

Автономная некоммерческая организация высшего образования
«Поволжский православный институт имени Святителя Алексия,
митрополита Московского»

Кафедра педагогики и психологии

Направление подготовки 44.03.01 Педагогическое образование
Направленность (профиль) «Информатика и информационные технологии»

БАКАЛАВРСКАЯ РАБОТА

на тему:

«Программно-методическое обеспечение для подготовки выпускников к ЕГЭ
по информатике (по разделу «Программирование»)»

Выполнила студентка
4 курса группы ИТ-301
очной формы обучения
Коновалова Анжелика
Денисовна

(подпись)

Научный руководитель
Дудина Ирина Павловна
к.п.н., доцент

(подпись)

Допустить к защите:
Заведующий кафедрой

(подпись)

(И.О.Ф)

« ___ » _____ 2020 г.

Тольятти, 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Глава 1. Теоретические основы обучения информатике по разделу "Алгоритмизация и программирование" в 10 – 11 классах старшей школы	10
1.1. Реализация раздела "Алгоритмизация и программирование" в курсе информатики	10
1.2. Методические особенности подготовки к единому государственному экзамену по информатике (по разделу «Алгоритмизация и программирование»)	22
1.3. Электронный учебно-методический комплекс по обучению программированию как компонент программно-методического обеспечения подготовки к единому государственному экзамену по информатике	28
Выводы по главе 1.....	30
Глава 2. Проектирование и реализация электронного учебно-методического комплекса по разделу «Алгоритмизация и программирование».....	31
2.1. Дидактические, программно-технологические и технические характеристики электронного учебно-методического комплекса	31
2.2. Проектирование и реализация теоретико-познавательного модуля электронного учебно-методического комплекса.....	47
2.3. Проектирование и реализация тренингово-практического (коммуникативного) модуля.....	56
2.4. Проектирование и реализация контрольного модуля	64
Выводы по главе 2.....	67
Глава 3. Оценка эффективности разработанного электронного учебно-методического комплекса	68
3.1. Общая характеристика исследования	68
3.2. Методика проведения и результаты педагогического эксперимента.....	70
ЗАКЛЮЧЕНИЕ	79
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	80
ПРИЛОЖЕНИЯ.....	86

ВВЕДЕНИЕ

В настоящее время технологии стремительно развиваются и происходит рост популярности современных IT-специализаций. Поэтому выпускники все чаще выбирают ЕГЭ по информатике как итоговое испытание по окончанию средней школы. Данный предмет сдают те, кто в будущем хочет выбрать профессию, связанную с программированием, информационной защитой и безопасностью и т.д.

ЕГЭ по информатике является одним из самых сложных экзаменов, поэтому для успешной сдачи необходимо детально изучить теоретические материалы по программированию, а также развить практические навыки решения типовых заданий. Для подготовки удобно воспользоваться комплексным подходом – изучение теории подкреплять решением практических заданий.

Целью ЕГЭ является оценка общеобразовательной подготовки по информатике и ИКТ учеников старших классов. Это необходимо для организации отбора достойных абитуриентов на зачисление в учреждение высшего профессионального образования. Следовательно, экзамен могут сдать только те, кто подробно изучил темы базового и углубленного курса по информатике. Данная проблема является одной из наиболее актуальных, для ее решения требуется провести оптимизацию педагогических инструментов в обучении информатики в школах.

На сегодняшний день в ЕГЭ по информатике выпускнику при решении заданий, связанных с алгоритмизацией и программированием, на выбор даются несколько языков программирования: Бейсик, Паскаль, Си и Python. До недавнего времени именно язык программирования Паскаль пользовался наибольшей популярностью, как для изучения, так и для сдачи экзаменов. В последнее время (с 2018 года) на ЕГЭ по информатике школьникам предлагаются варианты условий задач по программированию, реализованных на более современном и перспективном языке программирования Python. Сложившаяся ситуация определила реальную проблему методики обучения

информатике: несмотря на то, что выпускник профильной школы должен уметь программировать на языке Python, на сегодняшний день не существует ни одного официально утвержденного учебника, который готовит к сдаче экзамена на данном языке. Система образования не позволяет реализовать быструю смену обучения языку программирования Паскаль на Python, так как для этого потребуется переподготовка тысячи специалистов и оснащение новыми школьными учебниками. Поэтому во многих школах пока обучение программированию на Python либо не реализуется совсем, либо происходит по инициативе учителя.

Обучением алгоритмизации и программированию в школьном курсе информатики занимались такие педагоги и ученые как Босова Л.Л. и Босова А.Ю. [9], Семакин И.Г. [27, 28, 29, 30], Поляков К.Ю. [21, 22, 23, 24, 25, 5], Бородин М.Н. [8].

Существенный вклад в развитие методики преподавания информатики и программирования внесли Ершов А.П. [11, 12], Григорьев С.Г. и Гриншкун В.В. [10], Лапчик М.П., Хеннер Е.К [15].

Теоретическим и практическим основам обучения алгоритмизации и программированию посвящены работы таких авторов как Окулов С.М. [17, 18], Ушаков Д.М. [32], Зорина Е.М. [13], Самылкина Н.Н. [26].

Большинство образовательных ресурсов направлены на подготовку к ЕГЭ по информатике с использованием языка программирования Pascal, хотя в контрольно-измерительные материалы ЕГЭ все чаще включаются задания, решение которых представлено на более перспективном языке программирования Python.

Актуальность темы исследования определяется необходимостью разработки и использования новых педагогических инструментов, позволяющих проводить обучение программированию в профильных классах в соответствии с современными требованиями ЕГЭ по информатике и ИКТ, т.к. существующие в достаточном количестве профессиональные справочные материалы и онлайн-курсы по изучению языка программирования Python,

ориентированные на IT-специалистов, в большинстве своем не могут быть использованы для обучения школьников.

Цель исследования – разработка, обоснование структуры, содержания и методов реализации программно-методического обеспечения подготовки выпускников средних школ к ЕГЭ по информатике по разделу «Программирование» с использованием языка программирования Python.

Объект исследования – процесс подготовки к ЕГЭ по информатике по разделу «Программирование» в профильных классах средней школы.

Предмет исследования – электронный учебно-методический комплекс по обучению программированию как компонент программно-методического обеспечения подготовки выпускников средних школ к ЕГЭ по информатике.

Задачи:

1. Провести анализ психолого-педагогической, научно-методической и учебной литературы, связанной с проблемой введения профильного обучения на старшей ступени общего образования.

2. Изучить элементы содержания и требования к уровню подготовки выпускников образовательных организаций для проведения ЕГЭ по информатике и ИКТ.

3. Проанализировать учебный материал по программированию за весь курс учебного предмета для подготовки к ЕГЭ.

4. Изучить экзаменационные задания по информатике.

5. Выявить основные проблемы при подготовке к ЕГЭ по разделу «Программирование».

6. Составить учебный план занятий по подготовке учащихся к ЕГЭ по разделу «Программирование».

7. Определить роль и место электронного учебно-методического комплекса по обучению программированию в процессе обучения информатике и сформулировать основные требования к содержанию теоретического материала и практических заданий.

8. Выделить и обосновать этапы разработки и использования ЭУМК по

программированию.

9. Выполнить программную реализацию ЭУМК, разработку: интерфейса пользователя; функционала компонентов на базе созданных сценариев, режимов использования.

10. Разработать методические рекомендации по использованию ЭУМК по программированию при подготовке к ЕГЭ по информатике.

Гипотеза исследования: если использовать разработанный программно-методический комплекс в процессе обучения алгоритмизации и программированию учащихся средних школ, это позволит повысить уровень подготовленности выпускников к единому государственному экзамену по предмету «Информатика и ИКТ».

Методы исследования:

- теоретические: системный анализ отечественной и зарубежной психолого-педагогической, научно-методической литературы по педагогике, психологии, информатике; изучение и анализ нормативных документов в сфере общего образования, критический анализ существующих подходов к обучению информатике и программированию, а также использованию электронных ресурсов по рассматриваемой проблеме;

- эмпирические: обобщение опыта преподавания информатики; анализ содержания учебных программ, планов, пособий, материалов конференций по вопросам обучения программированию в школе; наблюдение, беседа, анкетирование, тестирование учащихся с целью выяснения целесообразности использования предложенной методики и ее эффективности в области развития познавательного потенциала и обучения программированию.

Первая глава курсовой работы отражает теоретическое обоснование проводимого исследования. На основе изученных нормативных документов, научных и методических работ рассмотрены различные подходы к подготовке учащихся к сдаче ЕГЭ по информатике.

Вторая глава включает в себя описание этапов разработки и реализации электронного учебно-методического комплекса для подготовки выпускников средних школ к ЕГЭ по информатике.

Третья глава представляет собой результаты апробации разработанного электронного учебно-методического комплекса для подготовки к ЕГЭ по информатике в практике обучения учащихся средней школы.

В заключении подводятся итоги проделанной работы.

Библиографический список содержит перечень источников информации, использованных при выполнении курсовой работы.

Приложения включают конкретные теоретические и практические материалы, задания и демонстрационные примеры их решения.

Глава 1. Теоретические основы обучения информатике по разделу "Алгоритмизация и программирование" в 10 – 11 классах старшей школы

1.1. Реализация раздела "Алгоритмизация и программирование" в курсе информатики

Обучению алгоритмизации и программирования посвящены работы А.П. Ершова [12, 13]. В своем учебном пособии «Введение в теоретическое программирование» автор указывает, что обучение информатике и программированию строится не только на теории, но и на технологии изучаемой деятельности. Теория является фундаментом изучаемого, в то время как технология – это процесс для достижения цели, он должен быть тесно взаимосвязан с теорией. Когда теория и технология объединяются, то появляются зрелые и тщательно проработанные науки. В настоящее время программирование не относится к таким. По мнению автора, в данной науке между теорией и технологией существует разрыв, в теории наблюдается худосочие и неразвитость математического аппарата, технология носит случайный характер. Ершов А.П. считает, что требуется созревание предпосылок для быстрого превращения программирования в развитую отрасль. Ранее в своих выступлениях автор обобщал результаты изучения программирования и на их основании совершал постановку главных задач отрасли. В первой главе пособия рассмотрен основной аппарат теоретического программирования, описываются этапы постановки задач теории программирования, а также пути и предпосылки их решения. Во второй главе проводится анализ методологии программирования, и систематизируются этапы разработки компьютерных программ. Третья глава направлена на рассмотрение подходов к реализации алгоритмических языков нового поколения.

Статья Ершова А.Р. «Программирование – вторая грамотность» [13] направлена на демонстрацию важности компьютерной грамотности в современном мире. Автор называет программирование процессом, являющимся

неотъемлемой частью жизни людей. Все, о чем думает или говорит человек, запрограммировано на выполнение различных функций. Ершов А.Р. утверждает, что человек – это серьезная генетическая программа, а развитие организма является выполнением данной программы, которая записана в геномном наборе. Таким образом, автор приводит читателя к мысли о том, что человечество находится на пороге к беспредельному развитию и распространению электронной вычислительной техники. Вычислительная машина все больше становится центральным инструментом обработки информации. человеческих жизни и деятельности. Также она открывает новые возможности в образовании и развитии ребенка, позволяя ускорить его интеллектуальное созревание.

В учебнике Григорьева С.Г., Гриншкуна В.В. «Информатизация образования» [10] авторы затрагивают такую важную тему как «информатизация». По их мнению, информатизация образования – это информационная революция, она представляет собой деятельность человек, которая направлена на сбор, хранение, обработку и распространение информации, а применение новейших методик обучения и формирование у детей новых знаний. Она направлена на повышение эффективности обучения школьников и студентов. Так к информатизации образования можно отнести переход от традиционного обучения к образованию в режиме дистанционных онлайн-уроков. Данный вид образования имеет множество преимуществ, так что при обучении программированию целесообразно использовать электронные образовательные ресурсы.

Учебное пособие Лапчика М.П., Семакина И.Г., Хеннера Е.К. «Методика преподавания информатики» [15] посвящено методике преподавания информатики. В данном пособии детально рассказывается о целях, принципах содержания и методов для обучения информатике в средней школе. Предложенную авторами информацию можно использовать как в базовом, та и в профильном курсах информатики. Содержание пособия разделено на две части. Первая часть включает в себя теоретические сведения, общую методику

преподавания информатики. Вторая часть является конкретной методикой по базовому и профильному курсам. Анализ данного пособия полезен для планирования, организации и проведения занятий по информатике, в том числе и раздела «Алгоритмизация и программирование».

В статье Муртузалиевой А.С. «О значимости изучения алгоритмизации и программирования в школьном курсе информатике» [16] рассматриваются особенности содержания раздела «Алгоритмизация и программирования». Также автор дает методические рекомендации по обучению данной тематики. Муртузалиева А.С. рассматривает проблематику предложенного раздела, суть которой заключается в сокращении его содержания. Учащиеся изучают программирование в очень сжатом формате, не углубляясь в тему, а рассматривая поверхностно. Данная проблема может привести к тому, что обучающиеся не будут иметь достаточное развитие логического, аналитического и абстрактного мышления. Автор считает, что проблему можно решить, стимулируя познавательный интерес к разделу «Алгоритмизация и программирование», побуждая интерес учащихся к изучению программирования. Введение Единого государственного экзамена по информатике позволило усилить внимание к программированию у учеников старших классов.

В статье «методы достижения метапредметных результатов в обучении программированию в основной школе» автор Парменова Л.В. [20] считает, что обучение разделу «Алгоритмизация и программирование» позволяет создать благоприятные условия для достижения метапредметных результатов. Программирование помогает стимулировать интерес учащихся к обучению, развить самостоятельность, навыки конструирования, анализа и оценки результатов. По мнению автора, самым главным, что развивает программирование, является формирование логического, конструктивного и творческого мышления. Это способствует повышению интеллектуального развития учащихся. Также алгоритмическое мышление лежит в основе

множества видов деятельности, что позволяет ученикам использовать его в других предметных областях.

Рассмотрим учебники информатики для средней школы, которые вошли в федеральный перечень учебников, рекомендуемых для образовательных учреждений Министерством просвещения [2].

В учебнике «Информатика. Углубленный уровень» для 11 класса [29] на раздел «Алгоритмизация и программирование» Семакин И. Г. отводит 53 часа. Весь учебный материал по алгоритмизации и программированию представлен 17-ю параграфами. Рекомендованное тематическое планирование уроков представлено в таблице 1.

Таблица 1 – Тематическое планирование по учебнику Семакин И. Г. «Информатика. Углубленный уровень» для 11 класса

Тема	Часы	Раздел учебника
4. Структурное программирование		
4.1. Паскаль — язык структурного программирования Элементы языка и типы данных	2	§2.2.1. §2.2.2.
4.2. Операции, функции, выражения	2	§2.2.3.
4.3. Оператор присваивания, ввод и вывод данных	3	§2.2.4.
4.4. Структуры алгоритмов и программ	2	§2.2.5.
4.5. Программирование ветвлений	4	§2.2.6.
4.6. Программирование циклов	4	§2.2.7.
4.7. Вспомогательные алгоритмы и подпрограммы	4	§2.2.8.
4.8. Массивы	4	§2.2.9.
4.9. Типовые задачи обработки массивов	6	§2.2.10.
4.10. Метод последовательной детализации	4	§2.2.11.
4.11. Символьный тип данных	2	§2.2.12.
4.12. Строки символов	5	§2.2.13.
2.2.13. Комбинированный тип данных	6	§2.2.14.
5. Рекурсивные методы программирования		
5.1. Рекурсивные подпрограммы	2	§2.3.1.
5.2. Задача о Ханойской башне	1	§2.3.2.
5.3. Алгоритм быстрой сортировки	2	§2.3.3.
Всего часов:	53	

Анализ учебника, практикума и методического пособия Семакина И.Г. [27, 28, 29, 30] позволил сделать выводы по целостности содержания теоретических и практических материалов в работах автора. В учебнике представлена подробная, детальная и понятная для учеников теоретическая часть. Также автором продумана система закрепления теоретического материала с помощью практикума с заданиями различных уровней сложности. Вариативные задания разнообразны по содержанию и представлены в большом количестве, что позволяет учителю организовать практическую часть, учитывая уровень подготовки каждого ученика.

Учебник Полякова К.Ю. «Информатика. Углубленный уровень» для 10 класса [22] также имеет в содержании подробный разбор раздела «Алгоритмизация и программирование». Автор определяет на его изучение 24 часа. Учебный материал по алгоритмизации и программированию представлен в виде 15-ти параграфов. В таблице 2 представлено тематическое планирование уроков.

Таблица 2 – Тематическое планирование по учебнику Полякова К.Ю. «Информатика. Углубленный уровень» для 10 класса

Тема	Часы	Раздел учебника
Глава 8. Алгоритмизация и программирование		
Алгоритм и его свойства	1	§ 54
Простейшие программы	1	§ 55
Вычисления	1	§ 56
Ветвления	1	§ 57
Циклические алгоритмы	2	§ 58
Процедуры	2	§ 59
Функции	2	§ 60
Рекурсия	2	§ 61
Массивы	2	§ 62
Алгоритмы обработки массивов	2	§ 63
Сортировка	2	§ 64
Двоичный поиск	1	§ 65
Символьные строки	1	§ 66
Матрицы	2	§ 67
Работа с файлами	2	§ 68
Всего часов:	24	

Теоретический материал представлен подробно и доступно для обучающихся. К.Ю. Поляков опубликовал авторский сайт [5] по обучению информатике и для подготовки к ЕГЭ. Сайт содержит большое количество учебных материалов, автор постоянно обновляет информацию и добавляет новые задания и упражнения. К.Ю.Поляков приводит рекомендации, которые позволяют эффективно и быстро решать предложенные задания. Автор представляет подробный разбор задач из тестов ЕГЭ, сравнивает различные способы решения, анализируя при этом их достоинства и недостатки, и находя проблемные моменты. Кроме этого, на сайте представлены всевозможные тесты для контроля процесса подготовки к ЕГЭ, статьи и презентации по теме, разного рода дополнительные материалы, а также видеоматериалы с рекомендациями. Что важно, задания для практики предложены не только на языке программирования Pascal, но и также используются C++ и Python.

