

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.П. Астафьева» (КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики

Кафедра Информатики и информационных технологий в образовании

Тимофеев Владимир Витальевич

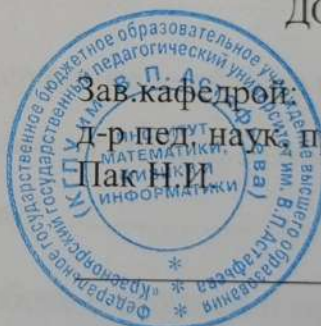
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

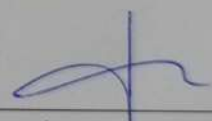
**Электронный курс как средство дополнительной подготовки к решению
задач по математическим объектам информатики**

Направление подготовки: 44.03.05 Педагогическое образование

Направленность (профиль) образовательной программы:
Математика и информатика

ДОПУСКАЮ К ЗАЩИТЕ




(дата, подпись)

Руководитель :

канд. пед. наук, доцент

Ивкина Л.М.

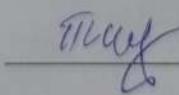


(дата, подпись)

Дата защиты 24.06.2023

Обучающийся:

Тимофеев Владимир Витальевич



(дата, подпись)

Оценка _____

Содержание

ВВЕДЕНИЕ.....	2
Глава 1. Теоретические аспекты самостоятельной подготовки к единому государственному экзамену по информатике.	5
1.1 Образовательные возможности платформы Moodle для организации самостоятельной подготовки к ЕГЭ	5
1.2 Анализ отчетов ФИПИ по информатике 2021-22 года	7
1.3 Описание структуры учебного курса для самостоятельной подготовки к ЕГЭ	14
Вывод по главе 1.	18
Глава 2. Методические рекомендации по организации самостоятельной подготовки к написанию ЕГЭ по информатике обучающихся 10-11 классов.....	19
2.1 Теоретическое и практическое содержание учебного курса для решения задач единого государственного экзамена, относящихся к математическим объектам информатики	19
2.2 Методические рекомендации по работе с теорией и практикой задач единого государственного экзамена, относящихся к математическим объектам информатики	56
2.3 Описание процедуры и результатов частичной апробации учебного курса для подготовки к ЕГЭ	62
Вывод по главе 2	67
Заключение	68
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	70

ВВЕДЕНИЕ

Изучение информатики становится необходимым для подготовки к профессиональной деятельности практически в любой области, а также для повседневного использования современных технологий. Наиболее серьезным этапом аттестации при изучении информатики является написание единого государственного экзамена (ЕГЭ) по данному предмету. Чтобы успешно справиться с некоторыми заданиями ЕГЭ, необходимо хорошо знать математические аспекты, связанные с решением задач в информатике.

С развитием Интернета и IT-технологий появляются новые возможности обучения. Такие как самостоятельное дистанционное обучение. Благодаря развитию Интернета и современных методов общения и обмена данными, становится возможным создавать и применять в обучении новые способы обучения, такие как электронные курсы с возможностями самостоятельной дистанционной подготовки к ЕГЭ. Одним из таких ресурсов является Moodle *Modular Object-Oriented Dynamic Learning Environment* (модульная объектно-ориентированная динамическая обучающая среда), позволяющая преподавателям создавать свой собственный веб-сайт, наполненный динамичными курсами, которые расширяют процесс обучения в любое время и в любом месте[15].

Противоречие:

Наличие большого количества материалов для подготовки к ЕГЭ по информатике с одной стороны и проблемой выбора необходимого материала для устранения дефицита теоретических и практических знаний для решения задач с другой стороны.

Проблема:

Согласно анализа ФППИ за два года, задания где требуются знания математических основ, решают не более 40% экзаменуемых.

Цель исследования: разработка электронного курса в LMS Moodle для подготовки к решению заданий ЕГЭ относящихся к математическим объектам информатики и методических рекомендаций по его использованию.

Объект исследования: процесс подготовки к ЕГЭ по информатике.

Предмет исследования: электронный курс в LMS Moodle для подготовки к решению задач ЕГЭ, относящихся к математическим объектам информатики.

Задачи:

1. Описать образовательные возможности платформы Moodle для организации самостоятельной подготовки к ЕГЭ.

2. Анализ отчетов ФИПИ по информатике для выявления типичных ошибок и заданий, относящихся к математическим объектам информатики и рекомендаций по их устранению.

3. Создать электронный курс для подготовки к ЕГЭ по информатике на платформе Moodle Лицея №6 “Перспектива”.

4. Описать методические рекомендации по использованию учебного курса для самостоятельной подготовки к ЕГЭ по информатике.

Для решения поставленных задач использовались следующие методы исследования: теоретические (анализ научной, методической, психолого- педагогической литературы), эмпирические (опрос).

Эмпирическая база: Электронный курс на образовательной платформе Moodle Лицей №6 “Перспектива”, результат апробации среди

учеников 11-10 классов, анализ результатов ЕГЭ 2022 года по информатике.

Структура выпускной квалификационной работы состоит из введения, двух глав, заключения и библиографического списка.

Глава 1. Теоретические аспекты самостоятельной подготовки к единому государственному экзамену по информатике.

1.1 Образовательные возможности платформы Moodle для организации самостоятельной подготовки к ЕГЭ

Moodle - это действующий и постоянно развивающийся проект. Представляет собой свободное веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения. Первая версия была разработана Мартином Дугиамасом 20 августа 2002 года, и с тех пор руководит проектом. Версии Moodle регулярно обновляются.

Самое главное преимущество, на которое сразу обращают внимание потенциальные пользователи, – бесплатное использование Moodle. Система работает по схеме open source – открытого исходного кода. За счет этого обширное сообщество программистов регулярно создает полезные расширения и модули в среде Moodle – сейчас таковых насчитывается около 1500[14].

Платформа Moodle – это мощный инструмент для организации эффективной подготовки к ЕГЭ. Это проект с открытым исходным кодом и он позволяет создавать онлайн-курсы, тесты, задания, форумы для обсуждения, учебные материалы и многое другое.

С помощью Moodle ученики могут работать над своими знаниями в любое время и в любом месте, где есть доступ в Интернет. Платформа поддерживает мобильную версию поэтому ученик может заниматься с помощью смартфона или планшета. Образовательные материалы на платформе можно предоставлять в различных форматах: текст, видео, презентации или аудио.

Вместо того чтобы выдумывать формы и возможности подготовки к ЕГЭ, можно использовать платформу: пополнять, обновлять контент и более полно использовать возможности этой системы.

Moodle обладает рядом уникальных возможностей, важных для организации самостоятельной подготовки к ЕГЭ. Каждый курс в Moodle содержит тесты и задания, которые позволяют проверить знания учеников. Кроме того, на платформе можно создавать различные виды тестов и заданий, которые помогут оценить достижения учеников на том или ином этапе обучения.

Еще одним преимуществом платформы Moodle является возможность сотрудничества с другими учениками и преподавателями. На платформе можно создать группы общения для взаимной поддержки и совместного обучения. В Moodle также предусмотрена возможность обратной связи с преподавателями, что позволяет быстро получить ответы на свои вопросы.

Moodle отслеживает успеваемость учеников и составляет отчёты для преподавателей. Например, показывает, сколько времени ребята проходили курс, какие ошибки допустили в тесте. Также преподавателю доступен журнал где он выставляет оценки и оценки за тест выставляются автоматически.

Интегрированные функции онлайн-курсов и заданий на платформе Moodle помогают ученикам подготовиться к ЕГЭ более эффективно. Она позволяет ученикам построить свой собственный учебный процесс, контролировать его и достичь желаемых результатов.

Плюс к еще вышеперечисленному учитель сам подбирает доступный и понятный материал для учеников. Ведь правильно подобранный материал который не запутает ученика, залог успешной подготовки к ЕГЭ по информатике.

От ученика только требуется самостоятельное изучение теории для успешного выполнения практических заданий.

1.2 Анализ отчетов ФИПИ по информатике 2021-22 года

ЕГЭ по информатике охватывает основное содержание курса информатики, важнейшие его темы, наиболее значимый в них материал, однозначно трактуемый в большинстве преподаваемых в школе вариантов курса информатики. Работа содержит как задания базового уровня сложности, проверяющие знания и умения, соответствующие базовому уровню подготовки по предмету, так и задания повышенного и высокого уровней, проверяющие знания и умения, владение которыми основано на углубленном изучении предмета.

ЕГЭ по информатике, проходит с использованием компьютера. Всего 27 заданий с одним кратким ответом. 11 заданий базового уровня сложности, 11 заданий повышенного уровня и 5 заданий высокого уровня сложности. Задания с 1-го по 25-е позволяют набрать по 1 первичному баллу. Задания 26, 27 позволяют набрать по 2 первичных балла каждый. Таким образом, максимум первичных баллов, которые можно получить за решение всех заданий, — 29.

Контрольными измерительными материалами охватываются основное содержание курса информатики, важнейшие его темы, наиболее значимый в них материал, однозначно трактуемый в большинстве преподаваемых в школе вариантов курса информатики. Работа содержит как задания базового уровня сложности, проверяющие знания и умения, соответствующие базовому уровню подготовки по предмету, так и задания повышенного и высокого уровней, проверяющие знания и умения, владение которыми основано на углубленном изучении предмета.

ЕГЭ по информатике охватывает следующие темы школьного курса информатики:

- информация и ее кодирование;
- моделирование и компьютерный эксперимент;
- системы счисления;

- логика и алгоритмы;
- элементы теории алгоритмов; программирование;
- обработка числовой информации;
- технологии поиска и хранения информации.

Экзаменационная модель проверяет соответствие уровня подготовки участников предметным результатам, установленным Федеральным государственным образовательным стандартом среднего общего образования.

Для базового уровня изучения информатики:

- владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- владение умениями понимать программы, написанные на выбранном для изучения универсальном Алгоритмическом языке высокого уровня, и анализировать алгоритмы с использованием таблиц; знание основных конструкций программирования;
- владение стандартными приёмами написания на Алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- сформированность представлений о компьютерно-математических моделях и необходимости анализа соответствия модели и моделируемого объекта (процесса), о способах хранения и простейшей обработке данных, знание понятия баз данных и средств доступа к ним, владение умением работать с ними.

Для углубленного уровня изучения информатики и ИКТ:

- овладение понятием сложности алгоритма; знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;

- владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных, умением использовать основные управляющие конструкции;

- владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;

- сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, об алгоритмах анализа этих объектов, о кодировании и декодировании данных и причинах искажения данных при передаче; систематизация знаний, относящихся к математическим объектам информатики; умение строить математические объекты информатики, в том числе логические формулы;

- сформированность знаний базовых принципов организации и функционирования компьютерных сетей;

- владение основными сведениями о базах данных, об их структуре.

Благодаря проведению экзамена в электронном формате была возможность оценить уровень знаний в практической работе с компьютером, включая навыки программирования, обработки информации в электронных таблицах и поиска необходимой информации.

Общее число участников экзамена в 2022 г. – 101 664 человек. Продолжается тенденция ежегодного роста числа сдающих ЕГЭ по информатике. В 2021 г. экзамен сдавало 94 023 человека, в 2020 – 84 531 человек.

