.02.2023). - Текст : электронный.

2. Информационно-правовой портал Гарант : [сайт]. - Москва. — URL: <http://www.garant.ru> (дата обращения: 20.02.2023). - Текст : электронный.

3. Министерство науки и высшего образования РФ : официальный сайт. - Москва. - URL: <https://minobrnauki.gov.ru> (дата обращения 20.02.2023). - Текст: электронный.

4. Министерство по делам молодежи РТ : официальный сайт. - Казань. - URL: <https://minmol.tatarstan.ru> (дата обращения: 20.02.2023). - Текст: электронный.

5. Министерство спорта РТ : официальный сайт. - Казань. - URL: <https://minsport.tatarstan.ru> (дата обращения: 20.02.2023). - Текст: электронный.

6. Федеральная служба государственной статистики : официальный сайт. - Москва. - URL: <https://rosstat.gov.ru> (дата обращения: 20.02.2023). - Текст: электронный.

7. Федеральная служба по надзору в сфере образования и науки : официальный сайт. - Москва. - URL: <https://obrnadzor.gov.ru> (дата обращения 20.02.2023). - Текст : электронный.

8. Федеральный портал «Российское образование» : [сайт]. - Москва. - URL: <https://www.edu.ru> (дата обращения 20.02.2023). - Текст : электронный.

9. Электронный фонд актуальных правовых и нормативно-технических документов : [сайт]. - URL: <https://docs.cntd.ru> (дата обращения 20.02.2023). - Текст : электронный.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ БАЗЫ ПРАКТИКИ

Рабочее место оснащается достаточным материально-техническим оснащением в соответствии с требованиями, правилами безопасности и санитарно-гигиеническими нормами.

Практика для обучающихся инвалидов и лиц с ограниченными возможностями здоровья (далее – лица с ОВЗ) проводится с учетом особенностей их психофизического развития, индивидуальных возможностей и состояния здоровья.

Выбор мест прохождения практик для студентов-инвалидов и студентов с ОВЗ производится с учетом требований их доступности для данных обучающихся и рекомендации медико-социальной экспертизы, а также индивидуальной программе реабилитации инвалида. Формы проведения практики для инвалидов и лиц с ОВЗ устанавливаются с учетом особенностей их психофизического развития, индивидуальных возможностей и состояния здоровья.

При направлении инвалида в организацию для прохождения предусмотренной учебным планом практики Университет согласовывает с организацией условия и виды труда с учетом рекомендаций медико-социальной экспертизы и индивидуальной программы реабилитации инвалида. При необходимости для прохождения практик

могут создаваться специальные рабочие места в соответствии с характером нарушений, а также с учетом профессионального вида деятельности и характера труда, выполняемых студентом-инвалидом трудовых функций.

Наименование специальных помещений и помещений для самостоятельной работы:

420010, Республика Татарстан (Татарстан), г. Казань, тер. Деревня Универсиады, зд. 35, аудитория для проведения лекционных и практических занятий.

Оснащенность специальных помещений и помещений для самостоятельной работы: Интерактивная доска SMART Board 685i5 со встроенным проектором UF75W, персональный компьютер ICL RAY, Моноблок S922 Mi.1 (12 шт.), акустическая система активная Sven (2x25 Вт), доступ к Интернету.

Для организации самостоятельной работы обучающимся предоставляется электронный читальный зал и читальный зал информационно-ресурсного центра:

- абонемент (269,28 кв.м.): персональный компьютер ICL RAY – 4 шт., МФУ Xerox Phaser 3320, МФУ Xerox PS Fax, МФУ HP Laserjet V1530 MFP;

- электронный читальный зал (108 кв.м.): персональный компьютер ICL RAY - 30 шт., МФУ Xerox Phaser 3320 XPS;

- читальный зал (1130,42 кв.м.): 88 посадочных мест для читателей;

- научный зал (55 кв.м.): персональный компьютер ICL RAY – 6 шт., доступ к Интернет, МФУ Xerox Phaser 3320, МФУ Xerox PS Fax, МФУ HP Laserjet V1530 MFP; интерактивная доска SMART Board 685i5 со встроенным проектором UF75W.

- мобильный класс - Аппаратно-программный комплекс (АПК) на 21 ноутбук, 35 посадочных мест.

Рабочая программа практики разработана в соответствии с учебным планом по направлению подготовки 09.03.02 Информационные системы и технологии 2023 года приема.

Автор: к.ф.-м.н., доцент Мифтахов Р.Ф.

МИНИСТЕРСТВО СПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФИЗИЧЕСКОЙ КУЛЬТУРЫ, СПОРТА И ТУРИЗМА»

МЕЖДУНАРОДНЫЙ ИНСТИТУТ ГОСТИНИЧНОГО МЕНЕДЖМЕНТА
И ТУРИЗМА

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ФИДЖИТАЛ СПОРТА

Направление подготовки: 09.03.02 Информационные системы и
технологии

Направленность (профиль) подготовки: Информационные системы и
технологии

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ:
ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА**

студента(ки) I курса _____ группы

ФИО (в родительном падеже)

Сроки практики _____

Студент: _____ ФИО

Руководитель практики:
должность, ученое звание, степень
_____ ФИО

Курсовой руководитель:
должность, ученое звание, степень
_____ ФИО

Дата сдачи отчета _____

Оценка за практику _____

**Индивидуальный план
проведения учебной практики: ознакомительная практика
студентки ФИО (в родительном падеже)**

Дата проведения	Содержание мероприятий	Часы	Подпись студента

Дата:

Руководитель практики

ФИО

Студент

ФИО

ДНЕВНИК ПРАКТИКАНТА
проведения учебной практики: ознакомительная практика
студентки ФИО

Дата проведения	Содержание мероприятий	Подпись руководителя практики

Руководитель практики

ФИО

Студент

ФИО

ОТЗЫВ-ХАРАКТЕРИСТИКА

на студентку 1 курса _____ группы ФИО _____
(Ф.И.О. студента)

Студенка проходила учебную практику: ознакомительную практику на базе ФГБОУ ВО «Поволжский ГУФКСиТ» в период с _____ по _____.

За _____ период практики студентка проявила _____.

К прохождению практики относилась _____.
 Показала _____ уровень теоретической и практической подготовленности. Все задания и поручения выполняла _____, документацию вела _____.

Проведенная работа отличается _____.

Студентка обладает способностями _____.

По итогам прохождения учебной практики: ознакомительной практики у студента *сформированы / частично сформированы / не сформированы (нужное подчеркнуть)* требуемые компетенции для реализации профессиональной деятельности.

Оценка за практику _____
(цифра / прописью)

Руководитель практики ,

(должность, ученое звание, степень)

(подпись)

(И.О. Фамилия)

Курсовой руководитель

(должность, ученое звание, степень)

(подпись)

(И.О. Фамилия)