Пример поурочного планирования раздела «Алгоритмизация и программирование» в учебнике Семакина И.Г. «Информатика. Базовый уровень» для 10 класса показан в таблицах 3 и 4 (предложено 2 варианта поурочного планирования).

Таблица 3 – Первый вариант поурочного планирования по учебнику Семакина И.Г. «Информатика. Базовый уровень» для 10 класса

Тема	Часы	Раздел учебника
Алгоритмы, структура алгоритмов, структурное программирование	1	§12 – 14
Программирование линейных алгоритмов	2	§15 – 17
Логические величины и выражения, программирование ветвлений	3	§18 – 20
Программирование циклов	3	§21 – 22
Подпрограммы	2	§23
Работа с массивами	4	§24 – 26
Работа с символьной информацией	3	§27 – 28
Всего часов:	18	

Таблица 4 – Второй вариант поурочного планирования по учебнику Семакина И.Г. «Информатика. Базовый уровень» для 10 класса

Тема	Часы	Раздел учебника
Алгоритмы, структура алгоритмов, структурное программирование	2	§12 – 14
Программирование линейных алгоритмов	3	§15 – 17
Логические величины и выражения, программирование ветвлений	4	§18 – 20
Программирование циклов	5	§21 – 22
Подпрограммы	3	§23
Работа с массивами	7	§24 – 26
Организация ввода-вывода с использованием файлов	3	
Работа с символьной информацией	4	§27 – 28
Комбинированный тип данных	4	
Всего часов:	35	

Пособие Бородина М.Н. [8] представляет собой сборник учебных программ и тематических планов по обучению информатике. Сборник направлен как на базовый, так и на углубленный уровни изучения, что может оказать существенную помощь учителям при планировании уроков.

Отдельно можно выделить работу учителя информатики МБОУ СОШ №18 Данилова Н.С. (по учебникам Семенова А.Л., Босовой Л.Л.) [4]. Педагог предлагает тематическое планирование по разделу «Алгоритмизация и программирование» для 10 класса, рассчитанное на 21 час (Таблица 5).

Таблица 5 – Тематическое планирование Данилова Н.С.

Глава 4. Алгоритмизация и основы программирования. (21 час).	Часы
Знакомство с системой Turbo Pascal 7.0 Управление окнами текстового редактора. Дисковые операции. Текстовый редактор	2
Программирование алгоритмов линейной структуры	2
Численные переменные и типы данных	1
Логические переменные	1
Управляющие конструкции языка. Условный и составной операторы	1
Управляющие конструкции языка. Оператор выбора	1
Управляющие конструкции языка. Оператор безусловного перехода	1
Графические возможности. Графический режим. Система координат	1
Графические возможности. Абсолютные и относительные координаты. Заполненные фигуры.	1
Циклические конструкции. Оператор цикла с предусловием.	1
Циклические конструкции. Оператор цикла с постусловием	1
Циклические конструкции. Оператор цикла с параметром.	1
Массивы	1
Методы обработки массивов	1

Продолжение таблицы 5 – Тематическое планирование Данилова Н.С.

Упорядочивание (сортировка) массивов	1
Поиск элемента в упорядоченном массиве	1
Символьные переменные. Тип данных CHAR	1
Решение задач по основам программирования.	1
Контрольная работа № 4 по теме «Алгоритмизация и основы программирования»	1
Всего часов	21

Босова Л.Л., Босова А.Ю. в пособии «Информатика. 10-11 классы. Базовый уровень. Примерная рабочая программа» [9] заостряют внимание, прежде всего, на рекомендации для проведения уроков по информатике. Данная работа содержит развернутые поурочные планирования, включающие образовательные результаты и учебные задачи, которые требуется решить.

Качественный и детальный теоретический материал представлен в учебниках Окулова С.М. «Основы программирования» и «Программирование в алгоритмах» [17, 18]. Учебник «Основы программирования» включает в себя сведения об основных конструкция программирования, процедурах, функциях, массивах и другой информации, необходимой для изучения программирования. В учебнике «Программирование в алгоритмах» рассмотрены основные алгоритмы, которые можно использовать при решении практических задач, в том числе для подготовки учащихся к участию в олимпиадах по программированию. Раздел «Алгоритмизация и программирование» подробно и фундаментально представлен в рассмотренных учебниках, но они не направлены на систематическую подготовку к ЕГЭ по информатике.

Рассмотрим сборники тестовых заданий ЕГЭ по информатике. В настоящее время представлено достаточное количество учебных практических материалов по данной теме, но особенно можно выделить сборники таких авторов как Ушаков Д.М., Зорина Е.М., Самылкина Н.Н. [32, 13, 26]. Издания включают в себя типовые тренировочные варианты для подготовки К ЕГЭ по информатике, которые имеют различные уровни сложности и будут полезны как для учителей при планировании и подготовке к уроку, так и для учеников при самостоятельной проверке знаний и готовности к сдаче экзамена.

Основой разрабатываемых методических материалов является Федеральный государственный образовательный стандарт основного общего образования [1]. Согласно стандарту, каждый ученик основной школы должен на базовом уровне освоить раздел «Алгоритмизация и программирование».

Умения и навыки, полученные при изучении этого раздела, позволят сформировать у выпускников следующие компетенции:

1. Сформированность склонности к логическому, алгоритмическому и математическому мышлению.

2. Использование знаний по разделу при решении задач различных предметных областей.

3. Сформированность представлений о влиянии современных информационных технологий на жизнедеятельность человека.

Федеральный государственный образовательный стандарт основного общего образования определяет следующие результаты обучения разделу «Алгоритмизация и программирование»:

Личностные – владение умениями и навыками анализа и критичной оценки получаемой информации; общения и сотрудничества с одноклассниками и учителями в процессе учебной деятельности.

Метапредметные – умение оперировать понятиями «объект», «алгоритм», «программирование» и т.д.; осуществлять постановку целей перед собой и находить пути ее достижения, соотносить планируемые результаты со своими действиями.

Предметные – развитие информационной культуры; формирование навыков алгоритмического мышления; формирование знаний о компьютере и навыков работы с ним; владение одним или несколькими языками программирования.

К предметным результатам базового курса информатики по разделу «Алгоритмизация и программирование» образовательный стандарт предъявляет следующие требования:

1. Овладение склонностью к алгоритмическому мышлению и пониманию отображения алгоритмов.

2. Овладение способностью читать и анализировать программы, написанные на каком-либо языке программирования, и разбираться в их смысле; знание основ программирования; умение проводить анализ алгоритмов.

3. Овладение приемами оформления программы на каком-либо языке программирования; применение стандартных компьютерных программ согласно предложенной специализации.

К предметным результатам профильного курса информатики по разделу «Алгоритмизация и программирование» образовательный стандарт предъявляет следующие требования:

1. Знание понятия алгоритма, умение составлять основные алгоритмы обработки численной и буквенной информации, поисковые и сортировочные алгоритмы.

2. Овладение одним из языков программирования на высшем уровне, представление о стандартных типах данных; умение пользоваться основными управляющими конструкциями.

3. Овладение навыком и опытом создания программ на одном из языков программирования, умение тестировать и отлаживать программы.

Также федеральный государственный образовательный стандарт определяет, что в итоге изучения раздела «Алгоритмизация и программирование», выпускники на базовом и углубленном уровне научатся или получают возможность научиться выполнению следующих операций:

Выпускники на базовом уровне научатся:

1. Прогнозировать итог алгоритма, выполненного по заданным исходным данным.

2. Различать алгоритмы по обработке последовательностей чисел; воссоздавать на их базе простые программы по анализу данных.

3. Уметь понимать простые программы, созданные на одном из языков программирования высокого уровня.

4. Прodelывать поэтапно простые алгоритмы по анализу численных и буквенных данных.

5. Воссоздавать при помощи алгоритма программы с целью решить типовые задачи на базовом уровне.

Выпускники на базовом уровне получают возможность научиться:

1. Применять умения и навыки по разработке программ на одном из языков программирования, в том числе использовать навык тестирования и отладки программ.

2. Пользоваться основными управляющими конструкциями с последовательным программированием и библиотеками прикладных программ.

Выпускники на углубленном уровне научатся:

1. Устанавливать уровень сложности стандартных алгоритмов.

2. Проводить анализ заданных алгоритмов.

3. Формировать, исследовать и создавать стандартные алгоритмы в формате программы, связанных с анализом простых функций, записей чисел в позиционной системе счисления и строк.

4. Реализовывать алгоритмы с обработкой массивов и рекурсивные алгоритмы.

5. Воссоздавать алгоритмы, с помощью которых можно будет решать прикладные задачи.

6. Уметь использовать для решения задачи списки, словари, деревья, очереди.

7. При составлении алгоритмов применять стандартные операции со структурами данных.

8. Использовать основные понятия, структуры данных и конструкции последовательного программирования, применять в программах данные всевозможных типов.

9. Использовать базовые и собственные подпрограммы для обработки символьных строк.

10. Обрабатывать данные, хранящиеся в виде массивов разной размерности, использовать различные типы цикла в зависимости от решаемой подзадачи.

11. Выполнять стандартные операции с двоичными и текстовыми файлами, выделять подзадачи, решение которых обязательно для полноценного решения предложенной задачи.

12. Выполнять решения подзадач в виде подпрограмм, объединять в единую программу разработанные подпрограммы».

13. Применять модульный принцип написания программ, обращаться к библиотекам стандартных подпрограмм.

Выпускники на углубленном уровне получают возможность научиться:

1. Находить несколько способов заданной задачи, имеющие разные уровни сложности.

2. Применять переборный алгоритм.

3. Применять универсальный алгоритм и показывать образцы алгоритмически нерешаемых проблем.

4. Использовать два или несколько языков программирования; видеть достоинства и недостатки каждого языка программирования.

5. Придумывать учебные проекты с созданием программ для средних по сложности задач.

В соответствии с федеральным базисным учебным планом для среднего общего образования Российской Федерации [2], на изучение курса информатики и ИКТ на базовом уровне отводится 70 часов на 2 года (35 часов в год, 3.8 часов в месяц и 1 час в неделю), а на профильном уровне – 280 часов на 2 года (140 часов в год, 15.5 часов в месяц, 4 часа в неделю).

Таким образом, анализ всех выше предложенных учебников, методических пособий и научной литературы позволяет сделать вывод, что

раздел «Алгоритмизация и программирование» является неотъемлемой и одной из важнейших частей обучения информатике. Но даже в учебниках профильного уровня представлено недостаточное количество практических заданий для-подготовки к ЕГЭ. В частности, это касается очень малого объема или даже отсутствия информации для изучения языка программирования Python и подготовки к ЕГЭ по информатике с его использованием.

1.2. Методические особенности подготовки к единому государственному экзамену по информатике (по разделу «Алгоритмизация и программирование»)

Единый государственный экзамен (ЕГЭ) является государственной итоговой аттестацией по учебным программам среднего общего образования.

Учащимся необходима углубленная подготовка к ЕГЭ по информатике, это обусловлено достаточно высокой сложностью заданий, предложенных для решения в экзаменационных тестах. Поэтому для успешной сдачи экзамена недостаточно только базового уровня.

Поэтому перед учителем информатики средней школы стоит задача дать учащимся базу знаний и подготовить к успешной сдаче единого государственного экзамена (ЕГЭ). ЕГЭ по информатике ориентирован на учащихся профильных и специализированных классов и проводится с использованием специально подобранных серий заданий, позволяющих установить уровень освоения выпускниками федерального компонента государственного образовательного стандарта среднего общего образования. Правильно решить все содержащиеся в варианте задания могут лишь хорошо подготовленные выпускники.

Наиболее сложными заданиями ЕГЭ по информатике являются задания содержательной линии «Алгоритмизация и программирование».

Предлагаемые на ЕГЭ серии заданий состоят из двух частей. Первая состоит из 23 заданий с кратким ответом, а вторая – из 4 заданий, требующих развернутые ответы. Задач, связанных с умением строить и анализировать

алгоритмы, писать программы для ЭВМ – восемь: 5 из первой части и 3 из второй. Задания из второй части принесут наибольшее количество баллов (3, 2 и 4 балла соответственно за 24, 25 и 27 задания). Самой трудной задачей считается 27, по статистике только 60 – 70% сдающих ЕГЭ по информатике решают ее.

Рассмотрим тематику предлагаемых заданий и необходимый для успешного их выполнения уровень подготовки выпускника.

При решении задания 8 ученики должны иметь устойчивые навыки анализа несложных алгоритмов. Проверяются знания основных конструкций языка программирования, способов объявления переменных, механизм работы оператора присваивания, операторов ввода и вывода данных, типы операторов цикла. Учащиеся должны уметь выполнять ручную прокрутку программы, выделять переменную цикла, от изменения которой зависит количество шагов цикла, определять количество шагов цикла и значение переменной, которая выводится на экран. Для этого необходимо четко понимать принцип работы циклических алгоритмов, поскольку в ответе на задание нужно указать конкретные значения переменных, изменяющихся в теле цикла. На выполнение задания 8 отводится 3 минуты, за правильный ответ ученик получает 1 балл.

В задании 11 на ЕГЭ проверяется знание учащимися приемов использования рекурсивных алгоритмов. Для подготовки к выполнению этого задания учащиеся должны усвоить, что под рекурсией понимается приём, позволяющий свести исходную задачу к одной или нескольким более простым задачам того же типа. Для организации рекурсивного алгоритма необходимо корректно задать условие останова рекурсии, определить и записать на языке программирования рекуррентную формулу. Также учитель должен акцентировать внимание обучаемых на том, что любую рекурсивную процедуру можно запрограммировать с помощью цикла, но рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным. На выполнение задания отводится 5 минут, за правильный ответ ученик получает 1 балл.

Задания 19 и 20 также ориентированы на проверку сформированности навыков анализа программ, содержащих ветвления и циклы, но относятся к повышенному уровню сложности. 19-е задание контролирует умения учащихся по обработке массивов данных. Для решения необходимо понимание работы цикла с управляющей переменной с заранее заданным числом повторений, циклов while или repeat, а также конструкций с вложенными циклами.

При подготовке к решению задания 21 нужно акцентировать внимание учащихся на том, что функция является вспомогательным алгоритмом, который возвращает некоторое значение – результат. Задание требует знания синтаксиса описания (в заголовке функции записывают имя функции, в скобках – список параметров, далее через двоеточие – тип возвращаемого значения) и вызова функций. Эта тема является одной из наиболее сложных в программировании, и ей необходимо уделить больше времени и внимания на занятиях по подготовке к ЕГЭ.

Задание 24 также относится к повышенному уровню сложности. При его выполнении учащийся должен найти разрешение проблемной ситуации: исправить ошибки в программе с условными операторами. Необходимо обратить внимание учащихся на то, что нужно исправить ошибки в имеющейся программе, а не писать свою, возможно использующую другой алгоритм решения. Это задание, как и предыдущие, проверяет сформированность у выпускников навыков анализа программ, их уровень алгоритмического и логического мышления. На выполнение задания отводится 30 минут, за правильный ответ ученик получает 3 балла.

Задания 25 и 27 относятся к высокому уровню сложности, это задания с «развернутым ответом» Учащимся предлагается написать программы средней сложности из 30-50 строк на языке программирования или алгоритм на естественном языке. Задания проверяют практические навыки и умения построения и реализации алгоритмов на выбранном языке программирования. При решении используются все основные конструкции языка программирования.

Можно отметить основные затруднения, возникающие при решении заданий ЕГЭ по программированию:

- Анализ алгоритмов и программ – достаточно сложный и затратный по времени процесс, требующий значительных усилий и устойчивой мотивации от начинающих программистов;
- Самой сложной для понимания и практического использования в школьном курсе программирования является тема «Подпрограммы», требующая модульного подхода к написанию программ.
- Учащиеся со слабой математической подготовкой с большим трудом усваивают учебный материал по теме «Рекурсивные алгоритмы»;
- Самостоятельно разрабатывать программы в 30-50 строк при решении заданий высокого уровня сложности могут лишь хорошо подготовленные к ЕГЭ выпускники.

Дискуссионной остается проблема выбора языка программирования при изучении этого раздела информатики и подготовке к единому госэкзамену.

Данные задания предполагают выбор языка программирования, с помощью которого сдающий ЕГЭ может решать задачу: Бейсик, Паскаль, Си и Python.

Язык программирования Python разрешено использовать на ЕГЭ с 2018 года. Он предлагает возможность быстрого создания программных систем, помогает интегрировать программное обеспечение, решающее производственные задачи. У данного языка программирования имеется богатая стандартная библиотека и огромное число модулей расширения.

Ясный синтаксис языка программирования Python позволяет быстрое и простое его изучение. Созданные при помощи данного языка программы имеют структурированную форму, логика работы легко прослеживается. При помощи языка программирования Python могут быть изучены следующие темы: операторы присваивания, арифметические действия, операторы циклов, функции и процедуры, списки, строки, массивы, матрицы и работа с файлами.