Существенное влияние на снижение среднего балла оказало то, что число участников экзамена по сравнению с прошлым годом выросло в

значительной степени за счет участников с низким уровнем подготовки. Так, например, число участников, набравших 0 или 1 первичный балл в текущем году, составило 2794 человек, против 1750 человек в прошлом году, притом ни модели, ни сложность самых решаемых, т.е. самых простых, заданий базового уровня сложности не изменились по сравнению с 2021 г. Кроме того, в 2022 г. было увеличено разнообразие сюжетов заданий повышенного и высокого уровней сложности при сохранении их тематики и сложности, что, видимо, вызвало затруднения у участников, ориентированных при подготовке на заученные решения конкретных формулировок заданий.

График распределения первичных баллов ЕГЭ 2022 г. приведен на рис. 1.

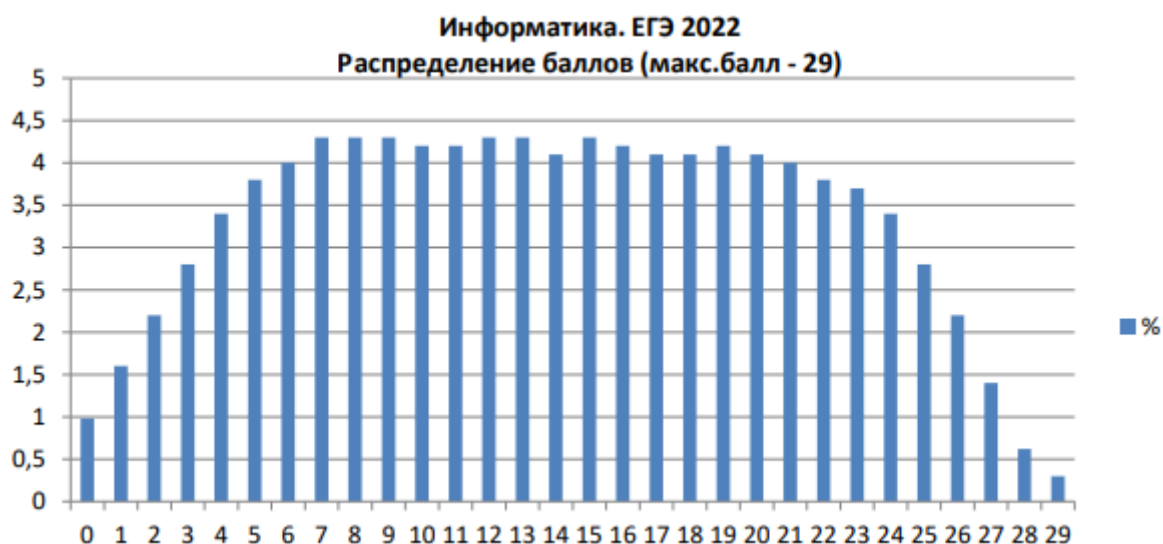


Рисунок 1. График распределения первичных баллов ЕГЭ 2022 г.

Минимальный балл ЕГЭ 2022 г., как и в предыдущие годы, составил 6 первичных баллов, что приравнялось к 40 тестовым баллам. Доля участников ЕГЭ, не набравших минимального количества баллов в 2022г., составила около 15%, что, как и снижение среднего тестового балла, связано с большим притоком на экзамен участников с низким уровнем подготовки.

Доля высокобалльников в 2022 г. превышает 18%.

В Приложении приведены результаты (средний процент выполнения) экзаменационной работы по каждой линии заданий. Средние проценты выполнения заданий представлены на диаграмме (рис. 2).



Рисунок 2. Средние проценты выполнения заданий ЕГЭ по информатике

Исходя из значений нижних границ процентов выполнения заданий различных уровней сложности (60% для базового, 40% для повышенного и 20% для высокого), можно говорить о сформированности у участников экзамена проверяемых на экзамене знаний и умений.

Участниками экзамена при выполнении заданий базового и повышенного уровней сложности был продемонстрирован наиболее высокий уровень сформированности следующих знаний и умений:

- умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы);
- умение строить таблицы истинности и логические схемы;
- умение осуществлять поиск информации в реляционных базах данных;
- умение кодировать и декодировать информацию;

- знание основных конструкций языка программирования, понятий переменной, оператора присваивания; знание о позиционных системах счисления и двоичном представлении информации в памяти компьютера;

- умение осуществлять информационный поиск средствами операционной системы или текстового процессора;

- умение исполнять алгоритм для конкретного исполнителя с фиксированным набором команд

- знание позиционных систем счисления;

- вычисление рекуррентных выражений;

- умения составлять алгоритм обработки числовой последовательности и записывать его в виде простой программы (10–15 строк) на языке программирования;

- умение анализировать алгоритм логической игры;

- умение находить выигрышную стратегию игры;

- умение анализировать алгоритм, содержащий ветвление и цикл.

У участников ЕГЭ 2022 г. возникли затруднения при выполнении заданий повышенного и высокого уровней сложности, контролирующих следующие знание и умения:

- знание основных понятий и законов математической логики;

- умение обрабатывать целочисленную информацию с использованием сортировки;

- умение составлять алгоритм обработки числовой последовательности и записывать его в виде простой программы (10– 15 строк) на языке программирования;

- умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей [27].

Исследуя диаграмму (рис.2) можно сделать выводы что задание №8 выполняют успешно лишь 30% экзаменующихся, а задание №15 выполняют 40% экзаменующихся. А в 2021 году задание №8 выполнило 49,6%, задание №15 выполнило 36,5%. Именно на этих заданиях стоит учителю предметнику заострить внимание при подготовке к ЕГЭ по информатике.

Задание №8 проверяет элементы содержания: Знание о методах измерения количества информации.

Типичные ошибки:

"При использовании способа решения со системой счисления с основанием N следует помнить, что слова в списке нумеруются с единицы, поэтому числу 0 будет соответствовать первое слово" - ФГБНУ "Федеральный институт педагогических измерений" [19]

Пример задания 8:

Варфоломей составляет коды из букв, входящих в слово ВАРФОЛОМЕЙ. Код должен состоять из 6 букв, буквы в коде не должны повторяться, согласных в коде должно быть больше, чем гласных, две гласные буквы нельзя ставить рядом. Сколько различных кодов может составить Варфоломей? [6]

Задание №15 проверяемое элементы содержания: Знание основных понятий и законов математической логики.

Типичные ошибки:

"Важно понимать, что выражение должно быть тождественно истинно, т.е. истинно при любых допустимых значениях переменных x и y , а не только при некоторых наборах значений" - ФГБНУ "Федеральный институт педагогических измерений"[20].

Пример задания 15:

На числовой прямой даны три отрезка: $P=[10,25]$, $Q=[15,30]$ и $R=[25,40]$. Какова максимальная длина отрезка A , при котором формула

$$((x \in Q) \rightarrow (x \notin R)) \wedge (x \in A) \wedge (x \notin P)$$

тождественно ложна, то есть принимает значение 0 при любом значении переменной x ?[6]

1.3 Описание структуры учебного курса для самостоятельной подготовки к ЕГЭ

Для того чтобы сдать ЕГЭ на высокий балл, ученик должен готовиться не только в школе, но и дополнительно как с репетитором так и самостоятельно. Для самостоятельной подготовки ученик должен следовать следующим правилам:

- Изучение теории. Перед тем как начать решать задачи, необходимо освоить теоретический материал. Это поможет понять алгоритмы и подходы к решению задач.
- Практика. Решать множество задач из разных разделов информатики. Это поможет научиться применять теорию на практике и улучшить навыки.
- Подготовка к экзамену. При подготовке к экзамену важно уделить внимание организации времени и распределению задач. Решить как можно больше тестовых заданий и задач, чтобы научиться быстро и правильно реагировать на задания.
- Самодисциплина и мотивация. Заниматься нужно регулярно и систематически.
- Изучать ошибки и понимать, как их можно было избежать.

Соблюдая эти принципы был разработан электронный курс на платформе LMS Moodle для решения задач связанных с математическими аспектами.



Рисунок 3. Структура изучения задания №8

Изначально курс знакомит учеником с теорией “Измерение количества информации”. Затем ученики знакомятся с решением задач, которые подразделяется на решение с помощью рассуждений и с помощью языка программирования Python. Затем предлагается видеоматериал решения заданий с помощью рассуждений и на языке программирования Python. Затем проводится тестирование на знание работы программы. После успешного тестирования ученики переходят к практике где им нужно решить подобные задания приведенные из теории.

Если у ученика возникли затруднения с решением, то на курсе есть форум “Вопрос-ответ” где он может дистанционно задать вопрос. На вопрос может ответить не только учитель, но и его одноклассники по курсу.



Рисунок 4. Структура изучения задания №15

После изучения темы “Измерение количества информации” переходим к изучению темы “Элементы математической логики”. Изучение начинается с знакомством основных элементов математической логики такие как в Таблице 1.

Таблица 1. Список логических операций и соответствующих функций в Python

Логические операции	Представления в Python
Отрицание \neg	not()
Логическое умножение \wedge	and
Логическое сложение \vee	or
Следование $A \rightarrow B$	not(A) or B
Равносильность \equiv	==

Далее обучающийся проходит тестирование на знание логических операций, затем переходит к изучению теории решения задач как с помощью рассуждений так и с помощью языка программирования Python. При изучении теории доступен видеоматериал. После переходим к практике.

Для лучшей навигации по электронному курсу задания разбиты на группы, например задание №8 разбито на три блока, первый блок “Буквы”, второй блок “Числа”, третий блок “Список” внутри блоков задачи разбиты еще на группы.

Буквы:

- Задача (Классическая);
- Задача (Каждую букву можно использовать один раз);
- Задача (Буквы составляют перестановкой);
- Задача (Проверяем первую букву слова);
- Задача (Количество гласных).

Числа:

- Задача (Перебор чисел);
- Задача (Числа, Закрепление);
- Задача (Две чётные и две нечётные цифры не стоят рядом).

Список:

- Задача (Со списками, классическая);
- Задача (Классика, Укажите номер слова).

А задание №15 разбито на группы задач:

- Задача (Неравенство, одна переменная);
- Задача (Неравенство, две переменные);
- Задача (Функция ДЕЛ);
- Задача (Поразрядная конъюнкция);
- Задача (числовая прямая);
- Задача (Числовая прямая, закрепление).

Вывод по главе 1.

1. Moodle - это действующий и постоянно развивающийся проект. Представляет собой свободное веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения. С помощью Moodle ученики могут работать над своими знаниями в любое время и в любом месте, где есть доступ в Интернет. Платформа поддерживает мобильную версию поэтому ученик может заниматься с помощью смартфона или планшета. Образовательные материалы на платформе можно предоставлять в различных форматах: текст, видео, презентации или аудио.

2. На основе анализа ФИПИ за 2021-22 гг. можно сделать выводы, что задание №8 выполнили успешно лишь 49,6% и 30% экзаменуемых соответственно. Задание №15 выполнили успешно 36,5% и 40% экзаменуемых. Следовательно создание электронного курса для подготовки к ЕГЭ с материалами по заданиям №8 и №15 актуально.

3. Структура электронного курса позволяет учителю и обучающемуся сформировать индивидуальную траекторию подготовки к ЕГЭ по информатике и включает все необходимые для этого материалы.

Глава 2. Методические рекомендации по организации самостоятельной подготовки к написанию ЕГЭ по информатике обучающихся 10-11 классов

2.1 Теоретическое и практическое содержание учебного курса для решения задач единого государственного экзамена, относящихся к математическим объектам информатики

Разработки электронных средств обучения являются достаточно трудоемкими. Попытки сделать их максимально эффективными приводят к появлению множества проблем, разрешение которых под силу лишь достаточно опытному и профессиональному разработчику. Учитель, имеющий отношение к созданию электронных средств обучения, обязан овладеть многими "тонкостями" искусства создания компьютерных программ, которые затем попадут в руки к школьнику.

Учебный курс состоит из заданий ЕГЭ по информатике, связанных с математическими аспектами, такими как комбинаторика, математическая логика и т. д.

Задания, связанные с комбинаторикой, это задания №8 ЕГЭ по информатике. Опишем теоретическое и практическое содержание раздела для решения заданий №8, опираясь на структуру изображенную на Рисунке 3. Начинаем с изучения теории, связанной с комбинаторикой и измерением количества информации. Также необходимо ознакомиться с основными понятиями такими как кодирование, алфавит, мощность алфавита и т. д. Необходимо знать основные формулы комбинаторики, например: $P = \frac{na+n*!}{na!n*}$ где $\square\square$ - количество букв, $\square*$ - количество вариантов. Затем на электронном курсе рассказывается как решать задания №8 с помощью рассуждений и на языке программирования Python.

Для удобства изучения задания были классифицированы по типам следующим образом:

- Задача (Классическая);
- Задача (Каждую букву можно использовать один раз);
- Задача (Буквы составляют перестановкой);
- Задача (Проверяем первую букву слова);
- Задача (Количество гласных);
- Задача (Перебор чисел);
- Задача (Числа, Закрепление);
- Задача (Две чётные и две нечётные цифры не стоят рядом);
- Задача (Со списками, классическая);
- Задача (Классика, Укажите номер слова).

Рассмотрим каждый тип.

Задача (Классическая):

Основная идея решения этой задачи это перебор всех возможных комбинаций из данного количества букв и длины слова и с данным ограничением.

Пример:

Женя составляет 5-буквенные слова, в которых встречаются только буквы А, Б, В, Г, причём буква А появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Женя?

Т. к. слова состоят из 5-ти символов, то мы формируем пять вложенных циклов! В каждом цикле перебираем все буквы, которые нам

дали. Внутри циклов мы составляем само слово в переменной s. Таким образом, в переменной s "прокрутятся" все возможные комбинации. Но мы подсчитываем не все комбинации, а только те, где всего одна буква А. Важно не перепутать русские и английские буквы.

Решение на языке Python:

```
k=0
for x1 in 'АБВГ':
    for x2 in 'АБВГ':
        for x3 in 'АБВГ':
            for x4 in 'АБВГ':
                for x5 in 'АБВГ':
                    s=x1+x2+x3+x4+x5
                    if s.count('A')==1:
                        k=k+1
print(k)
```

Ответ: 405

Пример:

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует трёхбуквенные слова, в которых могут быть только буквы Ш, К, О, Л, А, причём буква К появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Т. к. слова состоят из 3-ех символов, то мы формируем три вложенных цикла! В каждом цикле перебираем все буквы, которые нам дали. Внутри циклов мы составляем само слово в переменной s. Таким образом, в переменной s "прокрутятся" все возможные комбинации. Но мы подсчитываем не все комбинации, а только те, где всего одна буква К. Важно не перепутать русские и английские буквы.

Решение на языке Python:

```
k=0
for x1 in 'ШКОЛА':
    for x2 in 'ШКОЛА':
        for x3 in 'ШКОЛА':
            s=x1+x2+x3
            if s.count('K')==1:
                k=k+1
print(k)
```

Ответ: 48

Задача (Каждую букву можно использовать один раз)

Особенности данной задачи что из данного количества символов нужно составить слово при этом каждый символ можно использовать только один раз, и наложены еще дополнительные ограничения.

Пример:

Артур составляет 5-буквенные коды из букв Е, С, А, У, Л. Каждую букву нужно использовать ровно один раз, при этом нельзя ставить рядом две гласные. Сколько различных кодов может составить Артур?

В первом условии надо учесть, что каждая буква встречается в слове только один раз. Во второе условие говорит о том, что две гласные не стоят рядом (перебрать все возможные сочетания гласных).

Решение на языке Python:

```
k=0
for x1 in 'ЕСАУЛ':
    for x2 in 'ЕСАУЛ':
        for x3 in 'ЕСАУЛ':
            for x4 in 'ЕСАУЛ':
                for x5 in 'ЕСАУЛ':
                    s=x1+x2+x3+x4+x5
                    if s.count(x1)==1 and s.count(x2)==1 and s.count(x3)==1
and s.count(x4)==1 and s.count(x5)==1:
                        if s.count('EA')==0 and s.count('AE')==0 and
s.count('EY')==0 and s.count('YE')==0 and s.count('AY')==0 and
s.count('YA')==0:
                            k=k+1
print(k)
```

Ответ:12

Пример:

Петя составляет 7-буквенные слова из букв А, Б, Р, И, К, О, С. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные. Сколько различных кодов может составить Петя?

В первом условии надо учесть, что каждая буква встречается в слове только один раз. Во второе условие говорит о том, что две гласные или две согласные не стоят рядом (перебрать все возможные сочетания гласных или согласных) так как нельзя ставить подряд две гласные или две согласные мы можем выбрать в программном коде что ограничим.

Решение на языке Python:

```
k=0

for x1 in 'АБРИКОС':

    for x2 in 'АБРИКОС':

        for x3 in 'АБРИКОС':

            for x4 in 'АБРИКОС':

                for x5 in 'АБРИКОС':

                    for x6 in 'АБРИКОС':

                        for x7 in 'АБРИКОС':

                            s=x1+x2+x3+x4+x5+x6+x7

                            if s.count(x1)==1 and s.count(x2)==1 and
s.count(x3)==1 and s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1
and s.count(x7)==1:

                                if s.count('АИ')==0 and s.count('ИА')==0 and
s.count('АО')==0 and s.count('ОА')==0 and s.count('ИО')==0 and
s.count('ОИ')==0 :

                                    k=k+1

print(k)
```

Ответ: 144

Задача(Буквы составляют перестановкой)

Особенность этой задачи что два одинаковых символа не могут стоять рядом.

Пример:

Петя составляет шестибуквенные слова перестановкой букв слова КАБАЛА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?

Повторяющиеся буквы в строке, из которой берём символы в циклах, всегда убираем. Слова составляются перестановкой, значит, можно представить, что просто собирают из кубиков КАБАЛА различные слова. Следовательно, в наших словах будет ровно одна буква "К", три буквы "А", одна буква "Б" и одна буква "Л". Это программируем с помощью условия и функции `.count()`. Т.к. повторяется может только буква "А", то мы прописываем условие, что две буквы "А" подряд не могут находиться в наших комбинациях. Если все условия будут пройдены, то мы подсчитываем такие комбинации.

Решение на языке Python:

```
k=0
for x1 in 'КБЛА':
    for x2 in 'КБЛА':
        for x3 in 'КБЛА':
            for x4 in 'КБЛА':
                for x5 in 'КБЛА':
                    for x6 in 'КБЛА':
                        s=x1+x2+x3+x4+x5+x6
                        if s.count('К')==1 and s.count('А')==3 and
s.count('Б')==1 and s.count('Л')==1:
```

```
if s.count('AA')==0:
```

```
    k=k+1
```

```
print(k)
```

Ответ: 24

Пример:

Петя составляет шестибуквенные слова перестановкой букв слова АВРОРА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?

Повторяющиеся буквы в строке, из которой берём символы в циклах, всегда убираем. Слова составляются перестановкой, значит, можно представить, что просто собирают из кубиков АВРОРА различные слова. Следовательно, в наших словах будет две буквы "А", одна буква "В", две буквы "Р" и одна буква "О". Это программируем с помощью условия и функции .count(). Т.к. повторяться могут только буквы "А" и "Р", то мы прописываем условие, что две буквы "А" и две буквы "Р" подряд не могут находиться в наших комбинациях. Если все условия будут пройдены, то мы подсчитываем такие комбинации.

Решение на языке Python:

```
k=0
```

```
for x1 in 'ABPO':
```

```
    for x2 in 'ABPO':
```

```
        for x3 in 'ABPO':
```

```
            for x4 in 'ABPO':
```

```
                for x5 in 'ABPO':
```

```
                    for x6 in 'ABPO':
```

```

s=x1+x2+x3+x4+x5+x6
if s.count('A')==2 and s.count('B')==1 and
s.count('P')==2 and s.count('O')==1:
    if s.count('AA')==0 and s.count('PP')==0:
        k=k+1
print(k)

```

Ответ: 84

Задача (Проверяем первую букву слова)

Особенность этой задачи что нужно держать на контроле первую букву в слове.

Пример:

Сколько слов длины 5, начинающихся с гласной буквы, можно составить из букв Е, Г, Э? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.

Подсчитываем только те комбинации, которые начинаются с гласных букв.

Решение на языке Python:

```

k=0
for x1 in 'ЕГЭ':
    for x2 in 'ЕГЭ':
        for x3 in 'ЕГЭ':
            for x4 in 'ЕГЭ':
                for x5 in 'ЕГЭ':

```

```
s=x1+x2+x3+x4+x5
```

```
if x1=='E' or x1=='Э':
```

```
    k=k+1
```

```
print(k)
```

Ответ: 162

Пример:

Сколько слов длины 6, начинающихся и заканчивающихся согласной буквой, можно составить из букв Г, О, Д? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.

Подсчитываем только те комбинации, которые начинаются и заканчиваются с согласных букв.

Решение на языке Python:

```
k=0
```

```
for x1 in 'ГОД':
```

```
    for x2 in 'ГОД':
```

```
        for x3 in 'ГОД':
```

```
            for x4 in 'ГОД':
```

```
                for x5 in 'ГОД':
```

```
                    for x6 in 'ГОД':
```

```
                        s=x1+x2+x3+x4+x5
```

```
                            if x1=='Г' or x1=='Д':
```

```
                                if x6=='Г' or x6=='Д':
```

```
                                    k=k+1
```

```
print(k)
```

Ответ: 324

Задача (Количество гласных)

Особенность этой задачи в том что слово должно содержать определенное количество гласных букв.

Пример:

Василий составляет 4-буквенные коды из букв Г, А, Ф, Н, И, Й. Каждую букву можно использовать любое количество раз, при этом код не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?

Решение на языке Python:

```
k=0
for x1 in 'ГАФНИЙ':
    for x2 in 'ГАФНИЙ':
        for x3 in 'ГАФНИЙ':
            for x4 in 'ГАФНИЙ':
                s=x1+x2+x3+x4
                if x1!='Й' and s.count('А') + s.count('И') >= 1:
                    k=k+1
```

```
print(k)
```

Ответ: 888

Пример:

Василий составляет 4-буквенные коды из букв Г, Е, Р, О, Й. Каждую букву можно использовать любое количество раз, при этом код

не может начинаться с буквы Й и должен содержать хотя бы одну гласную. Сколько различных кодов может составить Василий?