Также этот язык программирования является свободно распространяемым программным обеспечением для различных платформ.

Таким образом, можно выделить преимущества языка программирования Python:

1. Простота освоения.
2. Понятный и лаконичный синтаксис.
3. Кроссплатформенность.
4. Богатая стандартная библиотека.
5. Наличие большого числа подключаемых к программе модулей, обеспечивающих различные дополнительные возможности.
6. Поддержка объектно-ориентированного программирования.
7. Интеграция с C/C++.

В школьной практике для изучения программирования в основном используется язык программирования Паскаль, так как по нему существует большое количество учебных пособий, учебников, методологических материалов, да и опыт учителей по обучению на данном языке программирования гораздо шире, чем на Python.

Так как учебные материалы по изучению языка программирования Python немногочисленны, школьник может выучить данный язык программирования и использовать его при решении задач по программированию на ЕГЭ или только по инициативе учителя, который, проанализировав ряд учебников, подготовит нужные конспекты уроков, или по собственному желанию, изучив соответствующие материалы.

В настоящее время не существует официально утвержденных учебников по подготовке к ЕГЭ по алгоритмизации и программированию на языке программирования Python. Но можно обратить внимание на учебники и сайт Полякова К.Ю. [22, 5].

На сегодняшний день, существует множество элективных и факультативных курсов по подготовке к ЕГЭ по информатике, некоторые из них даже предполагают дистанционное обучение. Но, в отличие от случая с

учебниками и учебными пособиями, не имеется ни одного электронного ресурса, направленного на подготовку к ЕГЭ с использованием языка программирования Python.

ЭУМК целесообразно задействовать на уроках информатики при изучении раздела «Алгоритмизация и программирование». Поэтому подготовку к ЕГЭ по этому разделу следует сразу включить при обучении учащихся соответствующим темам. Следовательно, необходимо выполнить планирование занятий так, чтобы разборы заданий из бланка ЕГЭ по информатике соответствовали по тематике материалам, которые освещаются на уроках.

За основу можно взять поурочное планирование на раздел «Алгоритмизация и программирование», составленное по учебнику Полякова К.Ю. «Информатика. 10 класс. Углубленный уровень» [22]. Примерное тематическое планирование занятий с встроенными пунктами по подготовке к решению заданий из бланка ЕГЭ по информатике представлено в таблице 6.

Таблица 6 – Тематическое планирование

Номер урока	Тема урока	Параграф учебника	Разбор заданий ЕГЭ	Часы
Глава 8. Алгоритмизация и программирование				
1	Алгоритм и его свойства	§ 54	-	1
2	Простейшие программы	§ 55	-	1
3	Вычисления	§ 56	-	1
4	Ветвления	§ 57	-	1
5	Циклические алгоритмы	§ 58	Задание 8. Анализ программ с циклами	2
			Задание 20. Анализ программы с циклами и условными операторами	
6	Процедуры	§ 59	-	2
7	Функции	§ 60	Задание 21. Анализ программы с циклами и подпрограммами	2
8	Рекурсия	§ 61	Задание 11. Рекурсивные алгоритмы	2
9	Массивы	§ 62	Задание 19. Обработка массивов и матриц	2
10	Алгоритмы обработки массивов	§ 63	Задание 25. Алгоритмы обработки массивов	2
11	Сортировка	§ 64	-	2
12	Двоичный поиск	§ 65	-	1

Продолжение таблицы 6 – Тематическое планирование

13	Символьные строки	§ 66	Задание 27. Обработка массивов, символьных строк и последовательностей	1
14	Матрицы	§ 67	Задание 24. Поиск и исправление ошибок в программе	2
15	Работа с файлами	§ 68	-	2
Всего часов:				24

1.3. Электронный учебно-методический комплекс по обучению программированию как компонент программно-методического обеспечения подготовки к единому государственному экзамену по информатике

Электронным учебно-методическим комплексом (ЭУМК) называется совокупность программных и технических средств, теоретических, практических и контролирующих материалов, учебников и учебных материалов, представленных в электронном виде и включающие в себя принципы дистанционности, интерактивности и информационной открытости.

Другими словами, **ЭУМК** – это учебно-методические материалы, разработанные и воспроизведённые с помощью компьютерных технологий.

Его главной задачей является модернизация образовательного процесса с выходом на современный уровень использования информационно-коммуникационных технологий, который будет интересен сегодняшним ученикам.

Введение в процесс обучения электронного учебно-методического комплекса позволяет создать современные педагогические инструменты и открывает новейшие возможности. Следует учесть, что данное внедрение добавляет изменения не только в саму структуру обучения, но и в функции, цели и задачи педагога, расширяя его возможности. ЭУМК подразумевает значительный объем самостоятельной учебной работы учащихся. Из этого следует, что требуется внедрение методик и подходов, используемых в данной форме обучения.

К преимуществу электронных учебно-методических комплексов относится систематизированная совокупность материалов, к которым относятся примерные планы лекционных и практических уроков, программы проверочных, контрольных, а также методические рекомендации по обучению дисциплины.

Выделим конкретные преимущества применения электронных учебно-методических комплексов в учебном процессе:

1. Представление информации в виде мультимедиа дает более эффективное усвоение материала. Ученики лучше воспринимают мультимедийную информацию (например, представленную в виде презентации, видеороликов и др.).

2. Интерактивность. Современные технологии позволяют представить учебный материал не только как видеокурс, но и как интерактивную презентацию или игровое обучающее приложение.

3. Возможность распространения материала по сети. Существующие в достаточном количестве образовательные платформы дают преподавателям возможность выкладывать свои учебные пособия и курсы в сеть.

4. Быстрый и удобный поиск нужной информации.

5. Возможность дистанционного обучения. В настоящее время для дистанционного обучения преподаватель может как выложить свои материалы в сеть, так и организовать видеосвязь со своими учениками через специальные приложения.

6. Возможность выбрать удобный темп изучения информации. Возможности каждого ученика индивидуальны, поэтому ЭУМК позволяет учащемуся изучать материал в комфортном для него темпе.

7. Адаптация по индивидуальным запросам ученика.

8. Простая организация системы контроля знаний.

Кроме этого, можно рассмотреть ряд существенных недостатков ЭУМК:

1. Проблема перехода от традиционного обучения к дистанционному.

2. Недостаток специального оборудования и программ для использования ЭУМК.

3. Несоблюдения временных норм по использованию компьютера может привести к быстрой утомляемости и даже развитию некоторых заболеваний.

4. Проблема недостаточной квалификации и подготовки преподавателей, которые способны эффективно проводить уроки с помощью ЭУМК.

Из всего вышесказанного можно сделать вывод, что учитель информатики должен обладать профессиональными компетенциями по разработке авторских электронных учебно-методических комплексов, в том числе, по подготовке к ЕГЭ.

ЭУМК для подготовки к ЕГЭ по информатике реализован в форме видеокурса, который опубликован на облачной образовательной web-платформе дистанционного обучения. У учащихся появится возможность изучать нужный материал не только на уроках, но и дома. Videocourse построен таким образом, что ученики имеют возможность многократно просматривать объяснение теоретического материала и демонстрационных примеров, а также практиковаться в решении задач.

ВЫВОДЫ ПО ГЛАВЕ 1

Нами были рассмотрены теоретические предпосылки разработки программно-методического обеспечения для подготовки выпускников средних школ к единому государственному экзамену по разделу «Алгоритмизация и программирование», обоснованы его структура и содержание, определены методические особенности.

Проведенная работа показывает, что существующие в настоящее время учебно-методические материалы по изучению языка программирования Python не ориентированы на обучение школьников, и тем более на подготовку их к сдаче ЕГЭ по информатике. Решением выявленной проблемы является разработка ЭУМК по этой тематике.

Глава 2. Проектирование и реализация электронного учебно-методического комплекса по разделу «Алгоритмизация и программирование»

2.1. Дидактические, программно-технологические и технические характеристики электронного учебно-методического комплекса

Для проектирования и реализации ЭУМК для подготовки к ЕГЭ по информатике необходимо проанализировать целевую аудиторию, выполнить постановку целей и задач электронного курса, определить его структуру и содержание, выбрать формы и средства представления учебных материалов. Для реализации ЭУМК была выбрана наиболее наглядная и доступная форма – видеокурс, представляющий собой серию видеоуроков, объединенных одной темой и имеющих логическую последовательность.

Главные вопросы, решаемые при данном анализе:

- Для какой целевой аудитории предназначен данный курс?
- Что по окончании курса учащиеся будут знать, уметь и какими навыками будут обладать?

При проектировании ЭУМК важно определить не только то, что обучающиеся будут знать и уметь, но и то, как и где они смогут применить свои знания.

Цель видеокурса для подготовки к ЕГЭ по разделу «Программирование» – это формирование у учащихся 10-11 классов навыков самоанализа и систематизации знаний, полученных в процессе обучения; развитие умений анализировать программы на языке программирования Python и решать типовые задания из тестов ЕГЭ.

Задачи видеокурса:

1. Освоить приемы работы с программными проектами на языке программирования Python.

2. Обучить учеников основам конструирования программ на языке программирования Python: объявление переменных, операторы присваивания, операторы ввода и вывода данных, типы операторов цикла.

3. Отработать навыки анализа несложных алгоритмов.
4. Научить учащихся выполнять тестирование программы, выделять переменную цикла, от изменения которой зависит количество шагов программы, определять количество шагов цикла и значения выходных переменных.
5. Сформировать у обучаемых навыки использования приемов обработки рекурсивных алгоритмов: задание условия выхода из рекурсии, определение и записи на языке программирования рекуррентных формул.
6. Обучить учеников навыкам анализа программ, содержащих ветвления и циклы.
7. Научить учащихся обрабатывать массивы данных и работать с циклами `while` и `repeat`.
8. Обучить учащихся приемам автоматизированной обработки данных, представленных в виде массивов и матриц.
9. Сформировать навыки использования циклических конструкций `while` и `repeat` при обработке массивов данных.
10. Обучить учеников синтаксису описания и вызова функций и процедур.
11. Обучить учеников находить разрешение проблемной ситуации: исправить ошибки в программе с условными операторами.
12. Развить у учеников навыки анализа программ, позволяющие повысить их уровень алгоритмического и логического мышления.
13. Развить у учеников практические навыки и умения построения и реализации алгоритмов на языке программирования Python.
14. Развить у учащихся навыки построения алгоритмов, при решении которых используются все основные конструкции языка программирования.
15. Разобрать примеры типовых заданий из теста ЕГЭ.

По окончании видеокурса, ученик должен:

Знать:

Основы алгоритмизации и программирования с использованием языка программирования Python; основные алгоритмические конструкции (линейную, условную и циклическую); основные операторы и приемы программирования алгоритмов на языке программирования Python;

Уметь:

использовать компьютерные устройства для решения практических задач; составлять и записывать алгоритм для конкретного исполнителя;

Владеть:

методами и навыками разработки, отладки и тестирования программ.

Видеоуроки представляют собой одну из разновидностей электронных образовательных ресурсов.

По мнению Осина А.В. [19], ЭОР нового поколения – это открытые образовательные модульные мультимедиа системы.

Структура электронного образовательного ресурса состоит из трех модулей (Рисунок 1. Модули структуры ЭОР):

1. Информация (И) – обучающая модель, а именно теория.
2. Практикум (П) – практика, к ним относятся практические и лабораторные работы.
3. Контроль (К) – тестирование и проверка знаний.

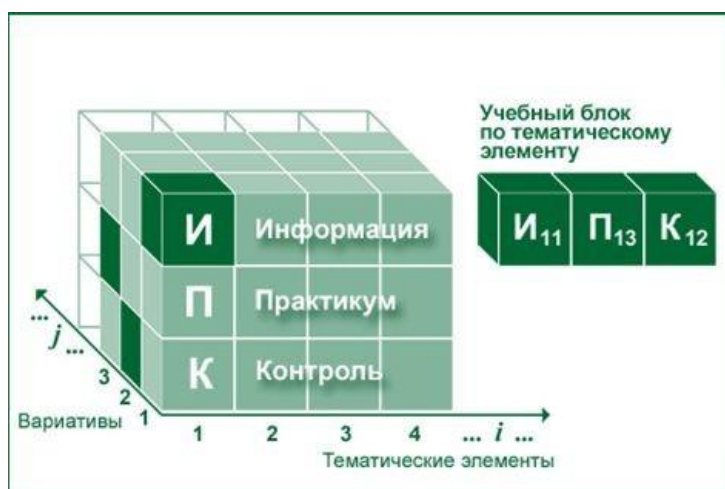


Рисунок 1 – Модули структуры ЭОР

А.В.Осин [19] пишет о том, что каждый ЭОР имеет свой вариатив.

Вариатив – это электронные учебные модули (ЭУМ), имеющие одинаковый тип (И, П или К) и относящиеся к одной и той же тематике предложенного предмета.

А.В.Осин выделяет отличия вариативов:

- глубина изложенных материалов;
- методика;
- характер учебных работ;
- технология представления учебного материала;
- наличие специальных возможностей (слабослышащие или слабовидящие);
- способы достижения учебной цели.

Таким образом, ЭУМК для подготовки к ЕГЭ по информатике должен состоять из трех тесно взаимосвязанных между собой частей:

1. **Теория** – ряд видеороликов, в которых производится разбор основной теории, использованной в дальнейшем для решения заданий.

2. **Практика** – ряд видеороликов, посвященных разбору примеров предложенного задания.

3. **Контроль** – домашнее задание, тесты, ряд дополнительных онлайн-упражнений и заданий для закрепления материала.

Таким образом, структура видеокурса «Подготовка к ЕГЭ по информатике с использованием языка программирования Python» представлена в следующем виде (рисунок 2):

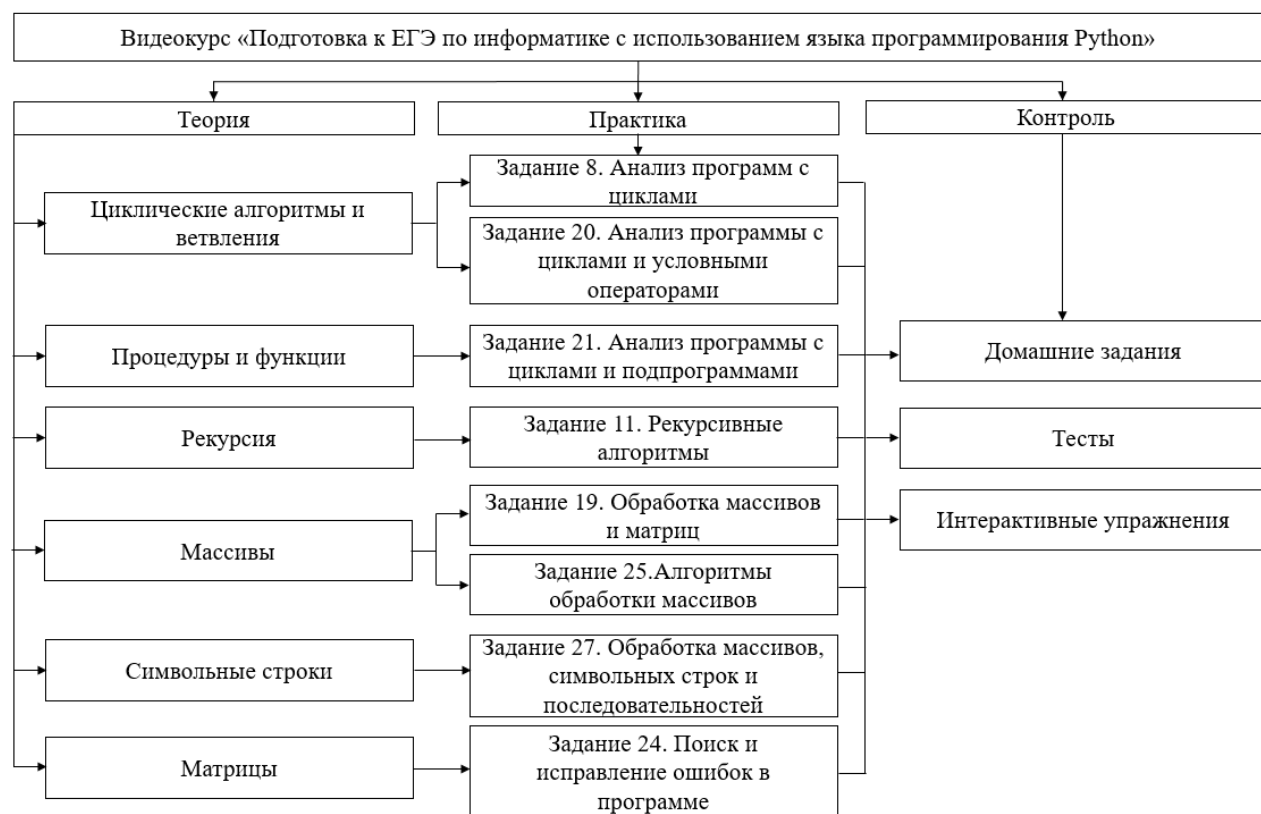


Рисунок 2 – Структура видеокурса

Перед подбором теоретического материала, практических заданий для закрепления изученной информации и материала для контроля знаний необходимо было определить общие сведения о видеокурсе (таблица 7).