Решение на языке Python:

```
k=0
for x1 in 'ГЕРОЙ':
    for x2 in 'ГЕРОЙ':
        for x3 in 'ГЕРОЙ':
            for x4 in 'ГЕРОЙ':
                s=x1+x2+x3+x4
                if x1!='Й' and s.count('Е') + s.count('О') >= 1:
                    k=k+1
print(k)
```

Ответ: 446

Задача (Перебор чисел)

Особенность этой задачи в том что в этом случае перебор всех возможных комбинаций чисел, при этом стоит помнить что число не может начинаться с нуля.

Пример:

Сколько существует чисел, восьмеричная запись которых содержит 7 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Число не может начинаться с нуля. Поэтому ноль будет исключён из первого цикла.

Первое условие следит за тем, чтобы каждая цифра встречалась один раз в числе. Второе условие подсчитывает количество вариантов, когда первая цифра чётная. Второе условие следит за тем, чтобы чётность и нечётность цифр чередовалась. Третье условие, наоборот, подсчитывает варианты, когда первая цифра нечётная.

Операция % - остаток от деления. Если остаток от деления на 2 равен нулю, то число чётное. Если остаток от деления на 2 равен 1, то число нечётное.

Функция int() преобразует символ в число. Ведь мы работаем именно с символами, а не с реальными числами.

Решение на языке Python:

```
k1=0
k2=0
for x1 in '1234567':
    for x2 in '01234567':
        for x3 in '01234567':
            for x4 in '01234567':
                for x5 in '01234567':
                    for x6 in '01234567':
                        for x7 in '01234567':
                            s=x1+x2+x3+x4+x5+x6+x7
                            if s.count(x1)==1 and s.count(x2)==1 and
s.count(x3)==1 and s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1
and s.count(x7)==1:
                                if int(x1)%2==0 and int(x2)%2==1 and
int(x3)%2==0 and int(x4)%2==1 and int(x5)%2==0 and int(x6)%2==1 and
int(x7)%2==0:
                                    k1=k1+1
                                if int(x1)%2==1 and int(x2)%2==0 and
int(x3)%2==1 and int(x4)%2==0 and int(x5)%2==1 and int(x6)%2==0 and
int(x7)%2==1:
```

```
k2=k2+1
```

```
print(k1+k2)
```

Ответ: 1008

Пример:

Сколько существует чисел, восьмеричная запись которых содержит 6 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Решение на языке Python:

```
k1=0
```

```
k2=0
```

```
for x1 in '1234567':
```

```
    for x2 in '01234567':
```

```
        for x3 in '01234567':
```

```
            for x4 in '01234567':
```

```
                for x5 in '01234567':
```

```
                    for x6 in '01234567':
```

```
                        s=x1+x2+x3+x4+x5+x6
```

```
                            if s.count(x1)==1 and s.count(x2)==1 and s.count(x3)==1  
and s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1:
```

```
                                if int(x1)%2==0 and int(x2)%2==1 and  
int(x3)%2==0 and int(x4)%2==1 and int(x5)%2==0 and int(x6)%2==1:
```

```
                                    k1=k1+1
```

```
                                if int(x1)%2==1 and int(x2)%2==0 and  
int(x3)%2==1 and int(x4)%2==0 and int(x5)%2==1 and int(x6)%2==0:
```

```
                                    k2=k2+1
```

```
print(k1+k2)
```

Ответ: 1008

Задача (Числа, Закрепление)

Особенность этой задачи — что происходит закрепления числа например: все возможные комбинации где только два числа повторяются.

Пример:

Сколько существует четырёхзначных чисел, записанных в восьмеричной системе счисления, в записи которых ровно две одинаковые цифры, причем стоящие рядом ?

Здесь следующий принцип составления условия. Два соседа должны быть равны. Берём одного соседа из пары, где цифры должны быть равны, и комбинируем его с другими цифрами. Пишем уже, чтобы цифры были не равны. Так же прописываем, чтобы две оставшиеся цифры также не были равны. Таким образом, перебираем все варианты.

Решение на языке Python:

```
k=0
for x1 in '1234567':
    for x2 in '01234567':
        for x3 in '01234567':
            for x4 in '01234567':
                if (x1==x2 and x2!=x3 and x2!=x4 and x3!=x4) or (x2==x3
                    and x3!=x1 and x3!=x4 and x1!=x4) or (x3==x4 and x3!=x2
                    and x3!=x1 and x1!=x2):
                    k=k+1
print(k)
```

Ответ: 882

Пример:

Сколько существует четырёхзначных чисел, записанных в десятичной системе счисления, в записи которых ровно две одинаковые цифры, причем стоящие рядом ?

Решение на языке Python:

```
k=0
for x1 in '123456789':
    for x2 in '0123456789':
        for x3 in '0123456789':
            for x4 in '0123456789':
                if (x1==x2 and x2!=x3 and x2!=x4 and x3!=x4) or (x2==x3
                    and x3!=x1 and x3!=x4 and x1!=x4) or (x3==x4 and x3!=x2 and x3!=x1 and
                    x1!=x2):
                    k=k+1
print(k)
```

Ответ: 1944

Задача(Две чётные и две нечётные цифры не стоят рядом)

Пример:

Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 7 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Перебираем 7 разрядов десятичного числа. С нуля число не может начинаться, поэтому из первого цикла удаляем ноль.

Число должно делиться на 5, значит, в последнем цикле оставляем только 0 и 5.

Все цифры различны, поэтому применяем условие, что символ x_1 встречается 1 раз, символ x_2 встречается 1 раз и т.д.

Фраза "две чётные и две нечётные цифры не стоят рядом" обозначает, что цифры должны чередоваться. Например, чётная, нечётная, чётная, нечётная и т.д. (или наоборот).

Направление задаёт именно первая цифра, остальные цифры выстраиваются по ней.

Проверить чётность/нечётность цифр просто, мы проверим существует ли конкретный символ в строке из чётных или нечётных цифр.

У нас два равноправных случая: когда первая цифра чётная, и когда первая цифра нечётная.

Решение на языке Python:

```
k=0
for x1 in '123456789':
    for x2 in '0123456789':
        for x3 in '0123456789':
            for x4 in '0123456789':
                for x5 in '0123456789':
                    for x6 in '0123456789':
                        for x7 in '05':
                            s=x1+x2+x3+x4+x5+x6+x7
                            if s.count(x1)==1 and s.count(x2)==1 and
s.count(x3)==1 and s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1
and s.count(x7)==1:
```

```
if x1 in '02468' and x2 in '13579' and x3 in '02468'
and x4 in '13579' and x5 in '02468' and x6 in '13579' and x7 in '02468':
```

```
k=k+1
```

```
if x1 in '13579' and x2 in '02468' and x3 in '13579'
and x4 in '02468' and x5 in '13579' and x6 in '02468' and x7 in '13579':
```

```
k=k+1
```

```
print(k)
```

Ответ: 2880

Пример:

Сколько существует чисел, делящихся на 5, десятичная запись которых содержит 6 цифр, причём все цифры различны и никакие две чётные и две нечётные цифры не стоят рядом.

Решение на языке Python:

```
k=0
```

```
for x1 in '123456789':
```

```
    for x2 in '0123456789':
```

```
        for x3 in '0123456789':
```

```
            for x4 in '0123456789':
```

```
                for x5 in '0123456789':
```

```
                    for x6 in '05':
```

```
                        s=x1+x2+x3+x4+x5+x6
```

```
                            if s.count(x1)==1 and s.count(x2)==1 and
s.count(x3)==1 and s.count(x4)==1 and s.count(x5)==1 and s.count(x6)==1:
```

```
                                if x1 in '02468' and x2 in '13579' and x3 in '02468'
and x4 in '13579' and x5 in '02468' and x6 in '13579':
```

```
                                    k=k+1
```

```
if x1 in '13579' and x2 in '02468' and x3 in '13579'  
and x4 in '02468' and x5 in '13579' and x6 in '02468':
```

```
k=k+1
```

```
print(k)
```

Ответ: 1296

Задача (Со списками, классическая)

В таких задачах дан список и нам нужно найти слово на каком то месте.

Пример:

Все 4-буквенные слова, составленные из букв А, Е, И, О записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААА

2. АААЕ

3. АААИ

4. АААО

5. ААЕА

...

Запишите слово, стоящее на 248-м месте от начала списка.

Важно: Нужно буквам присваивать цифры именно в алфавитном порядке, потому что буквы могут дать в "перепутанном порядке" (например Е, А, И, О), и тогда ничего не получится.

Переменная k это порядковый номер слов. Мы подсчитываем все слова без какого-либо условия.

Каждая переменная x перебирает буквы, которые используются для составления слов. Важно: буквы нужно перебирать в том порядке, в котором они идут в правом столбце списка, приведенного в задаче.

Мы составляем 4 вложенных цикла, т.к. длина слов равна 4-м буквам.

В переменной s формируется каждое слово из списка. Если порядковый номер равен 248, то печатаем слово s .

Решение на Python:

```
k=0
for x1 in 'АЕИО':
    for x2 in 'АЕИО':
        for x3 in 'АЕИО':
            for x4 in 'АЕИО':
                s=x1+x2+x3+x4
                k=k+1
                if k==248:
                    print(s)
```

Ответ: ООЕО

Пример:

Все 4-буквенные слова, составленные из букв А, Д, И, Р записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААА
2. АААД
3. АААИ

4. АААР

5. ААДА

...

Запишите слово, стоящее на 222-м месте от начала списка.

Решение на Python:

```
k=0
```

```
for x1 in 'АДИР':
```

```
    for x2 in 'АДИР':
```

```
        for x3 in 'АДИР':
```

```
            for x4 in 'АДИР':
```

```
                s=x1+x2+x3+x4
```

```
                k=k+1
```

```
                if k==222:
```

```
                    print(s)
```

Ответ: РДРД

Задача (Со списками, классическая, решение с помощью рассуждений)

Все 4-буквенные слова, составленные из букв А, Е, И, О записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААА

2. АААЕ

3. АААИ

4. АААО

5. ААЕА

...

Запишите слово, стоящее на 248-м месте от начала списка.

Решение:

Обозначим условно **А** - 0, **Е** - 1, **И** - 2, **О** - 3.

Важно: Нужно буквам присваивать цифры именно в том порядке, в котором они идут в самом правом столбце, потому что буквы могут дать в "перепутанном порядке" (например Е, А, И, О), и тогда ничего не получится.

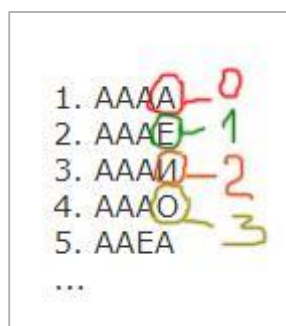


Рисунок 5. Кодирование символов

Теперь по рис. 5 запишем список с помощью цифр.

1. 0000

2. 0001

3. 0002

4. 0003

5. 0010

...

Получился обычный счёт в **четверичной системе!!** (всего используются 4 цифры: 0, 1, 2, 3). А слева нумерация показывает соответствие нашей десятичной системе. Но **все числа десятичной системы** в этой таблице соответствия **сдвинуты на 1**, ведь мы должны были начать с нуля.

Нас просят записать слово стоящее на 248, т.е. если была обычная таблица соответствия чисел десятичной системы и четверичной системы, слово стоящее на 248 месте, находилось бы на 247 (248 - 1) месте. Значит, наше искомое **четверичное число** соответствует 247 в десятичной системе.

Переведём число 247 в четверичную систему (Рис.6)!

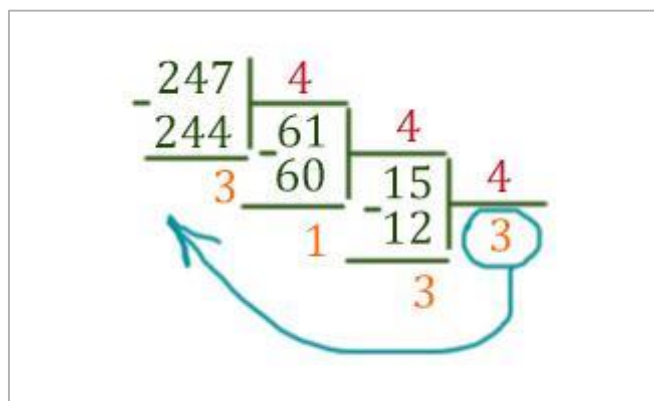


Рисунок 6. Перевод числа в четверичную систему

Получилось число **3313₄** в четверичной системе. Сделаем обратное декодирование в буквы. Таким образом, ответ будет **ООЕО**.

Задача (Классика, Укажите номер слова)

Пример:

Все 5-буквенные слова, составленные из букв А, Р, У, К записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. ААААК
3. ААААР
4. ААААУ
5. АААКА

.....

Укажите номер слова УКАРА

В переменную `if` присваиваем `s ==` нужное нам слово в данном случае `if s == 'УКАРА'`

Решение на Python:

```
k=0
```

```
for x1 in 'АКРУ':
```

```
    for x2 in 'АКРУ':
```

```
        for x3 in 'АКРУ':
```

```
            for x4 in 'АКРУ':
```

```
                for x5 in 'АКРУ':
```

```
                    s=x1+x2+x3+x4+x5
```

```
                    k=k+1
```

```
                    if s=='УКАРА':
```

```
                        print(k)
```

Ответ:841

Пример:

Все 4-буквенные слова, составленные из букв П, Р, В, Д, А, записаны в алфавитном порядке и

пронумерованы. Вот начало списка:

1. АААА

2. АААВ

3. АААД

4. АААП

5. АААР

6. ААВА

...

Найдите номер первого слова в этом списке, которое не содержит гласных и одинаковых букв.

В этом задании необходимо составить нужное нам слово. Раз это слово первое слова в этом списке, которое не содержит гласных и одинаковых букв. Значит если оно первое, то необходимо чтобы оно начиналось с первой согласной буквы в нашем случае это буква 'В' буквы не должны повторяться, далее согласные буквы составляем в алфавитном порядке. Нужно нам слово 'ВДПР'.

Решение на Python:

```
k=0
for x1 in 'ВДПР':
    for x2 in 'ВДПР':
        for x3 in 'ВДПР':
            for x4 in 'ВДПР':
                s=x1+x2+x3+x4
                k=k+1
                if s=='ВДПР':
                    print(k)
```

Ответ: 195

После каждой группы есть тест на решение таких заданий.

Перейдем теперь к заданию №15 “Элементы математической логики”

Изначально ученикам предлагается учебное пособие О. Ю. Агарева, Ю. В. Селиванов ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ. После разбиение задач на группы:

- Задача (Неравенство, одна переменная);
- Задача (Неравенство, две переменные);
- Задача (Функция ДЕЛ);
- Задача (Поразрядная конъюнкция);
- Задача (числовая прямая);
- Задача (Числовая прямая, закрепление).

Задача (Неравенство, одна переменная) решение с помощью рассуждений

Какое количество натуральных чисел удовлетворяет логическому условию:

$$\neg(X^2 \geq 9) \vee \neg((X < 7) \vee (X \geq 10)) ?$$

Натуральные числа - это целые, положительные числа. Например: 1, 2, 3, 4, и т. д.

Преобразуем первое выражение $\neg(X^2 \geq 9) = (X^2 < 9)$. Отрицание внесли в скобки. В этом случае знак, который находится в скобках, нужно поменять на противоположный.

Важно: Если было строгое неравенство, то оно станет нестрогим, и наоборот, если было неравенство нестрогим, то оно станет строгим.

Получается, что выражение $(X^2 < 9)$ будет истинно только при двух значениях: $X = 1, X = 2$.

Во втором выражении $\neg((X < 7) \vee (X \geq 10))$ удобно применить формулу Де Моргана.

Формула де Моргана:

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

Преобразуем выражение по формуле де Моргана и внесём отрицание в скобки:

$$\neg((X < 7) \vee (X \geq 10)) = \neg(X < 7) \wedge \neg(X \geq 10) = (X \geq 7) \wedge (X < 10)$$

Получилось выражение $(X \geq 7) \wedge (X < 10)$. Между двумя выражениями стоит логическое умножение. Значит, одновременно должны выполняться и первое неравенство, и второе. Таким образом, получается, что подходят три значения для выражения $(X \geq 7) \wedge (X < 10)$. Это $X = 7, X = 8, X = 9$.

Обратимся к самому начальному логическому условию. Там два выражения соединятся логическим сложением. Значит, мы должны объединить те случаи, когда у нас первое выражение становится истинным ($X=1, X=2$), и те случаи, когда второе выражение становится истинным ($X = 7, X = 8, X = 9$).

Получается всего 5 натуральных чисел удовлетворяют изначальному логическому условию.

Ответ: 5

Задача (Неравенство, две переменные)

Пример:

Для какого наибольшего целого неотрицательного числа A выражение

$$(x \geq A) \vee (y \geq A) \vee (x * y \leq 205)$$

тождественно истинно, т.е. принимает значение 1 при любых целых положительных x и y ?

В первом цикле перебираем значения для A . Здесь мы пытаемся подобрать ответ в диапазоне от 0 до 300. Этот диапазон меньше, чем в прошлой задаче. Потому что здесь три вложенных цикла, и если

перебирать числа от 0 до 1000, то программа может работать очень долго. При необходимости можно указать другой диапазон.

Для каждого A устанавливаем счётчик k в ноль.

Затем перебираем все числа в диапазоне от 1 до 300 (включительно) для переменных x и y , тем самым имитируем фразу "для любых x и y ".

Если логическое выражение сработает при каждом значении x и y , то считается, что значение A нам подходит, и в счётчике по окончании вложенных циклов будет значение 90000 ($300 * 300 = 90000$).

Наибольшее число, которое напечатает программа равно 15.

Решение на Python:

```
for A in range(0, 300):
    k=0
    for x in range(1, 301):
        for y in range(1, 301):
            if (x >= A) or (y >= A) or (x * y <= 205):
                k=k+1
        if k==90000:
            print(A)
```

Ответ: 15.

Пример:

Для какого наименьшего целого неотрицательного A выражение

$$(x \geq 27) \vee (2x < 3y) \vee (A > (x+2)(y-3))$$

тождественно истинно, т.е. принимает значение 1 при любых целых неотрицательных значениях переменных x и y ?

В первом цикле перебираем значения для A . Здесь мы пытаемся подобрать ответ в диапазоне от 0 до 400.

Для каждого A устанавливаем счётчик k в ноль.

Затем перебираем все числа в диапазоне от 1 до 400 (включительно) для переменных x и y , тем самым имитируем фразу "для любых x и y ".

Если логическое выражение сработает при каждом значении x и y , то считается, что значение A нам подходит, и в счётчике по окончании вложенных циклов будет значение 160000 ($400 * 400 = 160000$).

Наименьшее число, которое напечатает программа равно 393.

Решение на Python:

```
for A in range(0, 400):
    k=0
    for x in range(1, 401):
        for y in range(1, 401):
            if (x >= 27) or (2*x < 3*y) or (A > (x+2)*(y-3)):
                k=k+1
    if k==160000:
        print(A)
```

Ответ: 393

Задача (Функция ДЕЛ)

Пример:

Обозначим через ДЕЛ(n , m) утверждение «натуральное число n делится без остатка на натуральное число m ». Для какого наибольшего натурального числа A формула

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Здесь мы формируем функцию ДЕЛ (функцию D). Если n делится на m , то функция возвращает Истину, в противном случае функция возвращает Ложь.

Далее решаем примерно так же, как и в прошлых задачах: для каждого числа A перебираем все значения x . Следование расписываем по формуле $A \rightarrow B = \neg A \vee B$.

Наибольшее число здесь получается равно 18.

Решение на Python:

```
def D(n, m):
    if n%m==0:
        return True
    else:
        return False
for A in range(1, 1000):
    k=0
    for x in range(1, 1001):
        if D(x, A) or (not(D(x, 6)) or not(D(x, 9))):
            k=k+1
```

```
if k==1000:
```

```
    print(A)
```

Ответ: 18

Пример:

Обозначим через ДЕЛ(n , m) утверждение «натуральное число n делится без остатка на натуральное число m ». Для какого наименьшего натурального числа A формула

$$(\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 80)$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение на Python:

```
def D(n, m):
```

```
    if n%m==0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
for A in range(1, 1000):
```

```
    k=0
```

```
    for x in range(1, 1001):
```

```
        if ( not (D (x, 2)) or not D (x, 3)) or (x+A >= 80):
```

```
            k=k+1
```

```
if k==1000:
```

```
    print(A)
```

Ответ: 74

Функция ДЕЛ (с помощью рассуждений)

Рассмотрим случай (Рис.7), когда в левой части логического выражения будет 1, а в правой 0. В остальных случаях беспокоится не за что, потому что вся формула будет выдавать истину.



Рисунок 7. Случай, когда в левой части логического выражения будет 1, а в правой 0

Посмотрим, когда в правой части получается ноль. Функция ДЕЛ(x, 6) должна выдавать истину. Т.е. x должен делиться на 6. А функция $\neg \text{ДЕЛ}(x, 9)$ должна выдавать ноль. Т.е. без отрицания ДЕЛ(x, 9) должна выдавать истину. Значит, x так же делится на 9.

$$x \text{ делится на } 6 \Rightarrow x = 2 \cdot 3 \cdot n, n \in \mathbb{N}$$

$$x \text{ делится на } 9 \Rightarrow x = 3 \cdot 3 \cdot n, n \in \mathbb{N}$$

Чтобы выполнялся случай, когда в правой части получается ноль, икс должен быть равен $x = 3 \cdot 3 \cdot 2 \cdot n$ ($n \in \mathbb{N}$). Т.е. получается, что икс должен быть кратен 18.

Т.е. получается, что когда x делится на 18, в правой части логического выражения будет получаться ноль. Чтобы спасти ситуацию, мы должны в левой части логического выражения не получать 1. Следовательно, $\neg \text{ДЕЛ}(x, A)$ должно выдавать ноль. Значит, ДЕЛ(x, A)

должно выдавать 1. Таким образом, приходим к выводу, что A должно равняться 18.