Таблица 7 – Общие сведения видеокурса

Название	Подготовка к ЕГЭ по информатике с использованием языка программирования Python
О курсе	Данный курс будет полезен для учащихся старших классов, которые выбрали для сдачи ЕГЭ предмет «Информатика». Мы поэтапно разберем с вами все задания, связанные с программированием. Подробно изучим теорию, которую затем закрепим на практике.
Формат	Курс состоит из 9 уроков. Один урок включает в себя: 1. Видео с теоретическим материалом. 2. Видео с практическим материалом. 3. Домашнее задание, тест, интерактивное упражнение. Нагрузка: 1 занятие в неделю (2 видеоролика) Продолжительность одного видеоролика: 5 - 10 минут
Направление подготовки	Подготовка к ЕГЭ по информатике

Продолжение таблицы 7 – Общие сведения видеокурса

<p>Требования</p>	<p>Учащиеся должны знать:</p> <ul style="list-style-type: none"> - что такое алгоритмы; какая роль алгоритмов в системах управления; - основные свойства алгоритма; - способы записи алгоритмов: словесный, операторный, графический (блок-схемы), на алгоритмических языках программирования;- схему знакомства с исполнителем; - схемы взаимодействия с исполнителями; - основные управляющие команды организации действий в алгоритмах: следование, ветвление, циклы; структуры алгоритмов; - назначение вспомогательных алгоритмов; технологии построения сложных алгоритмов: «сверху вниз» или метод последовательной детализации алгоритма, «снизу вверх» или использование готовых вспомогательных алгоритмов; - способы организации (типы) данных; - назначение систем программирования; - этапы исполнения компьютером программы, написанной на языке программирования высокого уровня; - содержание этапов разработки программы на языке высокого уровня: алгоритмизация – кодирование – отладка – тестирование; - организацию действий и организацию данных в написании алгоритмов на одном из универсальных языков программирования. <p>Учащиеся должны уметь:</p> <ul style="list-style-type: none"> - приводить примеры алгоритмов, определять свойства выбранного алгоритма; - определять исполнителя; приводить примеры задач для выбранного исполнителя; писать алгоритмы решения задач для определенного исполнителя; - пользоваться частично формализованным языком для описания алгоритмов, языком блок-схем; - анализировать структуру алгоритма; - определять по выбранному методу решения задачи, какие алгоритмические конструкции могут войти в алгоритм; - выделять подзадачи; определять и использовать вспомогательные алгоритмы; пользоваться различными методами проектирования алгоритмов: «сверху вниз», «снизу вверх» и др.; - писать алгоритмы (линейные, разветвляющиеся, циклические) на алгоритмических языках высокого уровня.
--------------------------	---

Продолжение таблицы 7 – Общие сведения видеокурса

Перечень видеолекций	1. Введение. 2. Задание 8. Анализ программ с циклами 3. Задание 11. Рекурсивные алгоритмы 4. Задание 19. Обработка массивов и матриц 5. Задание 20. Анализ программы с циклами и условными операторами 6. Задание 21. Анализ программы с циклами и подпрограммами 7. Задание 24. Поиск и исправление ошибок в программе 8. Задание 25. Алгоритмы обработки массивов 9. Задание 27. Обработка массивов, символьных строк и последовательностей
Результаты	Формирование у учащихся 10-11 классов навыков самоанализа и систематизации знаний, полученных в процессе обучения; развитие умения анализировать программы на языке программирования Python и решать типовые задания из тестов ЕГЭ
Информация о преподавателе	Коновалова Анжелика Денисовна, студентка Поволжского православного института Электронная почта: konovalova.anzhel@yandex.ru Телефон: 89879344751

Такое использование видеокурса является рекомендованным, но его можно изучать и отдельно, при самостоятельном обучении. Примерное тематическое планирование видеокурса «Подготовка к ЕГЭ по информатике с использованием языка программирования Python» представлено в таблице 8.

Таблица 8 – Содержание видеокурса

Номер урока	Тема видеурока	Содержание			
		Описание	Компоненты		
			Теория	Практика	Форма контроля
1	Введение	Знакомство учеников с преподавателем и с содержанием курса, ознакомление с функциональными возможностями системы программирования Visual Studio	Ознакомительное видео	-	Анкета; тест

Продолжение таблицы 8 – Содержание видеокурса

2	Задание 8. Анализ программ с циклами	Формирование навыков составления алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
3	Задание 11. Рекурсивные алгоритмы	Формирование навыков составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
4	Задание 19. Обработка массивов и матриц	Формирование навыков составления алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
5	Задание 20. Анализ программы с циклами и условными операторами	Формирование навыков составления разветвляющихся и циклических алгоритмов на языке программирования Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
6	Задание 21. Анализ программы с циклами и подпрограммами	Формирование навыков анализа модульных программ, содержащих процедуры и функции и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение

Продолжение таблицы 8 – Содержание видеокурса

7	Задание 24. Поиск и исправление ошибок в программе	Формирование навыков анализа сложных программ на языке Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
8	Задание 25. Алгоритмы обработки массивов	Формирование навыков обучаемых по составлению алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение
9	Задание 27. Обработка массивов, символьных строк и последовательностей	Формирование навыков обучаемых по обработке структурированных строковых и символьных данных и закрепление полученных знаний на практике	Видео с теоретическим материалом	Видео с практическим материалом	Домашнее задание; интерактивное упражнение; тест

Для того, чтобы определить программно-технологические и технические характеристики ЭУМК, необходимо выбрать формат записи видеокурса, зависящий не только от тематики и целей, но и от необходимых ресурсов и навыков.

Рассмотрим основные виды видеокурсов:

- видеозапись с преподавателем, учениками, специальными актерами, разыгрывающие сценки;
- графический материал – слайды презентации, анимация, анимированные картинки и др.;
- материал с записью голоса – голосовые комментарии, сделанные преподавателем за кадром;
- интерактивные взаимодействия – действия, выполненные во время просмотра курса.

Наш итоговый видеокурс состоит из комбинации всех перечисленных форматов.

Требования к разработке и использованию электронного учебно-методического комплекса можно классифицировать следующим образом:

1. Педагогические.
2. Технические.
3. Эргономические.
4. Эстетические.

К педагогическим требованиям относятся:

1. Наглядность. Важно представить материал так, чтобы он легко воспринимался, для этого человеком должны быть задействованы важнейшие органы чувств: слух и зрение. Подача информации в виде мультимедиа (например, видеоролики, презентации, изображения, схемы) приводит к более эффективному обучению.

2. Доступность. Изложение материалов необходимо выполнять по специальной методике, чтобы сделать обучение доступным для восприятия. Например, представлять информацию от простейшего к сложному, от теории к практике.

3. Ориентированность на практику. Важно, чтобы в ЭУМК присутствовали не только теоретические материалы по предложенной теме, но и достаточное количество практических заданий для закрепления информации (например, тесты, лабораторные работы, онлайн-упражнения и задания).

4. Интерактивность. Обучающийся должен иметь возможность взаимодействовать с контентом ЭУМК (например, просматривать материал, отвечать на вопросы, выполнять домашнее задание с получением оценки или балла, иметь обратную связь с преподавателем).

5. Последовательность изложения. Представление материала должно быть логически взаимосвязанным (например, от легкого к сложному, от теории к практике).

Технические требования к разработке ЭУМК:

1. Простая навигация. У обучающихся не должна вызывать проблем навигация по сайту или платформе ЭУМК. Любой материал доступен и легко находится.

2. Правильное использование мультимедиа-компонентов. Средства мультимедиа должны быть использованы адекватно, не раздражая излишками, а настраивая на эффективное обучение.

3. Скорость загрузки. Изображения, текст, презентации и видео должны иметь минимальный объем памяти.

4. Высокая интерактивность. В этом случае максимальное взаимодействие учащихся с предложенным контентом принесет более эффективные результаты.

5. Качественная реализация. Материал для обучения должен быть корректным, в хорошем качестве. Также должна быть налажена обратная связь.

Эргономические требования:

1. Учет возрастных и индивидуальных особенностей обучающихся. У учеников различные типы мышления, интеллектуальную и эмоциональную работоспособность, поэтому материал должен быть адаптирован для всех.

2. Обеспечение мотивации обучения. У учащихся должны быть положительные стимулы для изучения материала.

3. Установка требований к представлению информации. Материалы должны быть разборчивыми и четкими.

Эстетические требования:

1. Оформление ЭУМК, соответствующее назначению.

2. Подходящий цветовой колорит.

3. Выразительные графические и изобразительные элементы.

На этапе реализации программного проекта с использованием веб-серверов проводилось многократное тестирование и коррекция электронных учебных материалов для оценки степени их использования в практике обучения.

Перенос готового курса на образовательную платформу позволяет ученикам получить онлайн и офлайн доступ к нужной информации.

Для выбора web-платформы публикации разработанного видеокурса был проведен анализ их функциональных возможностей, достоинств и недостатков:

1. Google Classroom. Это удобный сервис от Google, который был организован специально для обучения. Платформа направлена на создание своего класса или курса, организации записи учеников на курсы, она позволяет не только делиться с учащимися теоретическими и практическими учебными материалами, но создать систему автоматического оценивания результатов обучения, и даже организовать обратную связь с учениками для общения с ними.

Данная платформа позволяет создать несколько курсов, позволяя обучать учеников в различных направлениях. Например, составить курс для подготовки к ЕГЭ и виртуальный класс для самостоятельной работы как показано на рисунке 3.

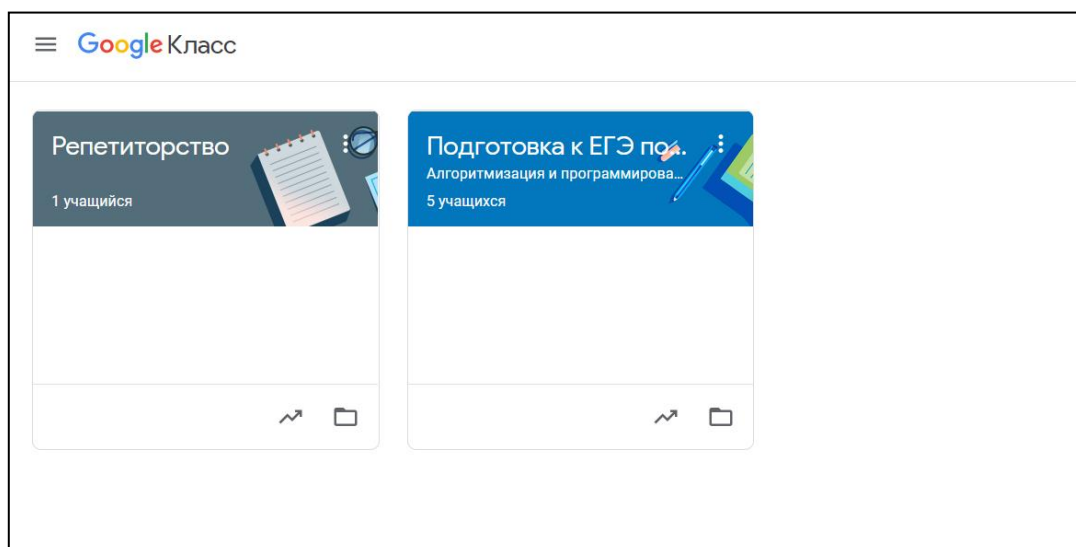


Рисунок 3 – Пример курсов на платформе Google Classroom

Платформа имеет простую и понятную структуру. Для создания курса доступны четыре основные вкладки: Лента, Задания, Пользователи, Оценки. Данные вкладки требуют более детального рассмотрения:

Лента. Здесь отображаются актуальные объявления по курсу, теоретические и практические материалы, также доступна функция комментариев как от преподавателя, так и от учеников.

Задания. Позволяет удобно распределить все учебные материалы по темам, чтобы ученики быстро и легко могли найти нужную им информацию.

Пользователи. Предоставляет список учеников, которые присоединились к курсу, также в этой вкладке можно добавить новых.

Оценки. Данная вкладка нужна для удобного просмотра выполненных учениками заданий и выставления за них оценок.

Также можно выделить преимущества и недостатки данной образовательной платформы.

Достоинства:

- удобный интерфейс;
- простая система создания курсов;
- использование Google диска для хранения материалов;
- обратная связь между преподавателем и учениками.

Недостатки:

- малое количество инструментов для создания курсов;
- ограниченное количество учеников;
- не предусмотрена система прямой трансляции.

2. Stepik. Данная платформа имеет обширные и простые возможности создания курсов, также она позволяет использовать созданные материалы на сторонних сайтах. Для своего курса можно добавить сертификат о его прохождении, что делает данную платформу более привлекательной. Большинство курсов на данном сайте выполнены в виде видеороликов, а направлением большого количества из них является информатика и программирование.

Достоинства:

- возможность добавления сертификата за курс;
- удобный интерфейс;

- неограниченное количество учеников;

Недостатки:

- отсутствие обратной связи между преподавателем и учениками;
- ограниченное количество инструментов для создания курсов.

3. Moodle (от англ. *Modular Object-Oriented Dynamic Learning Environment*). Аббревиатура расшифровывается как модульная объектно-ориентированная динамическая обучающая среда.

Moodle позволяет организовать курсы, учитывая логичную последовательность контента. Любой ЭОР можно разделить по темам, которые в последствии преподаватель наполнит готовыми теоретическими и практическими материалами. Важной особенностью данной платформы является встроенный редактор, позволяющий проектировать тесты. Автор курса загружает базу с большим количеством вопросов, а система тестирования составляет тест из нескольких заданий, выбирая их случайным способом.

Достоинства:

- удобный интерфейс;
- поддержка формата SCORM;
- неограниченное количество учеников;
- обратная связь между преподавателем и учениками;
- высокая производительность;
- простая установка и настройка системы;
- простая система создания курсов.

Недостатки:

- не предусмотрена система прямой трансляции;
- ограниченное количество инструментов для создания курсов.

4. Udemy. В своем названии компания обыгрывает выражение «U Academy», что переводится как «твоя академия». Сервис имеет простой интерфейс на русском языке. Курсы, которые создает преподаватель, могут быть как открытыми (доступными для всех), так и приватными (только по приглашению учителя).

Платформа имеет удобные инструменты для создания ЭОР, но, как и в случае со Sterik, большинство курсов на данном сервисе – это видео.

Достоинства:

- удобный интерфейс;
- возможность добавления сертификата за курс;
- большое количество инструментов для создания курсов.
- обратная связь между преподавателем и учениками;

Недостатки:

- не предусмотрена система прямой трансляции;
- сложная система создания курсов;
- ограниченное количество учеников.

5. LearningApps. Это сервис для создания интерактивных упражнений и заданий. Платформа имеет широкие возможности, удобную навигацию и простоту ее использования. Для того, чтобы облегчить работу пользователя на сайте предусмотрено использование готовых шаблонов упражнений и игр, которые всего лишь надо переделать под свою тему. Шаблоны можно разбить на группы по функционалу:

1. Selection – задания с выбором правильных ответов.
2. Assignment – упражнения с установлением соответствий.
3. Sequence - определение правильной последовательности.
4. Онлайн-игры – задания, в которых ученики могут соревноваться с компьютером или другими учащимися.

Примеры шаблонов можно увидеть на рисунке 4.

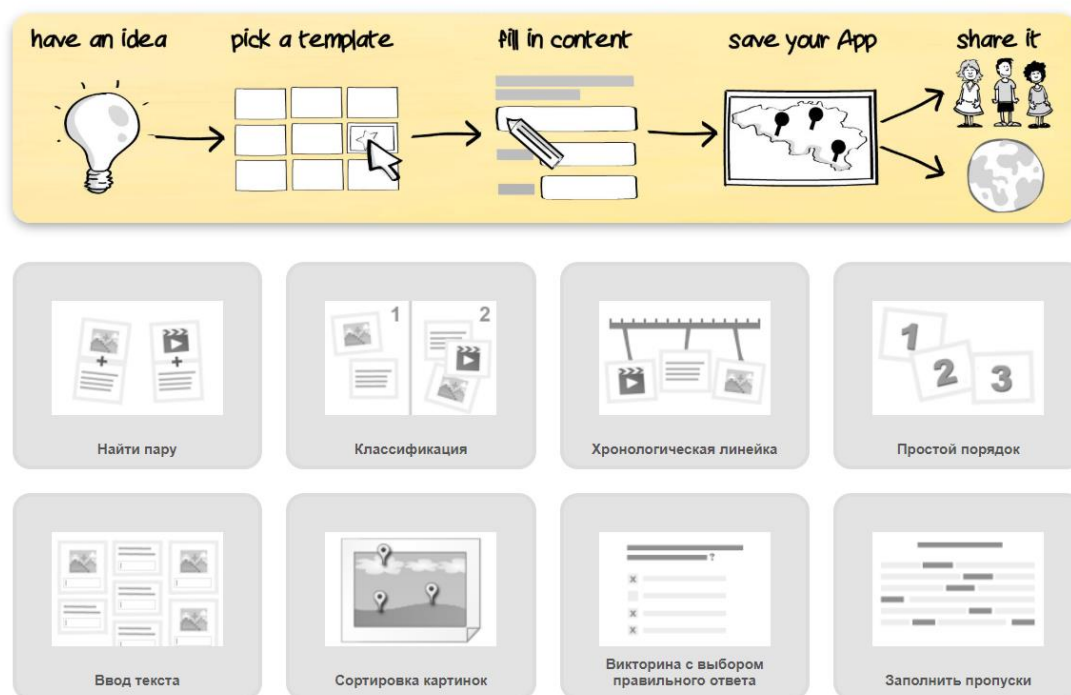


Рисунок 4 – Примеры шаблонов заданий

Кроме этого данная платформа предусматривает обратную связь с учениками, которые позволят учащимся уточнить у преподавателя непонятные моменты.

Достоинства:

- быстрое создание заданий;
- возможность использовать готовые шаблоны;
- возможность обмена заданиями.

Недостатки:

- Некоторые шаблоны не поддерживают кириллицу.

Таким образом, результаты анализа существующих образовательных платформ можно представить в виде следующей таблицы (Таблица 9).

Таблица 9 – Анализ образовательных платформ

Название платформы	Google Classroom	Stepik	Moodle	Udemy	LearningApps
Бесплатное использование	+	+/-	+/-	+	+
Русский интерфейс	+	+	+	+	+
Простота использования	+	+	+/-	+	+

Продолжение таблицы 9 – Анализ образовательных платформ

Обратная связь	+	-	+	+	+
Создание коммерческих проектов	-	+	-	+	-
Добавление сертификата за прохождение курса	-	+	-	+	-

Анализ функциональных возможностей электронных образовательных web-платформ привел к выводу, что для реализации готового видеокурса эффективнее использовать веб-сервер системы управления обучением Moodle, а для дополнительных интерактивных упражнений - облачный сервис LearningApps.

2.2. Проектирование и реализация теоретико-познавательного модуля электронного учебно-методического комплекса

Теоретико-познавательный модуль электронного учебно-методического комплекса состоит из 9 видеороликов (таблица 10). Каждый видеоролик включает в себя краткий теоретический материал (Приложение А). Длительность одного видеоролика – 5-10 минут.

Таблица 10 – План теоретико-познавательного модуля ЭУМК

Номер урока	Тема видеурока	Описание
1	Введение	Теоретический видеоролик знакомит учеников с преподавателем и с содержанием курса, а также с функциональными возможностями системы программирования Visual Studio
2	Задание 8. Анализ программ с циклами	Теоретический видеоролик знакомит учащихся с такими понятиями как объявление переменных, оператор присваивания, оператор вывода, цикл while, формулы арифметической прогрессии. Также происходит формирование навыков составления алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python

Продолжение таблицы 10 – План теоретико-познавательного модуля ЭУМК

3	Задание 11. Рекурсивные алгоритмы	Теоретический видеоурок знакомит учащихся с такими понятиями как создание функции, условный оператор if, ключевое слово return, рекурсия. Также происходит формирование навыков составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python
4	Задание 19. Обработка массивов и матриц	Теоретический видеоурок знакомит учащихся с такими понятиями как цикл for, массив, Матрица. Также происходит формирование навыков составления алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python
5	Задание 20. Анализ программы с циклами и условными операторами	Теоретический видеоурок знакомит учащихся с такими понятиями как операция div, операция mod, перевод чисел из одной системы счисления в другую. Также происходит формирование навыков составления разветвляющихся и циклических алгоритмов на языке программирования Python
6	Задание 21. Анализ программы с циклами и подпрограммами	Теоретический видеоурок знакомит учащихся с такими понятиями как процедура, функция. Также происходит формирование навыков анализа модульных программ, содержащих процедуры и функции
7	Задание 24. Поиск и исправление ошибок в программе	Теоретический видеоурок формирует у учащихся навыки анализа сложных программ на языке Python
8	Задание 25. Алгоритмы обработки массивов	Теоретический видеоурок формирует у учащихся навыки по составлению алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python
9	Задание 27. Обработка массивов, символьных строк и последовательностей	Теоретический видеоурок знакомит учащихся с таким понятием как работа со строками. Также происходит формирование навыков обучаемых по обработке структурированных строковых и символьных данных

Урок 1. Введение.

Цель видеоурока – познакомить учеников с преподавателем и с содержанием курса, познакомиться с функциональными возможностями системы программирования Visual Studio.

Задачи:

1. Установить систему программирования Visual Studio на персональный компьютер в соответствии с техническими требованиями.
2. Освоить приемы работы с программными проектами на языке программирования Python.

Изучив данный видеоурок, ученик должен:

Знать:

основы работы в программе Visual Studio, основные понятия языка программирования Python;

Уметь:

работать в системе программирования Visual Studio;

Владеть:

методами и навыками разработки, отладки и тестирования простейших программ.

Учебные вопросы:

1. Система программирования Visual Studio.
2. Вывод данных.
3. Функция Print ().

Урок 2. Задание 8. Анализ программ с циклами

Цель видеоурока – сформировать навыки составления алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python.

Задачи:

1. Обучить учеников объявлять переменные.
2. Обучить учащихся работать с операторами присваивания, ввода и вывода данных.
3. Познакомить учеников с типами операторов цикла.

Изучив данный видеоурок, ученик должен:

Знать:

основные понятия языка программирования Python, типы данных, операторы, структуру программы, базовые конструкции структурного программирования;

Уметь:

строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием циклов.

Учебные вопросы:

1. Объявление переменных.
2. Оператор присваивания.
3. Оператор вывода.
4. Цикл while.
5. Формулы арифметической прогрессии.
6. Составление алгоритмов и программ.

Урок 3. Задание 11. Рекурсивные алгоритмы.

Цель видеоурока – сформировать навыки составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python.

Задачи:

1. Обучить учеников задавать выход из рекурсии.
2. Сформировать у обучаемых навык использования определения и записи на языке программирования рекуррентных формул.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции, оператора if, ключевого слова return, определение и конструкцию рекурсии;

Уметь:

использовать конструкции функции и рекурсии, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования простейших программ с использованием функции и рекурсии.

Учебные вопросы:

1. Функция.
2. Оператор if.
3. Ключевое слово return.
4. Рекурсия.

Урок 4. Задание 19. Обработка массивов и матриц.

Цель видеоурока – сформировать навыки составления алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python.

Задачи:

1. Обучить учеников навыкам анализа программ, содержащих ветвления и циклы.
2. Научить учащихся обрабатывать массивы данных и работать с циклом repeat.

Изучив данный видеоурок, ученик должен:

Знать:

понятие цикла for, определения и конструкции массива и матрицы;

Уметь:

использовать конструкции массивов и матриц, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием массивов и матриц.

Учебные вопросы:

1. Цикл for.
2. Цикл while.
3. Цикл repeat.
4. Массив.
5. Матрица.

Урок 5. Задание 20. Анализ программы с циклами и условными операторами.

Цель видеурока – закрепить навыки составления разветвляющихся и циклических алгоритмов на языке программирования Python.

Задачи:

1. Обучить учеников навыкам анализа программ, содержащих ветвления и циклы.
2. Обучить учащихся приемам автоматизированной обработки данных, представленных в виде массивов и матриц.
3. Сформировать навыки использования циклических конструкций `while` и `repeat` при обработке массивов данных.

Изучив данный видеурок, ученик должен:

Знать:

понятие операций `div` и `mod`, определения и конструкции циклов;

Уметь:

переводить числа из одной системы счисления в другую, использовать конструкции массивов и матриц, использовать конструкции циклов `while` и `repeat`, строить алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием массивов и матриц, циклов `while` и `repeat`.

Учебные вопросы:

1. Цикл `while`.
2. Цикл `repeat`.
3. Массив
4. Матрица.
5. Операция `div`.
6. Операция `mod`.

Урок 6. Задание 21. Анализ программы с циклами и подпрограммами

Цель видеоурока – сформировать навыки анализа модульных программ, содержащих процедуры и функции.

Задачи:

1. Обучить учеников синтаксису описания функций и процедур.
2. Обучить учеников синтаксису вызова функций и процедур.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции и процедуры;

Уметь:

анализировать модульные программы, использовать конструкции функций и процедур, строить алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием функций и процедур.

Учебные вопросы:

1. Подпрограммы.
2. Функция.
3. Процедура.

Урок 7. Задание 24. Поиск и исправление ошибок в программе.

Цель видеоурока – сформировать навыки анализа сложных программ на языке Python.

Задачи:

1. Обучить учеников находить разрешение проблемной ситуации: исправить ошибки в программе с условными операторами.

2. Развить у учеников навыки анализа программ, позволяющие повысить их уровень алгоритмического и логического мышления.

Изучив данный видеоурок, ученик должен:

Знать:

типовые варианты разрешения проблемных ситуаций в программе;

Уметь:

анализировать сложные программы, находить и исправлять ошибки в программе и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием циклов, массивов, матриц, функций и процедур.

Учебные вопросы:

1. Алгоритмическое мышление.
2. Логическое мышление.
3. Поиск и решение ошибок в программе.

Урок 8. Задание 25. Алгоритмы обработки массивов

Цель видеоурока – сформировать умения обучаемых по обработке структурированных строковых и символьных данных.

Задачи:

Развить у учащихся навыки построения алгоритмов, при решении которых используются все основные конструкции языка программирования.

Изучив данный видеоурок, ученик должен:

Знать:

принципы анализа и обработки программ с массивами и матрицами;

Уметь:

строить и реализовать алгоритмы программ, анализировать сложные программы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием массивов и матриц.

Учебные вопросы:

1. Составление алгоритмов и программ.

2. Обработка массивов и матриц.

3. Построение и реализация алгоритмов программ.

Урок 9. Задание 27. Обработка массивов, символьных строк и последовательностей.

Цель видеоурока – закрепить навыки обучаемых по составлению алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python.

Задачи:

Развить у учеников практические навыки и умения построения и реализации алгоритмов на языке программирования Python.

Изучив данный видеоурок, ученик должен:

Знать:

понятие символьных строк и последовательностей;

Уметь:

анализировать и обрабатывать программы с массивами, матрицами, символьными строчками и последовательностями, анализировать сложные программы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием массивов, матриц, символьных строчек и последовательностей.

Учебные вопросы:

1. Символьные строки и последовательности.
2. Массивы и матрицы.
3. Обработка массивов и матриц.

2.3. Проектирование и реализация тренингово-практического (коммуникативного) модуля

Тренингово-практический (коммуникативный модуль) модуль электронного учебно-методического комплекса также, как и теоретико-познавательный состоит из 9 видеороликов (таблица 11). В каждом видеоуроке представлены примеры решения типовых заданий из бланков ЕГЭ по информатике (Приложение Б) и учащимся предлагается по образцу повторить и самостоятельно прорешать задачи, данные в видео. Длительность одного видеоурока – 5-10 минут.

Таблица 11 – План теоретико-познавательного модуля ЭУМК

Номер урока	Тема видеоурока	Описание
2	Задание 8. Анализ программ с циклами	Практический видеоурок позволяет учащимся закрепить навыки составления алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python
3	Задание 11. Рекурсивные алгоритмы	Практический видеоурок позволяет учащимся закрепить навыки составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции
4	Задание 19. Обработка массивов и матриц	Практический видеоурок позволяет учащимся закрепить навыки составления алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python
5	Задание 20. Анализ программы с циклами и условными операторами	Практический видеоурок позволяет учащимся закрепить навыки составления разветвляющихся и циклических алгоритмов на языке программирования Python
6	Задание 21. Анализ программы с циклами и подпрограммами	Практический видеоурок позволяет учащимся закрепить навыки анализа модульных программ, содержащих процедуры и функции
7	Задание 24. Поиск и исправление ошибок в программе	Практический видеоурок формирует у учащихся навыки анализа сложных программ на языке Python
8	Задание 25. Алгоритмы обработки массивов	Практический видеоурок позволяет учащимся закрепить навыки по составлению алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python
9	Задание 27. Обработка массивов, символьных строк и последовательностей	Практический видеоурок позволяет учащимся закрепить навыки обучаемых по обработке структурированных строковых и символьных данных

Для лучшего понимания следует разобрать методические указания для тренингово-практический (коммуникативный модуль) модуля по изучению видеокурса «Подготовка к ЕГЭ по информатике с использованием языка программирования Python».

Урок 2. Задание 8. Анализ программ с циклами

Цель видеоурока – закрепить навыки составления алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение объявлять переменные.
2. Закрепить у учащихся умение работать с операторами присваивания, ввода и вывода данных.
3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

основные понятия языка программирования Python, типы данных, операторы, структуру программы, базовые конструкции структурного программирования;

Уметь:

строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием циклов.

Учебные вопросы:

1. Объявление переменных.
2. Оператор присваивания.
3. Оператор вывода.
4. Цикл while.
5. Формулы арифметической прогрессии.

6. Составление алгоритмов и программ.

Урок 3. Задание 11. Рекурсивные алгоритмы.

Цель видеоурока – закрепить навыки составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение задавать выход из рекурсии.
2. Закрепить у обучаемых навык использования определения и записи на языке программирования рекуррентных формул.
3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции, оператора `if`, ключевого слова `return`, определение и конструкцию рекурсии;

Уметь:

использовать конструкции функции и рекурсии, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования простейших программ с использованием функции и рекурсии.

Учебные вопросы:

1. Функция.
2. Оператор `if`.
3. Ключевое слово `return`.
4. Рекурсия.

Урок 4. Задание 19. Обработка массивов и матриц.

Цель видеоурока – закрепить навыки составления алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение анализировать программы, содержащие ветвления и циклы.
2. Закрепить у учащихся умение обрабатывать массивы данных и работать с циклом `repeat`.
3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие цикла `for`, определения и конструкции массива и матрицы;

Уметь:

использовать конструкции массивов и матриц, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием массивов и матриц.

Учебные вопросы:

1. Цикл `for`.
2. Цикл `while`.
3. Цикл `repeat`.
4. Массив.
5. Матрица.

Урок 5. Задание 20. Анализ программы с циклами и условными операторами.

Цель видеоурока – закрепить навыки составления разветвляющихся и циклических алгоритмов на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение анализировать программы, содержащие ветвления и циклы.

2. Закрепить у учащихся умение использовать приемы автоматизированной обработки данных, представленных в виде массивов и матриц.

3. Закрепить у учеников навыки использования циклических конструкций `while` и `repeat` при обработке массивов данных.

4. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие операций `div` и `mod`, определения и конструкции циклов;

Уметь:

переводить числа из одной системы счисления в другую, использовать конструкции массивов и матриц, использовать конструкции циклов `while` и `repeat`, строить алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием массивов и матриц, циклов `while` и `repeat`.

Учебные вопросы:

1. Цикл `while`.
2. Цикл `repeat`.
3. Массив
4. Матрица.

5. Операция div.

6. Операция mod.

Урок 6. Задание 21. Анализ программы с циклами и подпрограммами

Цель видеоурока – закрепить навыки анализа модульных программ, содержащих процедуры и функции при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников знание синтаксиса описания функций и процедур.

2. Закрепить у учеников знание синтаксиса вызова функций и процедур.

3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции и процедуры;

Уметь:

анализировать модульные программы, использовать конструкции функций и процедур, строить алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования программ с использованием функций и процедур.

Учебные вопросы:

1. Подпрограммы.

2. Функция.

3. Процедура.

Урок 7. Задание 24. Поиск и исправление ошибок в программе.

Цель видеоурока – закрепить навыки анализа сложных программ на языке Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение находить разрешение проблемной ситуации: исправить ошибки в программе с условными операторами.

2. Закрепить у учеников навыки анализа программ, позволяющие повысить их уровень алгоритмического и логического мышления.

3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

типовые варианты разрешения проблемных ситуаций в программе;

Уметь:

анализировать сложные программы, находить и исправлять ошибки в программе и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием циклов, массивов, матриц, функций и процедур.

Учебные вопросы:

1. Алгоритмическое мышление.
2. Логическое мышление.
3. Поиск и решение ошибок в программе.

Урок 8. Задание 25. Алгоритмы обработки массивов

Цель видеоурока – закрепить умения обучаемых по обработке структурированных строковых и символьных данных при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учащихся навыки построения алгоритмов, при решении которых используются все основные конструкции языка программирования.

2. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

принципы анализа и обработки программ с массивами и матрицами;

Уметь:

строить и реализовать алгоритмы программ, анализировать сложные программы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием массивов и матриц.

Учебные вопросы:

1. Составление алгоритмов и программ.
2. Обработка массивов и матриц.
3. Построение и реализация алгоритмов программ.

Урок 9. Задание 27. Обработка массивов, символьных строк и последовательностей.

Цель видеоурока – закрепить навыки обучаемых по составлению алгоритмов и программ автоматизированной обработки массивов данных на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников практические навыки и умения построения и реализации алгоритмов на языке программирования Python.
2. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие символьных строк и последовательностей;

Уметь:

анализировать и обрабатывать программы с массивами, матрицами, символьными строчками и последовательностями, анализировать сложные программы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования сложных программ с использованием массивов, матриц, символьных строчек и последовательностей.

Учебные вопросы:

1. Символьные строки и последовательности.
2. Массивы и матрицы.
3. Обработка массивов и матриц.

Для практики учащиеся использовали систему программирования Visual Studio (рисунок 5). От учеников требовалось самостоятельное выполнение заданий, предложенных в практическом видеоуроке.