Если получится опасная ситуация, когда x кратен 18, то она будет нейтрализована, ведь в левой части будет получаться ноль.

Ответ: 18

Задача (Поразрядная конъюнкция)

Пример:

Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $14 \& 5 = 11102 \& 01012 = 4$

Для какого наименьшего неотрицательного целого числа A формула

$$x \& 51 \neq 0 \rightarrow (x \& A = 0 \rightarrow x \& 25 \neq 0)$$

тождественно истинна (то есть принимает значение 1 при любом неотрицательном целом значении переменной x)?

Здесь следование преобразовываем по формуле: $A \rightarrow B = \neg A \vee B$. Так же и A, и x неотрицательные числа. Поэтому мы перебираем их диапазон, начиная с нуля. Из-за этого в цикле, который перебирает переменную x, мы устанавливаем верхнюю границы равной 1000, а не 1001. Тогда тоже будет 1000 повторений в этом цикле.

Наименьшее число равно 34.

Решение на Python:

```
for A in range(0, 1000):
```

```
    k=0
```

```
        for x in range(0, 1000):
```

```
if x&51==0 or (x&A!=0 or x&25!=0):
```

```
    k=k+1
```

```
if k==1000:
```

```
    print(A)
```

Пример:

Введём выражение $M \& K$, обозначающее поразрядную конъюнкцию M и K (логическое «И» между соответствующими битами двоичной записи). Определите наименьшее натуральное число A , такое что выражение

$$(X \& 13 = 0) \rightarrow ((X \& 40 \neq 0) \rightarrow (X \& A \neq 0))$$

тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной X)?

Решение на Python:

```
for A in range(0, 1000):
```

```
    k=0
```

```
    for x in range(0, 1000):
```

```
        if (x&13!=0) or ((x&40==0) or (x&A!=0)):
```

```
            k=k+1
```

```
    if k==1000:
```

```
        print(A)
```

Ответ: 32

Поразрядная конъюнкция (с помощью рассуждений)

Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $14 \& 5 = 1110_2 \& 0101_2 = 4$

Для какого наименьшего неотрицательного целого числа A формула

$$x \& 51 \neq 0 \rightarrow (x \& A = 0 \rightarrow x \& 25 \neq 0)$$

тождественно истинна (то есть принимает значение 1 при любом неотрицательном целом значении переменной x)?

Переведём числа 51 и 25 в двоичную систему.

$$51 = 110011_2$$

$$25 = 11001_2$$

Формула будет тождественно ложна (рис.8)

Рисунок 8. тождественно ложная формула

Этого допустить нельзя!

При каком x получается в левой выражении формулы истина? Если у x в двоичном представлении в тех разрядах, где у числа 51 стоят 1, будет хотя бы в одном месте 1.

Рассмотрим правое выражение формулы (Рис.9). Ноль получается в единственном случае:

Рисунок 9. Случай когда в правой части ноль

Рассмотрим выражение $x \& 25 \neq 0$. Чтобы в этом логическом выражении получился ноль, нужно $x \& 25 = 0$. Посмотрим на двоичное представление числа 25. В тех разрядах, где стоят единицы, у x должны быть нули (для $x \& 25 = 0$).

Сформулируем окончательное условие для x , при котором возникает опасность превращение общей формулы в ложь.

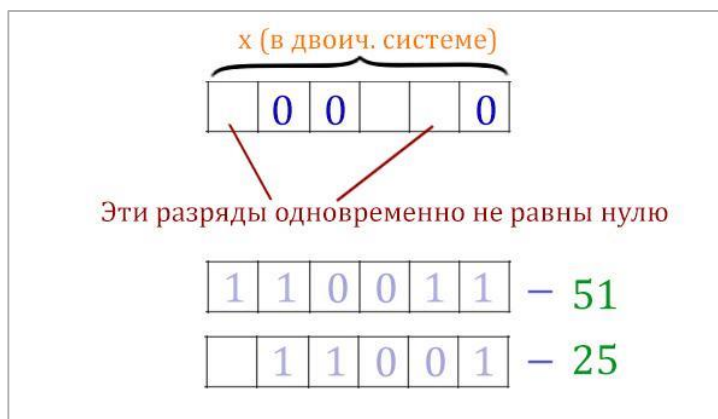


Рисунок 10. Условие для x , при котором возникает опасность

Нам нужно "поломать эту песенку" с помощью $x \& A = 0$. Т.е. нельзя допускать, чтобы это выражение было истинно.

Получается, что $A = 100010_2$. Это наименьшее из возможных число, при котором мы точно себя обезопасим от того, что вся формула будет ложна.

$A = 100010_2$ в десятичной системе будет 34.

Ответ: 34

Задача (числовая прямая)

Для начала ученикам предлагается Видеоролик где рассказывают как решать с помощью рассуждений и как решать с помощью Python. Затем мы приведем примеры решений.

Пример:

На числовой прямой даны два отрезка: $P = [24; 49]$ и $Q = [30; 53]$. Укажите наибольшую возможную длину такого отрезка A , что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любых x .

Решение на языке Python:

```
p1, p2, q1, q2 = 24, 49, 30, 53
P = [i / 10 for i in range(p1 * 10, p2 * 10 + 1)]
Q = [i / 10 for i in range(q1 * 10, q2 * 10 + 1)]
def f(x, A):
    return ((x in A) <= (x in P)) or (x in Q)
A = set([i / 10 for i in range(240, 550)])
for x in [i / 10 for i in range(240, 550)]:
    if not f(x, A):
        A.remove(x)
print(sorted(A))
```

программа выдаст массив чисел от 24 до 53. Ответом будет являться $53 - 24 = 29$

Ответ: 29

Пример:

На числовой прямой даны два отрезка: $P=[25, 42]$, $Q=[1, 98]$.
Найдите наименьшую возможную длину отрезка A , при котором формула

$$(x \in Q) \rightarrow (\neg(x \in P) \wedge (x \in Q) \rightarrow (x \in A))$$

тождественно истинна, то есть принимает значение 1 при любых x .

Решение на языке Python:

```
p1, p2, q1, q2 = 25, 42, 1, 98
P = [i / 10 for i in range(p1 * 10, p2 * 10 + 1)]
Q = [i / 10 for i in range(q1 * 10, q2 * 10 + 1)]
def f(x, A):
```

```

return (x in Q) <=( not (x in P) and (x in Q) <= (x in A))
A = set()
for x in [i / 10 for i in range(10, 1000)]:
    if not f(x, A):
        A.add(x)
    print(sorted(A))

```

программа выдаст массив чисел от 1 до 98. Ответом будет являться 98-1=97

Ответ: 97

Созданный электронный курс позволит ученикам изучить теорию, рассмотреть примеры решений задач, практические упражнения позволят закрепить знания. Что способствует увеличить успех в написании ЕГЭ по информатике.

2.2 Методические рекомендации по работе с теорией и практикой задач единого государственного экзамена, относящихся к математическим объектам информатики

Ученики сдают ЕГЭ по информатике для получения аттестата о среднем (полном) общем образовании и поступления в высшие учебные заведения. При поступлении в вузы баллы ЕГЭ по информатике могут являться одним из критериев отбора в рамках конкурса, в зависимости от направления подготовки. Также, успешная сдача ЕГЭ по информатике может повысить шансы на получение работы в компьютерных компаниях и на других вакансиях, где требуется знание этого предмета.

Для улучшения результатов ЕГЭ у обучающихся лица №6 создан электронный курс на платформе LMS Moodle на котором ученики могут самостоятельно готовится к единому государственному экзамену. Курс

будет рассчитан на подготовку к заданиям с математическим аспектом таким как комбинаторика, математическая логика.

Задание номер 8 в ЕГЭ по информатике представляет собой задачу по измерению количества информации. В задании требуется написать программу на языке программирования Python, которая должна работать с входными данными, выполнить определенные операции и вернуть результат в выходном файле.

Одна из особенностей этого задания в том, что для успешного решения надо не только иметь навыки программирования на языке Python, но и уметь ориентироваться в алгоритмах и структурах данных. В задании могут быть также требования к обработке больших объемов данных, поэтому необходимо уметь оптимизировать код и использовать эффективные алгоритмы.

Кроме того, задание 8 в ЕГЭ по информатике довольно длинное и сложное, и выполнение задания может занять много времени. Поэтому важно уметь планировать свое время и контролировать его в процессе выполнения задания.

Также в задании может быть требование на систематизацию полученных результатов и вывод информации о работе программы.

Изначально на курсе дается теория как решать задачи вручную и с помощью языка программирования Python.

Содержание:	
Объяснение темы	
Измерение количества информации	
Двоичное кодирование сообщений (равновероятностные события)	
Количество различных сообщений в алфавите разной мощности	
Количество сообщений при различном вхождении (встречаемости) букв	
Дополнительные формулы	
Тренировочные задания 8 ЕГЭ по информатике и их решение	
Сколько вариантов шифра или кодовых слов	
Перестановка букв в слове (каждая буква 1 раз)	
Сколько существует n -значных чисел, записанных в m -ной системе счисления	
Список в алфавитном порядке	
Вероятность событий	

Рисунок 11. Содержание теории электронного курса

Далее дается видео как решать задачи вручную и с помощью языка программирования, затем идет тестирования на знание теории (Рис. 12), если ученик знает теорию он может ее пропустить и перейти сразу к практике. Особенности этого курса что задачи разбиты на блоки (Рис. 13) и типы задач.








-  Теория. Как решать 8 задание
-  Ссылка на видео как еще можно еще написать программу
-  Тест по теории
-  Задание 8 (Классическая)
-  Задание 8 (классическая)

Рисунок 12. Пример как выглядит курс

-  Задание 8 БУКВЫ

-  Задание 8 Числа


-  Задание 8 Список

Рисунок 13. Блоки задач

И каждый блок углубляется в решение задач с помощью программирования на Python после каждой темы проводится практическое тестирования, ученикам предлагается решить похожие задачи которые они изучили в теории.

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 4-буквенные слова, в которых есть только буквы A, B, C, D, X, причём буква X появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Ответ:

Рисунок 15. Пример практического тестирования

После изучения задания №8 начинаем изучать задание №15

Задание номер 15 в ЕГЭ по информатике является одним из самых сложных в тесте и предназначено для проверки глубоких знаний учеников в области программирования и алгоритмизации. Оно состоит из двух частей: первая часть – это текстовое условие задачи, а вторая – код программы, осуществляющей решение задачи.

Задание 15 проверяет умение ученика создавать программу на языке программирования Python, используя знания и навыки по следующим темам:

- работа со строками и списками;
- создание и использование функций;
- решение задач с использованием циклов и условных операторов;
- работа с файлами и операционной системой;

- использование алгоритмических стратегий, таких как поиск в ширину или глубину.

В задании 15 важно понимать условие задачи, правильно организовать алгоритм решения и написать эффективный код, который соответствует заданной спецификации. Необходимо также уметь отлаживать программу и проверять ее работу на тестовых данных.

В целом, задание 15 требует от учеников высокой концентрации и умения работать быстро и точно, чтобы выполнить его в отведенное время и получить высокий балл. Оно сложно, но решаемо для тех, кто готов к тщательной подготовке и систематическому изучению материала.