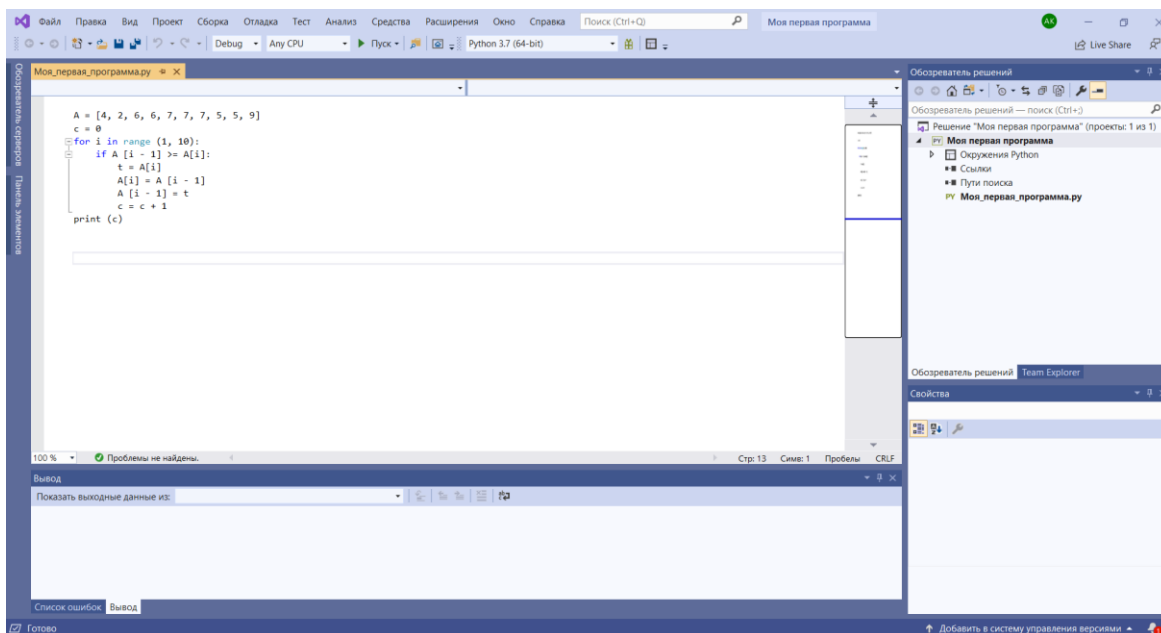


Рисунок 5 – Система программирования Visual Studio

2.4. Проектирование и реализация контрольного модуля

Контрольный модуль электронного учебно-методического комплекса входит во все 9 уроков ЭУМК (таблица 8).

Для контроля и оценивания знания учащихся в разработанном электронном обучающем курсе использованы:

1. Тесты и анкеты (Рисунок 6). Данная форма контроля на сегодняшний день является одной из самых востребованных. Обучение может начинаться с

входного тестирования и анкетирования, сопровождаться текущим контролем с помощью заданий в тестовой форме и заканчиваться объективным тестированием учебных достижений. Также тесты позволяют наладить самоконтроль.

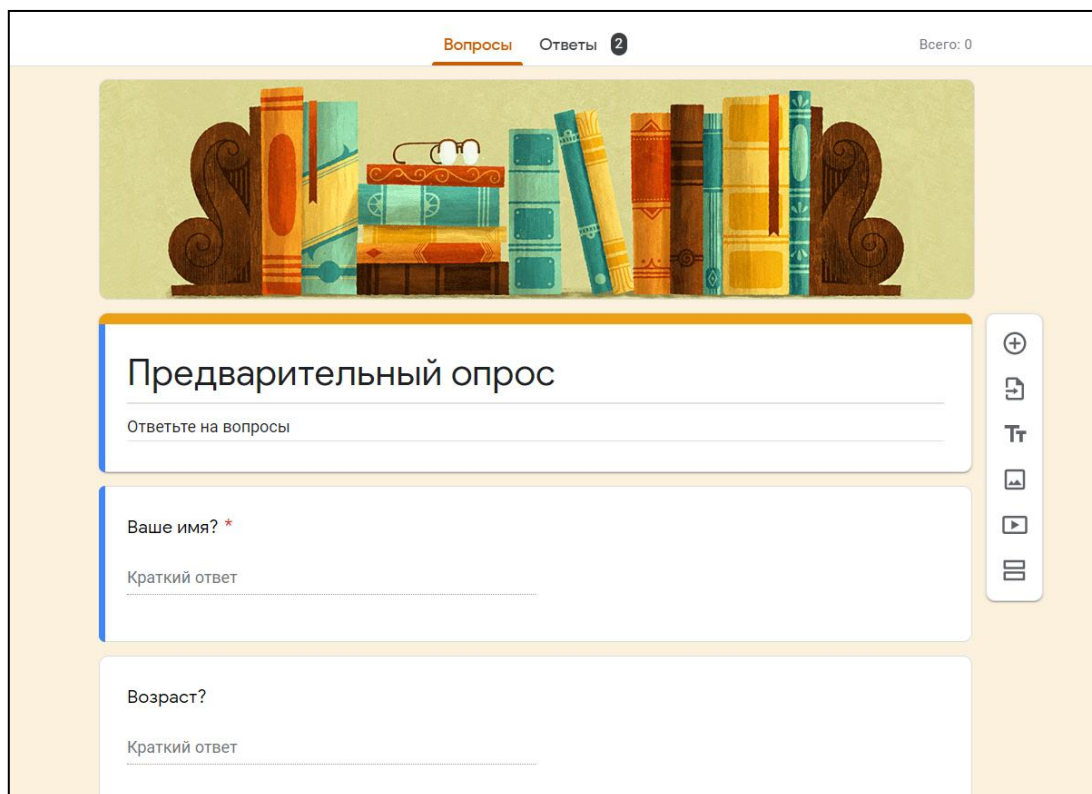


Рисунок 6 – Предварительный опрос

Оценивание знаний в такой форме контроля выполняется по 100-бальной системе. От количества выполненных пунктов теста и полученных баллов зависит итоговая оценка.

Оценка «5» (отлично) – 90%-100% верных ответов.

Оценка «4» (хорошо) – 70%-90% верных ответов.

Оценка «3» (удовлетворительно) – 50%-70% верных ответов.

Оценка «2» (неудовлетворительно) – 0%-50% верных ответов.

2. Типовые задания из ЕГЭ. В качестве домашнего задания (Приложение В) ученикам выдаются варианты с упражнениями по теме для выполнения на оценку (Рисунок 7).

Не забудьте выполнить домашнее задание!

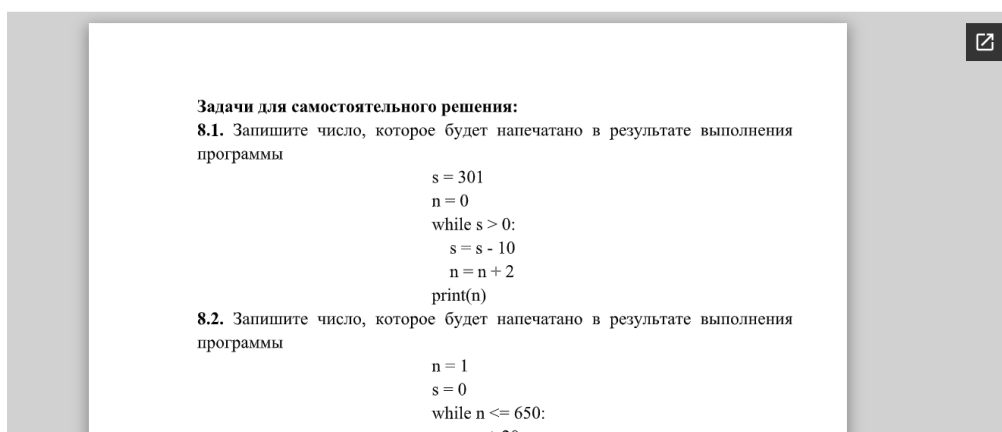


Рисунок 7 – Домашнее задание

Учащимся предлагается выполнить 5 задание, за их выполнение учащиеся получают баллы, от количества которых зависит оценка.

Оценка «5» (отлично) – 5 верно решенных заданий.

Оценка «4» (хорошо) – 4 верно решенных заданий.

Оценка «3» (удовлетворительно) – 3 верно решенных заданий.

Оценка «2» (неудовлетворительно) – 0-2 верно решенных заданий.

3. Упражнения на закрепление пройденного учебного материала, разработанные в облачном сервисе LearningApps. Чтобы ученики могли выполнять задания, учитель должен представить им ссылки на нужные упражнения. Данная форма контроля отличается от предыдущих тем, что практический материал представлен в **игровом формате**, что делает проверку знаний более интересной и увлекательной (Приложение Г).

Облачная платформа LearningApps позволяет организовать виртуальный класс, в который добавляются ученики (Рисунок 8).

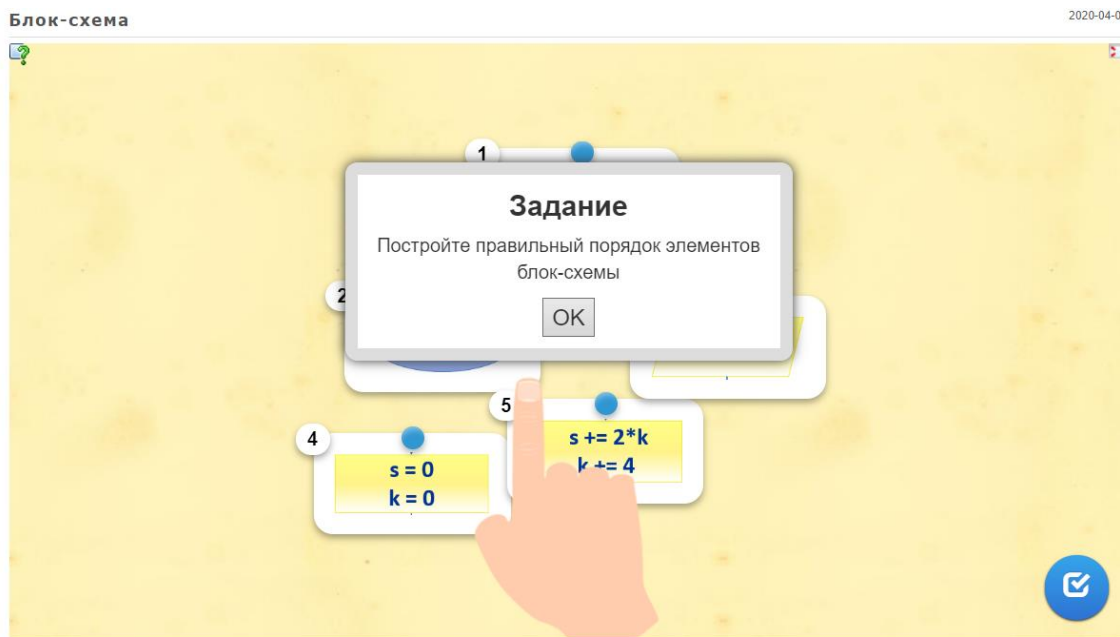


Рисунок 8 – Пример упражнения

При прохождении видеокурса «Подготовка к ЕГЭ по информатике с использованием Python» будет формироваться итоговая оценка учащихся на основе всех выполненных работ и упражнений.

Выводы по главе 2

При проектировании и реализации электронного учебно-методического комплекса по разделу «Алгоритмизация и программирование» был проведен анализ дидактических, программно-технологических и технических характеристик, представлено обоснование выбранных образовательных платформ для реализации курса. ЭУМК состоит из трех основных модулей: теоретико-познавательного, тренингово-практического (коммуникативного) и контрольного. Выполнена постановка целей и задач по каждому модулю, определены предметные и межпредметные результаты учащихся при освоении каждого образовательного ресурса.

Глава 3. Оценка эффективности разработанного электронного учебно-методического комплекса

3.1. Общая характеристика исследования

Для создания и реализации электронного учебно-методического комплекса для подготовки к Единому государственному экзамену по информатике был проведен педагогический эксперимент. Данный эксперимент был направлен на изучение реально складывающегося опыта организации учебного процесса. Также были реализованы следующие цели:

1. Анализ проблем теоретических и методических особенностей обучения разделу «Алгоритмизация и программирование» в общеобразовательной школе.

2. Построение методики обучения разделу «Алгоритмизация и программирование» и внедрение её в практику обучения учащихся 10-11 классов.

3. Проверка эффективности и результативности разработанной методики.

Проведение педагогического эксперимента было разделено на три этапа:

1. **Первый этап – поисковый.** Нами было рассмотрено и проанализировано следующее: состояние проблемы обучения разделу «Алгоритмизация и программирование» в средней школе; оптимальная последовательность представления учебных материалов; теоретические и методологические предпосылки разработки методики обучения разделу «Алгоритмизация и программирование» в курсе информатики и ИКТ средней школы.

Для этого был проведен анализ соответствующей научной, педагогической, дидактической, методической, психологической литературы, учебных планов и программ школьного предмета «Информатика и ИКТ», а также существующих учебных и методических пособий (Глава 1).

В результате поискового этапа эксперимента была обоснована актуальность темы исследования, связанная с выявившимися противоречием между необходимостью создания учебных пособий и методических рекомендаций, позволяющих организовать процесс обучения разделу «Алгоритмизация и программирование», и недостаточной разработанностью научно-практических рекомендаций в этой области.

Анализ актуальности и выявленных противоречий позволил сформулировать проблему исследования. Данная проблема заключается в обосновании дополнительных методических рекомендаций по обучению разделу «Алгоритмизация и программирование».

2. Второй этап – формирующий. Цель данного этапа педагогического эксперимента – детальная разработка каждого компонента методики обучения разделу «Алгоритмизация и программирование» учащихся средних школ: определение целей и задач обучения; обоснование принципов отбора содержания обучения с последующей его детализацией и преобразованием в учебный материал; выбор оптимальных методов, средств и форм организации учебного процесса. Главная задача второго этапа исследования – разработка методических рекомендаций по использованию электронных образовательных ресурсов.

Результаты этого этапа педагогического эксперимента позволили сформулировать **гипотезу** нашего исследования, согласно которой использование разработанной программно-методического комплекса в процессе обучения алгоритмизации и программированию учащихся средних школ, это позволит повысить уровень подготовленности выпускников к единому государственному экзамену по предмету «Информатика и ИКТ».

3. Третий этап – контрольный. Данный этап нужен для проверки выдвинутой гипотезы, он заключается в исследовании эффективности и результативности разработанной методики обучения разделу «Алгоритмизация и программирование» учащихся средних школ.

3.2. Методика проведения и результаты педагогического эксперимента

Педагогический эксперимент проводимого исследования был организован на базе МБУ «Лицей №57» в рамках прохождения педагогической и преддипломной практик. Изучение предмета «Информатика и ИКТ» в 10-11-х классах, реализуемого в рамках ФГОС СОО и учебного плана школы, проходит на углубленном и базовом уровнях.

МБУ «Лицей №57» оснащен 4-мя оборудованными кабинетами информатики. База практики оборудована следующим: белой классной доской под маркер, мультимедийным проектором и белым тканевым экраном, 32 компьютерными столами, 50 компьютерными стульями, 48 ученическими партами, 84 ученическими стульями, сервером, локальной сетью, 45 персональными компьютерами, интерактивными обучающими системами Smart.

К эксперименту были привлечены обучающиеся 11 «А» класса.

Всего в эксперименте участвовало 25 учащихся. Во время проведения педагогического эксперимента были использованы такие эмпирические методы исследования как наблюдение, анкетирование, экспертная оценка, тестирование.

Организация и методика проведения педагогического эксперимента разделена на 2 этапа:

1. Организация и методика проведения констатирующего этапа педагогического эксперимента

Констатирующий этап педагогического эксперимента проводился с целью анализа состояния сформированности предметных и межпредметных результатов обучения выпускников средней школы при освоении раздела «Алгоритмизация и программирование» в углубленном курсе информатики. На этом этапе решались задачи: формирование выборки обучающихся для участия в эксперименте, а также определение диагностического инструментария. Для проведения эксперимента потребовалось выбрать учащихся, изучавших язык программирования Python. Выборка составила 10 человек. Для проверки уровня

сформированности предметных и межпредметных результатов обучения разделу «Алгоритмизация и программирование» на констатирующем этапе ученикам было предложено решить КИМ ЕГЭ по информатике (типовые задания по программированию), а также проведен входной тест в режиме онлайн по авторской методике Полякова К.Ю. [5] (рисунок 9).

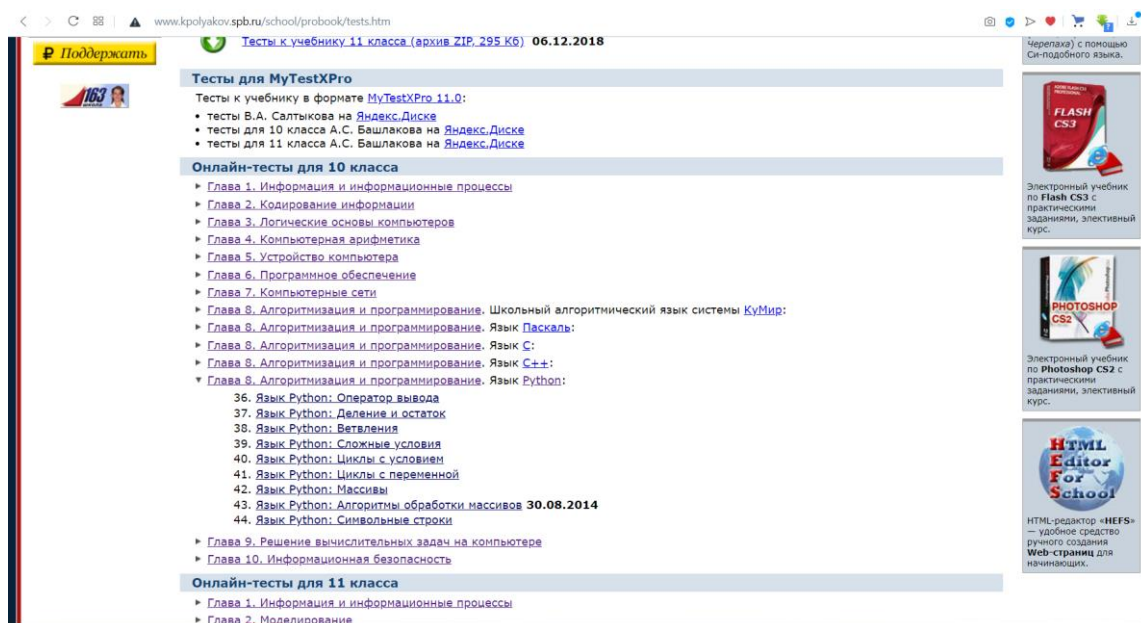


Рисунок 9 – Электронный образовательный ресурс К.Ю.Полякова для подготовки к ЕГЭ по информатике

Тест состоит из 20 вопросов. Типовые задания из бланка ЕГЭ (Приложение Д) и тест (Приложение Е) проверяют знания учащихся по следующим темам: оператор вывода, деление и остаток, ветвления, сложные условия, циклы с условием, циклы с переменной, массивы, алгоритмы обработки массивов, символьные строки, а также тестирует умение учащихся решать типовые задания из бланков ЕГЭ.