Работа с теорией и практикой задач единого государственного экзамена по математическим объектам информатики является важным этапом подготовки школьников к сдаче экзамена.












Первоначально необходимо закрепить теоретические знания по всем темам, которые могут быть представлены в задачах. Важным этапом будет разделение всех тем на основные и дополнительные. Основные темы должны быть изучены более тщательно, чтобы каждый ученик смог полностью понимать все условия задач.

Далее необходимо постоянно тренироваться на различных типах заданий. В этом случае необходимо учитывать наиболее часто встречающиеся типы задач.

Электронный курс изначально знакомит ученика с элементами математической логики затем проводится тестирование на знание элементов математической логики, если ученик уже знаком с элементами математической логики, то тест можно пропустить, далее идет погружение в теорию решения отдельных задач как с помощью

рассуждений так и с помощью языка программирования Python, далее следует практическое тестирование на решения каждого вида задач.

Задание 15

-  Учебное пособие по мат-логике
-  Видео задание №15
-  Тест на знание логических операций
-  Задача (Неравенство, одна переменная) решение с помощью рассуждений
-  Задача (Неравенство, одна переменная)
-  Задача (Неравенство, две переменные)
-  Неравенство, две переменные (с помощью рассуждений).
-  Неравенство, две переменные
-  Задача (Функция ДЕЛ)
-  Функция ДЕЛ (с помощью рассуждений)
-  Функция ДЕЛ

Активаци
Чтобы актив
----- "П"

Рисунок 16. Задание №15

Таким образом, рекомендуется работать с теорией и практикой в равной степени. Важно, чтобы каждый ученик не только знал теоретический материал, но и умел применять его на практике в решении задач. От этого зависит не только результат на экзамене, но и успех в дальнейшей жизни.

Условия для работы с материалами электронного курса:

1. Ученики уже умеют программировать на языке Python. Данный курс не учит азам программирования, он только готовит к решению предстоящих задач на ЕГЭ по информатике.

2. При подборе материала учитывалось чтобы он был понятен и не запутал ученика, а также учитывались требования к ЕГЭ по информатике.

3. Электронный курс прост и удобен в использовании.

4. Курс наполнен разнообразным материалом: теорией, примерами задач и их решением, видеоматериал, тесты, практические задания. Это позволит учащимся изучать материал в разных форматах.

5. Электронный курс будет постоянно обновляться и дополняться согласно требованиям к ЕГЭ по информатике.

6. Также в будущем планируется его расширение по охвату задач связанных с математическими аспектами в информатике.

2.3 Описание процедуры и результатов частичной апробации учебного курса для подготовки к ЕГЭ

Для дополнительной подготовки к ЕГЭ по информатике был разработан и создан электронный курс на платформе Moodle на базе лицея №6 “Перспектива” города Красноярск. Для апробации учебного курса была собрана группа учеников из 10 класса в количестве 10 человек. Ученики уже начали готовиться к ЕГЭ по информатике поэтому они обладали навыком работы с языком программирования Python. Изначально был проведен мини-опрос.

Опрос

1) Знакомы ли вы с языком программирования Python?

а) да б) больше да, чем нет в) нет г) больше нет, чем да

2) Умеете решать задание №8 в ЕГЭ по информатике?

а) да б) больше да, чем нет в) нет г) больше нет, чем да

3) Если умеете решать задание №8 в ЕГЭ по информатике как часто допускаете ошибки?

а) часто б) иногда в) редко г) не допускаю ошибок

4) Если не умеете решать №8 в ЕГЭ по информатике хотели бы научиться?

а) да б) больше да, чем нет в) нет г) больше нет, чем да

5) Если умеете решать задание №8 в ЕГЭ по информатике какие задания вызывают наибольшее затруднение?

а) задание с буквами б) задание с числами в) задание со списками

В опросе участвовало 10 человек, после опроса были получены такие результаты:

Результат опроса

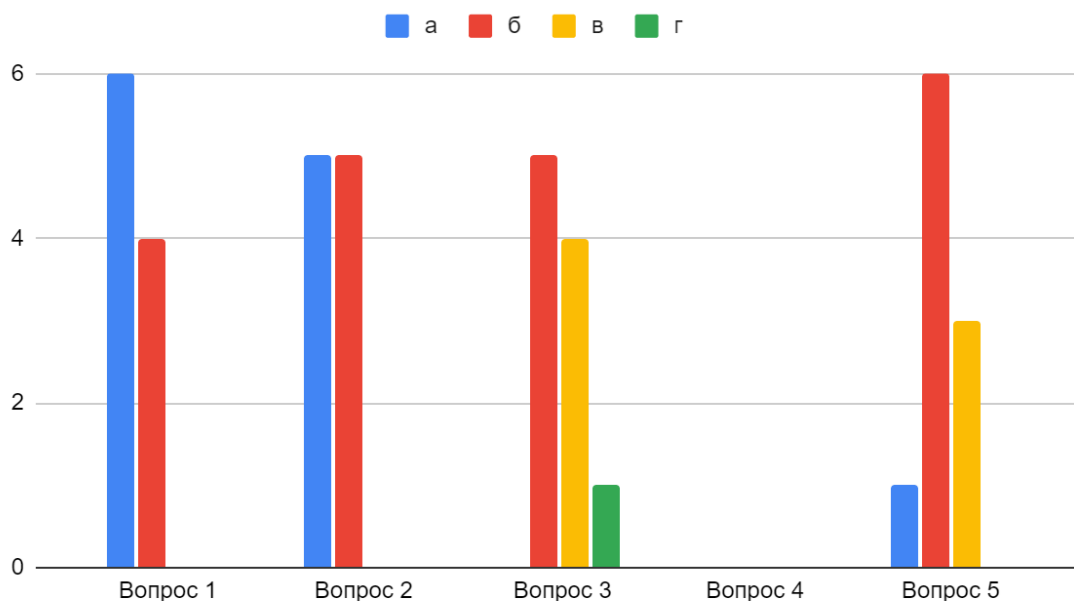


Рисунок 17. Результаты опроса

Далее учеников познакомили с электронным курсом, его структурой и содержанием. Обучающиеся сами для себя подбирали траекторию обучения при самостоятельной подготовке к решению заданий №8. Варианты самостоятельной подготовки:

- 1) Ученик изучает теорию “Измерения количества информации” смотрит видео материал, изучает как решать задание с помощью рассуждений и как решать задание на языке программирования Python.
- 2) Ученик изучает теорию “Измерения количества информации” смотрит видео материал, изучает как решать задание на языке программирования Python.
- 3) Ученик уже знает теорию и наращивает задания на языке программирования Python.

После двухнедельного самостоятельного изучения было введено контрольное тестирование где 1-2 задание (Классическая), 3-4 задание (Перебор чисел), 5-6 со списками (Классическая) которое показало следующие результаты:

Таблица 2. Результаты контрольного тестирования

	Ученик 1	Ученик 2	Ученик 3	Ученик 4	Ученик 5	Ученик 6	Ученик 7	Ученик 8	Ученик 9	Ученик 10
Задание 1	1	1	1	1	0	1	1	0	1	1
Задание 2	1	0	1	0	1	0	1	1	1	1
Задание 3	1	1	1	0	1	1	0	1	0	1
Задание 4	1	1	1	1	1	1	1	1	0	0
Задание 5	0	1	1	1	1	1	0	1	1	0
Задание	1	0	1	1	0	1	1	1	1	1

Процент выполнения заданий

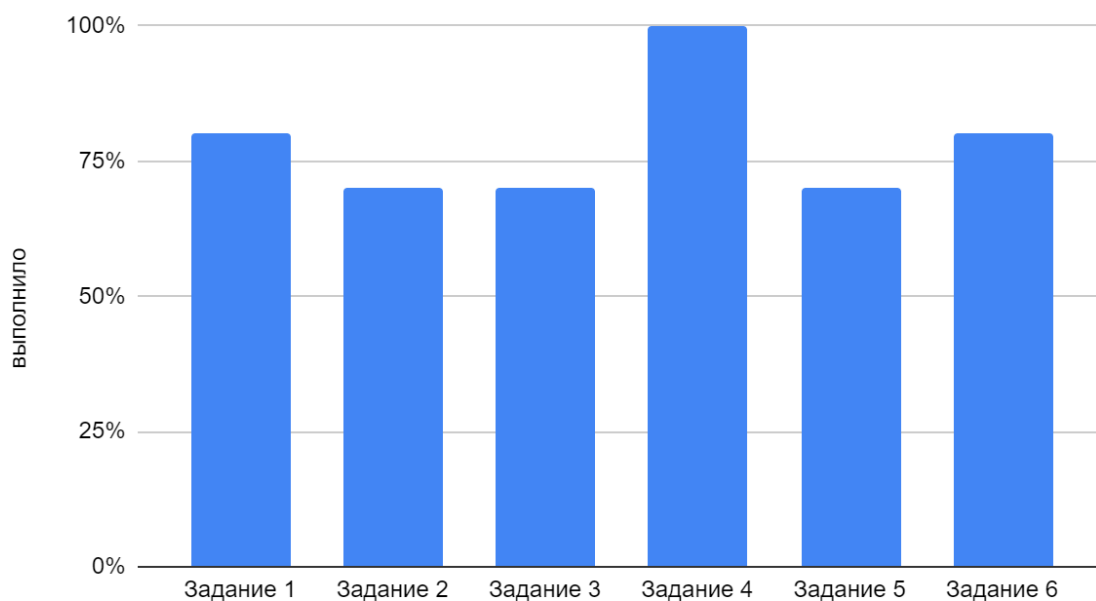


Рисунок 18. Процент выполнения заданий

После контрольного тестирования по решению заданий №8 был проведен мини опрос по электронному курсу.

Опрос по курсу

1) Как вы считаете данный курс полезен при подготовки к ЕГЭ по информатике?

а) да; б) нет; в) не знаю; г) свой вариант ответа: _____

2) Удобен ли курс в использование?

а) да; б) нет; в) не знаю; г) свой вариант ответа: _____

3) Понятен ли материал курса?

а) да; б) нет; в) не знаю; г) свой вариант ответа: _____

4) Было ли достаточно примеров и задач для закрепления знаний?

а) да; б) нет; в) не знаю; г) свой вариант ответа: _____

5) Какую бы поставили оценку данному курсу?

а) отлично; б) хорошо; в) удовлетворительно;

г) неудовлетворительно

6) Что на ваш взгляд можно было улучшить?

В опросе участвовало 10 человек, после опроса были получены такие результаты:

Результат опроса по электронному курсу

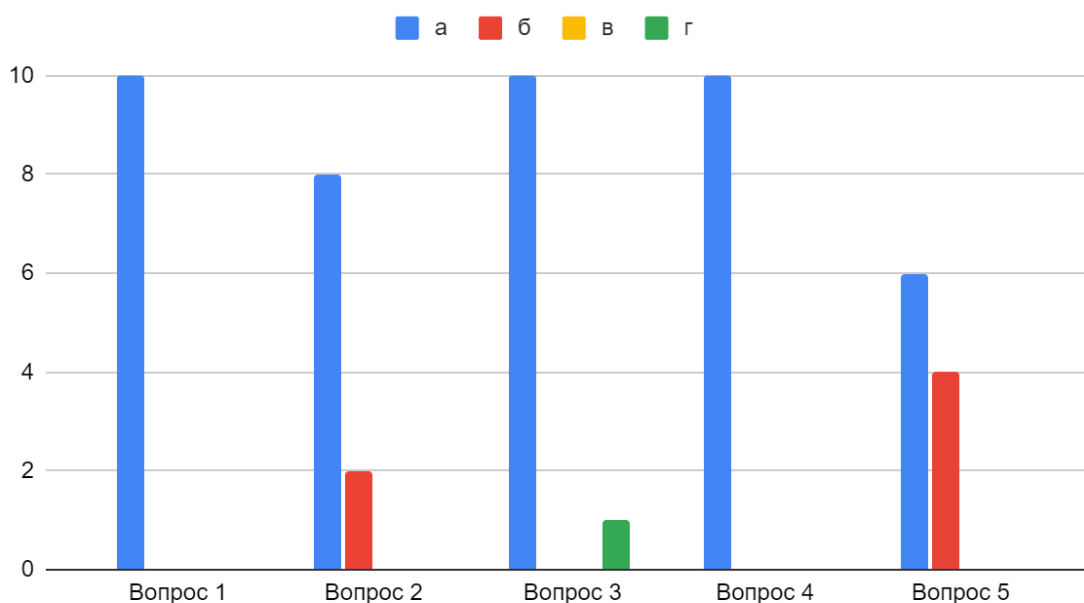


Рисунок 19. Результат опроса по электронному курсу

На вопрос №6 были получены следующие ответы:

Добавить больше примеров решения задач, добавить больше видеоматериалов, добавить другие способы решения задач на языке программирования Python.

Все пожелания школьников были услышаны, курс в будущем будет только расширяться и периодически обновляться.

Из результатов частичной апробации и опросы можно сделать вывод: что курс действительно полезен при подготовки к ЕГЭ и с помощью его можно самостоятельно подготовиться к написанию ЕГЭ по информатике.

Вывод по главе 2

1) Для улучшения результатов ЕГЭ у обучающихся лица №6 создан электронный курс на платформе LMS Moodle на котором ученики могут самостоятельно готовится к единому государственному экзамену. Курс будет рассчитан на подготовку к заданиям с математическим аспектом таким как комбинаторика, математическая логика. Для удобства изучения задания были классифицированы по типам. С возможностью самостоятельной подготовки к ЕГЭ по информатике.

2) Из результатов частичной апробации и опросы можно сделать вывод: что курс действительно полезен при подготовки к ЕГЭ и с помощью его можно самостоятельно подготовиться к сдачи ЕГЭ по информатике. Курс в будущем будет развиваться и становиться только лучше и расширять свои способности.

Заключение

ЕГЭ - важный инструмент образовательного процесса. Он позволяет сделать общий срез знаний учеников, и становится главным показателем при поступлении в высшее учебное заведение.

В результате выполнения задач, поставленных в выпускной квалификационной работе были получены следующие результаты:

1) Moodle — система управления образовательными электронными курсами. Это свободное приложение, позволяющее создавать онлайн-ресурсы для обучения. Moodle позволяет ученикам работать над своим образованием в любое удобное для них время. На платформе поддерживается мобильная версия, то есть ученики могут работать со смартфона или планшета. Информационные материалы могут быть представлены в разных форматах, например, текст, аудио или видео.

2) На основе анализа ФИПИ за 2021-22 гг. можно сделать выводы, что задание №8 выполнили успешно лишь 49,6% и 30% экзаменуемых соответственно. Задание №15 выполнили успешно 36,5% и 40% экзаменуемых. Следовательно создание электронного курса для подготовки к ЕГЭ с материалами по заданиям №8 и №15 актуально.

3) Электронный курс для учителя и обучающегося позволяет сформировать индивидуальный маршрут подготовки к экзамену по информатике и включает в себя все необходимые материалы, которые необходимы для успешного написания ЕГЭ.

4) Для улучшения результатов ЕГЭ у обучающихся лица №6 создан электронный курс на платформе LMS Moodle на котором ученики могут самостоятельно готовится к единому государственному экзамену. Курс будет рассчитан на подготовку к заданиям с

математическим аспектом таким как комбинаторика, математическая логика. Для удобства изучения задания были классифицированы по типам. С возможностью самостоятельной подготовки к ЕГЭ по информатике.

5) Из результатов частичной апробации и опросы можно сделать вывод: что курс действительно полезен при подготовки к ЕГЭ и с помощью его можно самостоятельно подготовиться к сдачи ЕГЭ по информатике. Курс в будущем будет развиваться и становиться только лучше и расширять свои способности.

Все поставленные задачи выполнены, цель достигнута.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Алгебра логики [Электронный ресурс] URL: <https://studfile.net/preview/3611792/page:3/> (дата обращения: 15.05.2023)
2. Босова Л.Л. Босова А.Ю. Информатика: учебник 10 класс базовый уровень/ Босова Л.Л. Москва БИНОМ. Лаборатория знаний, 2016.-288 с.
3. Босова Л.Л. Босова А.Ю. Информатика: учебник 11 класс базовый уровень/ Босова Л.Л. Москва БИНОМ. Лаборатория знаний, 2017.-256 с.
4. Все, что нужно знать о ЕГЭ информатика [Электронный ресурс] URL: <https://media.foxford.ru/articles/ege-informatics> (дата обращения: 19.06.2023)
5. Джемалдинова М.Ю. Гузуева Э.Р. Роль обучения программированию в подготовке к ЕГЭ по информатике // Профессионально-педагогическое образование: состояние и перспективы / Изд-во ООО «Директ-Медиа», 2021. С. 261-263.
6. ЕГЭ по информатике [Электронный ресурс] URL: <https://kpolyakov.spb.ru/school/ege/gen.php?action=viewAllEgeNo&egeId=8&cat42=on&cat43=on&cat145=on>
7. ЕГЭ по информатике 2022 - Задание 8 (Супер-комбо) [Электронный ресурс] URL: https://code-enjoy.ru/ege_po_informatike_2022_zadanie_8_super_combo/ (дата обращения: 15.05.2023)
8. ЕГЭ по информатике 2022 - Задание 15 (Простым языком) [Электронный ресурс] URL: https://code-enjoy.ru/ege_po_informatike_2022_zadanie_15_prostim_yazikom/ (дата обращения: 15.05.2023)

9. Задания 8. Перебор слов и системы счисления. Подсчет количества слов с ограничениями [Электронный ресурс] URL: https://inf-ege.sdangia.ru/test?filter=all&category_id=379 (дата обращения: 18.06.2023)

10. Задание 15 ЕГЭ по информатике [Электронный ресурс] URL: <https://ctege.info/informatika-teoriya-ege/zadanie-15-ege-po-informatike.html> (дата обращения: 18.06.2023)

11. Задание 15. Преобразование логических выражений [Электронный ресурс] URL: <https://klub-kod.ru/ege-inf/sbornik-ege-2023/zadanie-15-2023> (дата обращения: 15.05.2023)

12. Информатика. Подготовка к ЕГЭ 2023. [Электронный ресурс] URL: <https://www.at.alleng.org/edu/comp2.htm> (дата обращения: 15.05.2023)

13. Какова важность математики в информатике? [Электронный ресурс] URL: <http://medicina-treat.ru/kakova-vazhnost-matematiki-v-informatike/> (дата обращения: 18.06.2023)

14. Кондратьев Д.В. Рекомендации по использованию языка программирования Python при подготовке к ЕГЭ по информатике// Современные физика, математика, цифровые и нанотехнологии в науке и образовании/ БГПУ им. М. Акмуллы, 2022. С. 128-130.

15. Обзор системы дистанционного обучения Moodle [Электронный ресурс] URL: <https://www.mirapolis.ru/blog/obzor-moodle/> (дата обращения: 05.05.2023)

16. О возможностях Moodle для образовательной организации, преподавателей и обучающихся [Электронный ресурс] URL: <https://training-net.ru/articles/about-moodle.html> (дата обращения: 19.06.2023)

17. Особенности подготовки учителей к созданию и использованию электронных средств обучения [Электронный ресурс] URL: <https://multiurok.ru/index.php/files/osobennosti-podgotovki-uchitelei-k-sozdaniiu-i-isp.html> (дата обращения: 18.06.2023)

18. Основные возможности системы дистанционного обучения Moodle [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/osnovnyye-vozmozhnosti-sistemy-dstantsionnogo-obucheniya-moodle> (дата обращения: 18.06.2023)

19. Подоляко А.А. Подвесовская М.А Локальное приложение для подготовки индивидуальных заданий по разделу “Основы математических систем” дисциплины “Информатика” // Свидетельство о государственной регистрации программы для ЭВМ / БГТУ 2018. С. 38-41.

20. Разбор 8 задания ЕГЭ по информатике и теории количества информации [Электронный ресурс] URL: <https://labs-org.ru/ege-8/> (дата обращения: 08.05.2023)

21. Разбор основных типов 15 задания в ЕГЭ по информатике [Электронный ресурс] URL: <https://dzen.ru/a/YZvcQCMhRzpOOt7O> (дата обращения: 19.06.2023)

22. Роль математики в эффективном изучении информатики [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/rol-matematiki-v-effektivnom-izuchenii-informatiki> (дата обращения: 19.06.2023)

23. Связь математики и информатики [Электронный ресурс] URL: <https://multiurok.ru/files/sviaz-matematiki-i-informatiki.html> (дата обращения: 15.05.2023)

24. Система электронного обучения Moodle: обзор возможностей и функционала [Электронный ресурс] URL: <https://lmslist.ru/free-sdo/obzor-moodle/> (дата обращения: 19.06.2023)

25. Теория и решение 15 задания по информатике [Электронный ресурс] URL: <https://labs-org.ru/ege-15/> (дата обращения: 15.05.2023)

26. Технологии организации дистанционного и смешанного обучения с применением LMS "MOODLE" [Электронный ресурс] URL: <https://dodiplom.ru/ready/101371> (дата обращения: 05.05.2023)

27. Технологии организации дистанционного и смешанного обучения с применением "Moodle" [Электронный ресурс] URL: https://knowledge.allbest.ru/pedagogics/3c0a65635b2bd69a5d43b88521216c37_0.html (дата обращения: 05.05.2023)

28. ФГБНУ "Федеральный институт педагогических измерений" [Электронный ресурс] URL: <https://fipi.ru/> (дата обращения: 05.05.2023)

29. ФГОС [Электронный ресурс] URL: <https://fgos.ru/> (дата обращения: 15.06.2023)

30. Фирсова С.А. Особенности решения некоторых задач компьютерного ЕГЭ по информатике // Вестник Таганрогского института имени А.П. Чехова 2021. С. 47-51.

31. Шкуркин А.А. Методические рекомендации по подготовке учащихся к ЕГЭ и ОГЭ по математике и информатике// Вестник научных конференций/ ООО "Консалтинговая компания Юком", 2021. С. 106-108.

32. Элементы комбинаторики и их применение для решения комбинаторных задач ЕГЭ по информатике [Электронный ресурс] URL: <https://infourok.ru/elementy-kombinatoriki-i-ih-primeneniye-dlya-resheniya-kombinatornyh-zadach-ege-po-informatiki-6535825.html> (дата обращения: 15.06.2023)