Количество правильных ответов учащихся свидетельствует об уровне сформированности предметных и межпредметных результатов обучения разделу «Алгоритмизация и программирование» обучающихся. Высокий уровень предусматривает, что учащийся дает правильные ответы или решает не менее 90% заданий, базовый уровень – от 70% до 89% заданий, начальный – 50% до 69% и низкий – менее 50% (таблицы 12 и 13).

Таблица 12 – Описание уровней результатов решения типовых заданий из бланка ЕГЭ (по программированию)

Уровень	Процент правильных ответов	Количество правильных ответов	Оценка
Высокий	не менее 90%	не менее 18	5
Базовый	70% – 89%	14 – 17	4
Начальный	50% – 69%	10 – 13	3
Низкий	менее 50%	Менее 9	2

Таблица 13 – Описание уровней результатов тестирования

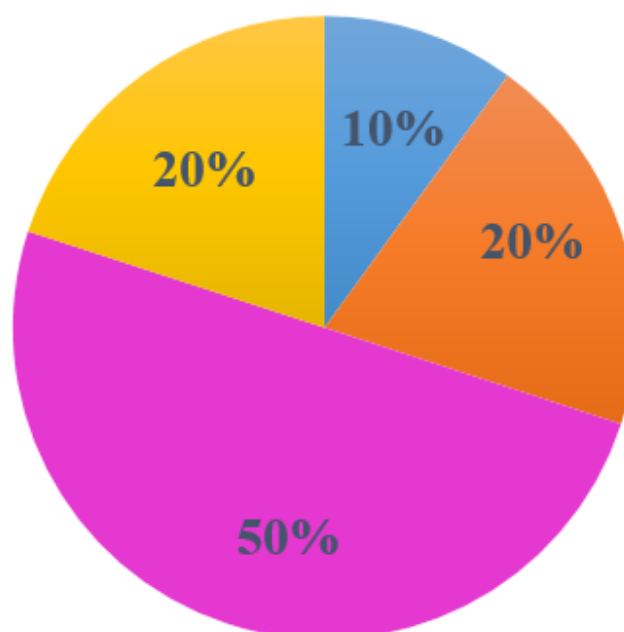
Уровень	Процент правильных ответов	Количество правильных ответов	Оценка
Высокий	не менее 90%	8	5
Базовый	70% – 89%	6 – 7	4
Начальный	50% – 69%	4 – 5	3
Низкий	менее 50%	Менее 3	2

Результаты тестирования учащихся экспериментальной группы на констатирующем этапе педагогического эксперимента представлены в таблицах 14 и 15.

Таблица 14 – Уровень предметных и межпредметных результатов обучения по разделу «Алгоритмизация и программирование» на констатирующем этапе по результатам решения типовых заданий из бланка ЕГЭ

Уровень	Оценка	Экспериментальная группа			
		Количество оценок	Процент выполнения	Общее количество баллов	Среднее значение результата
Высокий	5	1	10%	5	
Базовый	4	2	20%	8	
Начальный	3	5	50%	15	
Низкий	2	2	20%	4	
Всего:		10		32	3,2

Данные проведенной проверки по решению типовых заданий показали, что среди обучаемых экспериментальной группы на низком уровне – 20% учащихся, на начальном уровне – 50%; на базовом уровне – 20%; на высоком уровне – 10% (рисунок 10).



■ Высокий ■ Базовый ■ Начальный ■ Низкий

Рисунок 10 – Уровень предметных результатов обучения в экспериментальной группе на констатирующем этапе эксперимента по результатам решения типовых заданий из бланка ЕГЭ

Таблица 15 – Уровень предметных и межпредметных результатов обучения по разделу «Алгоритмизация и программирование» на констатирующем этапе по результатам тестирования

Уровень	Оценка	Экспериментальная группа			
		Количество оценок	Процент выполнения	Общее количество баллов	Среднее значение результата
Высокий	5	2	20%	10	
Базовый	4	3	30%	12	
Начальный	3	4	40%	12	
Низкий	2	1	10%	2	
Всего:		10		36	3,6

Данные проведенного тестирования показали, что среди обучаемых экспериментальной группы на низком уровне – 10% учащихся, на начальном уровне – 40%; на базовом уровне – 30%; на высоком уровне – 20% (рисунок 11).

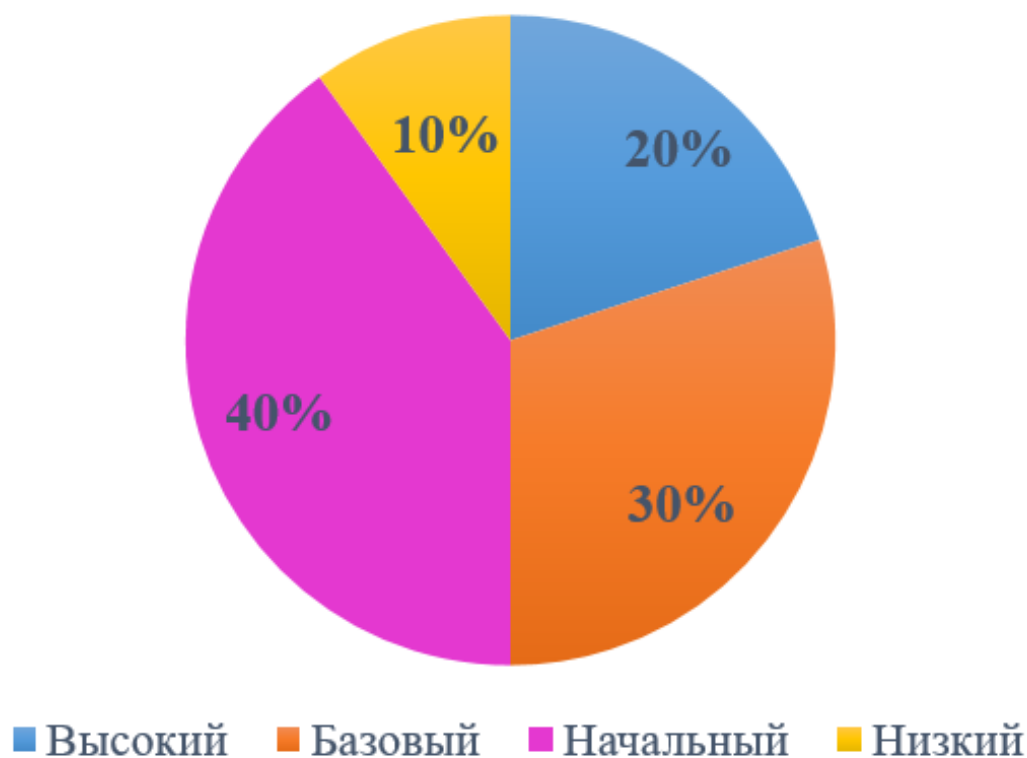


Рисунок 11 – Уровень предметных результатов обучения в экспериментальной группе на констатирующем этапе эксперимента по результатам тестирования

2. Организация и методика проведения контрольного этапа педагогического эксперимента

Цель контрольного этапа – проверка эффективности применения разработанного электронного учебно-методического комплекса в образовательном процессе учащихся 11-х классов для формирования предметных и межпредметных результатов обучения по разделу «Алгоритмизация и программирование».

На контрольном этапе педагогического эксперимента учащиеся экспериментальной группы осваивали раздел «Алгоритмизация и программирование» с использованием разработанного ЭУМК. После изучения раздела с учащимися экспериментальной группы был вторично проведен контрольный тест на проверку уровня предметных и межпредметных результатов обучения в соответствии с ФГОС СОО [1].

Таблица 16 – Уровень предметных и межпредметных результатов обучения по разделу «Алгоритмизация и программирование» на контрольном этапе по результатам решения типовых заданий из бланка ЕГЭ

Уровень	Оценка	Экспериментальная группа			
		Количество оценок	Процент выполнения	Общее количество баллов	Среднее значение результата
Высокий	5	1	10%	5	
Базовый	4	5	50%	20	
Начальный	3	3	30%	9	
Низкий	2	1	10%	2	
Всего:		10		36	3,6

Данные проведенной проверки по решению типовых заданий (таблица 16) показали, что учащиеся экспериментальной группы получили более высокие предметные и межпредметные результаты обучения: на низком уровне – 10% обучающихся; на начальном уровне – 30%; на базовом уровне – 50% и на высоком уровне – 10% обучающихся (рисунок 12).

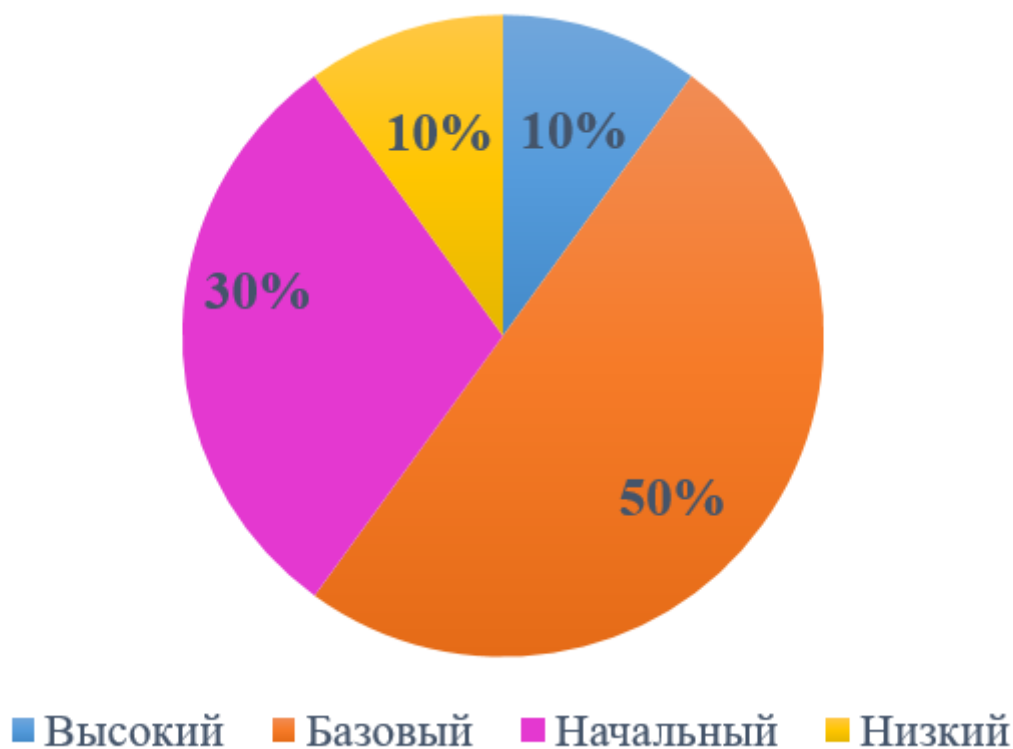


Рисунок 12 – Уровень предметных результатов обучения в экспериментальной группе на контрольном этапе эксперимента по результатам решения типовых заданий из бланка ЕГЭ

Таблица 17 – Уровень предметных и межпредметных результатов обучения по разделу «Алгоритмизация и программирование» на контрольном этапе по результатам тестирования

Уровень	Оценка	Экспериментальная группа			
		Количество оценок	Процент выполнения	Общее количество баллов	Среднее значение результата
Высокий	5	2	20%	10	
Базовый	4	6	60%	24	
Начальный	3	2	20%	6	
Низкий	2	0	0%	0	
Всего:		10		40	4,0

Данные проведенного тестирования (таблица 17) также показали более высокие предметные и межпредметные результаты обучения: на низком уровне – 0% обучающихся; на начальном уровне – 20%; на базовом уровне – 60% и на высоком уровне – 20% обучающихся (рисунок 13).

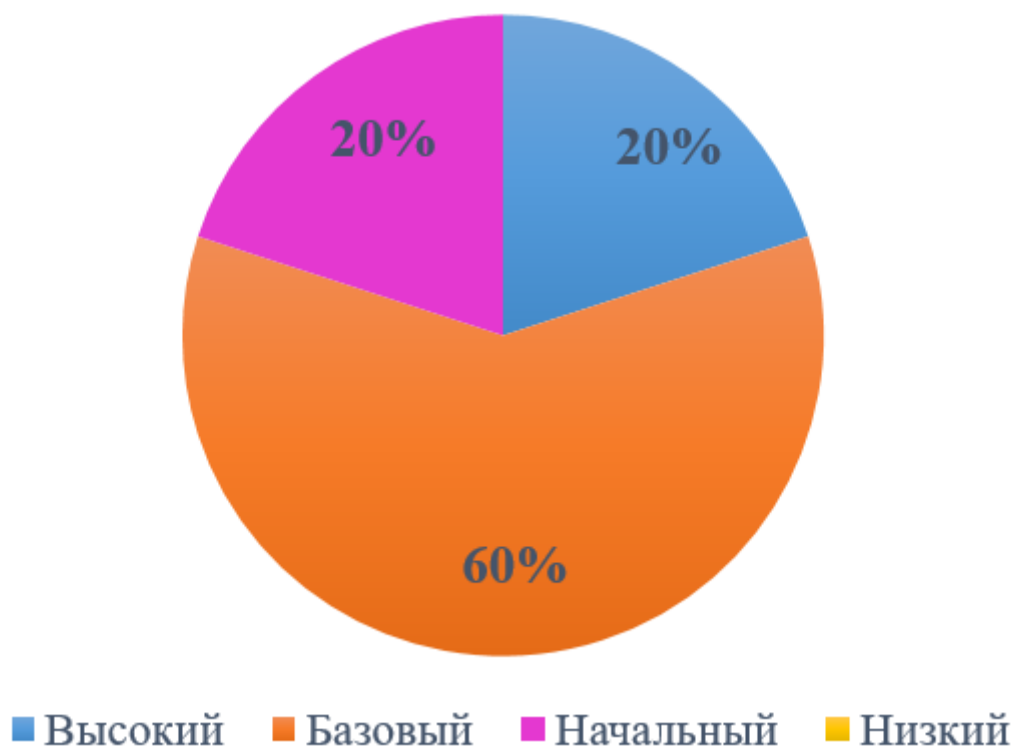


Рисунок 13 – Уровень предметных результатов обучения в экспериментальной группе на контрольном этапе эксперимента по результатам тестирования

Можно сделать вывод: предметные и межпредметные результаты обучения по разделу «Алгоритмизация и программирование» после обучения по предложенной методике значительно улучшились. Прослеживается снижение количества учеников на низком и начальном уровнях. В то же время, можно увидеть динамику роста количества обучающихся, находившихся на базовом. Количество учеников на высоком уровне не изменяется (рисунок 14 и 15).

Обучающиеся, имеющие на констатирующем этапе высокий уровень предметных и межпредметных результатов, в ходе прохождения видеокурса совершенствовали свои знания, практические умения и навыки по разделу «Алгоритмизация и программирование». Средняя оценка учащихся возросла примерно на 15%.

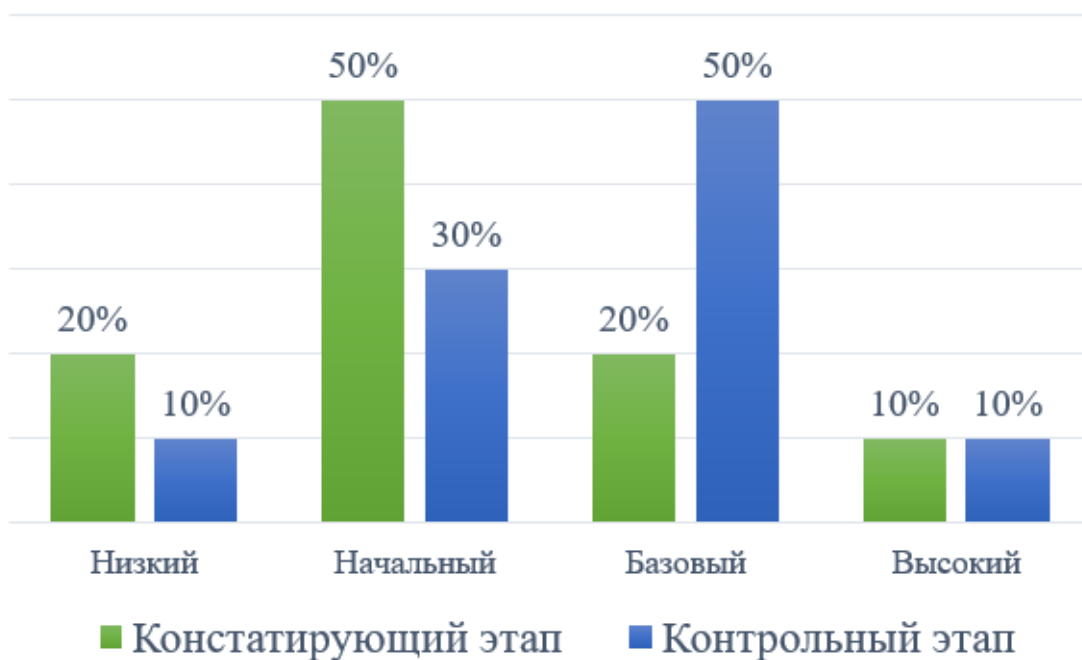


Рисунок 14 – Сравнение уровней сформированности предметных результатов обучения в экспериментальной группе на констатирующем и контрольном этапах эксперимента по результатам решения типовых заданий из бланка ЕГЭ

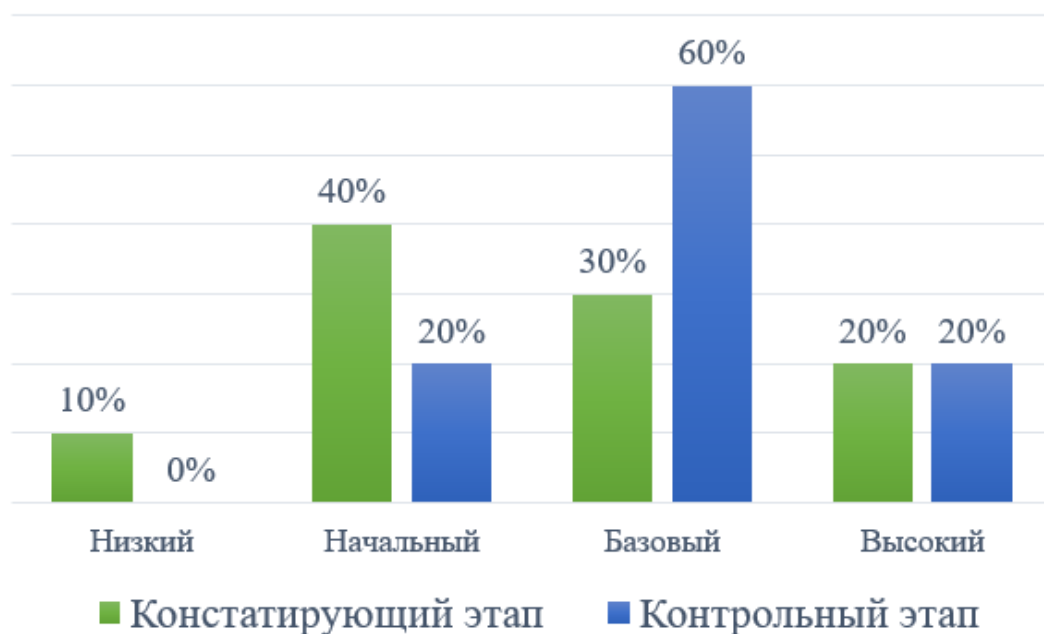


Рисунок 15 – Сравнение уровней сформированности предметных результатов обучения в экспериментальной группе на констатирующем и контрольном этапах эксперимента по результатам тестирования

Результаты педагогического эксперимента показали, что организация образовательного процесса с использованием разработанного электронного учебно-методического комплекса способствует повышению уровня предметных и межпредметных результатов обучения разделу «Алгоритмизация и программирование» учащихся средних школ в соответствии с образовательным стандартом общего среднего образования, а также положительно влияет на их уровень подготовки к ЕГЭ по информатике.

Также учащимся экспериментальной группы было предложено дополнительно пройти анкетирование (Приложение Ж) для выявления уровня интереса учащихся к деятельности, связанной с программированием. По результатам анкетирования 90% учащихся отметили положительный эффект и результативность использования разработанного электронного учебно-методического комплекса (видеоуроки, тесты, интерактивные упражнения) при освоении учебного материала раздела «Алгоритмизация и программирование».

Таким образом, мы можем сделать вывод о достоверном повышении результативности обучения по предлагаемой методике и подтверждении гипотезы, выдвинутой в начале исследований.

ЗАКЛЮЧЕНИЕ

Нами был проведен анализ теоретических основ разработки программно-методического обеспечения для подготовки выпускников средних школ к единому государственному экзамену по разделу «Алгоритмизация и программирование», обоснованы его структура и содержание, определены методические особенности, предоставляющих возможность организовать реализацию электронного учебно-методического комплекса.

Для проектирования и создания ЭУМК были рассмотрены его дидактические, программно-технологические и технические характеристики. Были определены и обоснованы структура, цели и задачи, содержание, методы, формы и средства обучения с использованием электронного учебно-методического комплекса.

С использованием современных web-сервисов и сред разработки электронных образовательных ресурсов разработан ЭУМК, представленный в виде видеокурса «Подготовка к ЕГЭ по информатике с использованием языка программирования Python» с интерактивными упражнениями и контрольным тестированием.

Разработанный ЭУМК представляет собой “открытую” систему, доступную для корректировки с учетом развития информатики и социального заказа общества. На её основе могут быть построены вариативные спецкурсы с учетом имеющихся условий и ресурсов.

Результаты исследования были опубликованы в электронном журнале в "Инновации. Наука. Образование", 2019, № 10, С. 7-9. в статье "Подготовка учащихся к единому государственному экзамену по информатике (по разделу «Программирование»)" [14].

В рамках поставленных задач выполненное исследование можно считать законченным.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ N 24480. Федеральный Государственный образовательный стандарт среднего общего образования: утвержден приказом Министерства образования и науки Российской Федерации от 17 мая 2012г., №413 / Министерство образования и науки Российской Федерации. – Москва: 2012г. – URL: http://school20.tgl.ru/sp/pic/File/2014/iyun/prikaz_MON_Ob_utverjdenii_federalnogo_gosudarstvennogo_obrazovatel'nogo_standarta_srednego_polnogo_obshhego_obrazovaniya.pdf. – (дата обращения:01.05.2020) – Текст: электронный.

2. ГОСТ N 258. Федеральный базисный учебный план и примерные учебные планы для образовательных учреждений Российской Федерации, реализующих программы общего образования: утвержден Постановлением Правительства Российской Федерации от 09.03.2004 N 258. – Москва: 2012 г. – URL: <http://docs.cntd.ru/document/901895864>. – (дата обращения: 03.05.2020) – Текст: электронный.

3. ГОСТ №2/16-з. Примерная образовательная программа среднего общего образования: одобрена решением федерального учебно-методического объединения по общему образованию, протокол от 28.06.2016г., №2/16-3 / Министерство образования и науки Российской Федерации. – Москва: 2016г. – URL: <https://mosmetod.ru/files/dokumenty/Primernaya-osnovnaya-obrazovatel'naya-programma-srednego-obshhego-obrazovaniya.pdf>. – (дата обращения: 05.05.2020) – Текст: электронный.

4. Информационно-коммуникационные технологии в образовании. Метаданные электронных образовательных ресурсов. Общие положения: ГОСТ Р 55750-2013. – Введ. 2015-01-01. – М.: Стандартинформ, 2015. – 11 с. – URL: <http://docs.cntd.ru/document/1200107223>. – (дата обращения: 02.05.2020) – Текст: электронный.

5. Информационно-коммуникационные технологии в образовании. Образовательные интернет-порталы федерального уровня. Рубрикация информационных ресурсов: ГОСТ Р 52657-2006. – Введ. 2008-07-01. – М.:

Стандартиформ, 2008. – 12 с. – URL: <http://docs.cntd.ru/document/1200053104>.
–(дата обращения: 02.05.2020) – Текст: электронный.

6. Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения: ГОСТ Р 53620-2009. – Введ. 2011-01-01. – М.: Стандартиформ, 2011. – 10 с. – URL: <http://docs.cntd.ru/document/1200082196>. – (дата обращения: 02.05.2020) – Текст: электронный.

7. Информационно-коммуникационные технологии в образовании. Электронные учебно-методические комплексы. Требования и характеристики.: ГОСТ Р 55751-2013. – Введ. 2015-01-01. – М.: Стандартиформ, 2008. – 15 с. – URL: <http://docs.cntd.ru/document/1200108264>. – (дата обращения: 03.05.2020) – Текст: электронный.

8. Бородин, М.Н. Информатика. Программы для образовательных организаций. 2 – 11 классы / М.Н. Бородин. – М.: БИНОМ. Лаборатория знаний, 2015. – 576 с.

9. Босова, Л.Л. Информатика. Базовый уровень. 10-11 классы: методическое пособие / Л.Л. Босова, А.Ю. Босова и др. – М.: БИНОМ. Лаборатория знаний, 2020. – 470, [10] с. – URL: <https://lbz.ru/books/740/9614/>. – (дата обращения: 20.05.2020). – Текст: электронный.

10. Григорьев, С.Г. Информатизация образования. Фундаментальные основы. / С.Г. Григорьев, В.В. Гриншкун. – М, 2005. – 231 с.

11. Ершов А.П. Программирование - вторая грамотность / А.П. Ершов / Проблемы информатики. – М., 2015. – URL: http://ershov.iis.nsk.su/ru/second_literacy/article. – (дата обращения: 19.05.2020). – Текст: электронный.

12. Ершов, А.П. Введение в теоретическое программирование / А.П. Ершов / М.: Наука, 1977. – 280 с. – URL: <http://bookre.org/reader?file=1499378>. – (дата обращения: 21.05.2020). – Текст: электронный.

13. Зорина, Е.М.: ЕГЭ 2020. Информатика. Сборник заданий. 350 заданий с ответами. / Е.М. Зорина, М.В. Зорин. — Москва: Эксмо, 2019. — 240 с.

14. Коновалова, А.Д. Подготовка учащихся к единому государственному экзамену по информатике (по разделу «Программирование») / А.Д. Коновалова, И.П. Дудина / Инновации. Наука. Образование. – Тольятти. - 2019. - № 10 (11). - С. 7 - 9. - URL: <https://elibrary.ru/item.asp?id=41351787>. – (дата обращения: 07.05.2020). – Текст: электронный.

15. Лапчик, М.П. Методика преподавания информатики: Учеб. пособие для студ. пед. Вузов: учебное пособие / М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер; под общей ред. М. П. Лапчика; Издательский центр «Академия». – М., 2001. — 624 с. - URL: http://zjkin.moy.su/nauka/metodika_prepodavaniya_informatiki_lapchik-semakin.pdf. – (дата обращения: 09.05.2020). – Текст: электронный.

16. Муртузалиева, А.С. О значимости изучения алгоритмизации и программирования в школьном курсе информатики» / А.С. Муртузалиева / Вестник социально педагогического института. – М., 2015. – URL: <https://cyberleninka.ru/article/n/o-znachimosti-izucheniya-algoritmizatsii-i-programmirovaniya-v-shkolnom-kurse-informatiki/viewer>. – (дата обращения: 25.05.2020). – Текст: электронный.

17. Окулов, С.М. Основы программирования / С.М. Окулов – М.: Лаборатория знаний, 2020. – 339 с. – URL: <https://nashol.me/20200205118203/osnovi-programmirovaniya-okulov-s-m-2002.html>. – (дата обращения: 20.05.2020). – Текст: электронный.

18. Окулов, С.М. Программирование в алгоритмах / С.М. Окулов – М.: Лаборатория знаний, 2002. – 341 с. - URL: <https://ru.scribd.com/doc/271063021/Окулов-С-М-Программирование-в-Алгоритмах>. – (дата обращения: 07.05.2020). – Текст: электронный.

19. Осин, А. В. ЭОР нового поколения: открытые образовательные модульные мультимедиа системы / А. В. Осин // Аналитическая записка. – 2011. - URL: https://iite.unesco.org/files/policy_briefs/pdf/ru/eer.pdf. – (дата обращения: 02.02.2020). – Текст: электронный.

20. Парменова, Л.В. Методы достижения метапредметных результатов в обучении программирования в основной школе / Л.В. Парменова // Ярославский педагогический вестник. – М., 2016. – URL: <https://cyberleninka.ru/article/n/metody-dostizheniya-metapredmetnyh-rezultatov-v-obuchenii-programmirovaniyu-v-osnovnoy-shkole/viewer>. – (дата обращения: 10.05.2020). – Текст: электронный.

21. Поляков, К.Ю. Алгоритмизация и программирование / К.Ю. Поляков // Информатика. – 2016. – URL: <https://www.kpolyakov.spb.ru/download/inf-2016-04.pdf>. – (дата обращения: 05.05.2020). – Текст: электронный.

22. Поляков, К.Ю. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. / К.Ю. Поляков, Е.А. Еремин. – М.: БИНОМ, 2013. - Ч. 2 – 304 с.

23. Поляков, К.Ю. Как нам реорганизовать ЕГЭ по информатике? / К.Ю. Поляков // Точка зрения. – 2019. – URL: <https://www.kpolyakov.spb.ru/download/ivsh3-2019.pdf>. – (дата обращения: 03.05.2020). – Текст: электронный.

24. Поляков, К.Ю. Язык Python: избранные алгоритмы. Часть 1. / К.Ю. Поляков // Первое сентября. – 2014. – URL: <https://www.kpolyakov.spb.ru/download/inf-2014-09b.pdf>. – (дата обращения: 02.05.2020). – Текст: электронный.

25. Поляков, К.Ю. Язык Python: избранные алгоритмы. Часть 2. / К.Ю. Поляков // Первое сентября. – 2014. – URL: <https://www.kpolyakov.spb.ru/download/inf-2014-10.pdf>. – (дата обращения: 23.05.2020). – Текст: электронный.

26. Самылкина, Н.Н. ЕГЭ 2019. Информатика. Задания, ответы, комментарии. / Н.Н. Самылкина, И.В. Сеницкая, В.В. Соболева. — Москва: Эксмо, 2019. – 240 с.

27. Семакин, И. Г. Информатика. Углубленный уровень: практикум для 10-11 классов: в 2ч. / И. Г. Семакин, Т. Ю. Шеина, Л. В. Шестакова. — М.: БИНОМ, 2013. - Ч.1 – 168 с.

28. Семакин, И. Г. Информатика. Углубленный уровень: практикум для 10-11 классов: в 2ч. / И. Г. Семакин, Е. К. Хеннер, Л. В. Шестакова. — М.: БИНОМ, 2013. - Ч.2 – 120 с.

29. Семакин, И. Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. / И. Г. Семакин, Е. К. Хеннер, Л. В. Шестакова. – М.: БИНОМ, 2014. - Ч.2. – 216 с.

30. Семакин, И.Г. Информатика. Углубленный уровень: методическое пособие для 10-11 классов. / И.Г. Семакин. – М.: БИНОМ. Лаборатория знаний, 2016. – 80 с.

31. Татаринцев, А. И. Электронный учебно-методический комплекс как компонент информационно-образовательной среды педагогического вуза / А. И. Татаринцев // Теория и практика образования в современном мире. – 2012. – С. 367. – URL: <https://moluch.ru/conf/ped/archive/21/1701/>. – (дата обращения: 01.06.2020). – Текст: электронный.

32. Ушаков, Д.М. ЕГЭ 2020. Информатика. 10 тренировочных вариантов экзаменационных работ для подготовки к единому государственному экзамену / Д.М. Ушаков. – Москва: АСТ, 2019. – 181 с.

ЭЛЕКТРОННЫЕ РЕСУРСЫ

1. Единая коллекция Цифровых Образовательных Ресурсов: сайт. - URL: <http://school-collection.edu.ru>. – (дата обращения: 08.05.2020). – Текст: электронный.

2. Единое окно доступа к образовательным ресурсам: сайт. - URL: <http://window.edu.ru>. – (дата обращения: 06.05.2020). – Текст: электронный.

3. Образовательный портал для подготовки к экзаменам «Решу ЕГЭ»: сайт. - URL: <https://inf-ege.sdamgia.ru>. – (дата обращения: 03.02.2020). – Текст: электронный.

4. Сайт Данилова Н.С.: сайт. - URL: <http://dannik.ru/teaching/prog.php>. – (дата обращения: 11.05.2020). – Текст: электронный.

5. Сайт Полякова К.Ю. «Преподавание, наука и жизнь»: сайт. - URL: <https://www.kpolyakov.spb.ru/school/ege.htm>. – (дата обращения: 15.05.2020). – Текст: электронный.

6. Сайт с электронными учебниками для школьников и студентов «11 klasov»: сайт. - URL: <https://s.11klasov.ru/informatics/>. – (дата обращения: 11.05.2020). – Текст: электронный.

7. Google Classroom: сайт. - URL: <https://classroom.google.com/u/0/h>. – (дата обращения: 01.06.2020). – Текст: электронный.

8. LearningApps: сайт. - URL: <https://learningapps.org>. – (дата обращения: 25.05.2020). – Текст: электронный.

9. Moodle: сайт. - URL: <https://moodlecloud.com/app/en/notfound/maiandroid>. – (дата обращения: 16.05.2020). – Текст: электронный.

10. Stepik: сайт. - URL: <https://welcome.stepik.org/ru>. – (дата обращения: 02.02.2020). – Текст: электронный.

11. Udemу: сайт. - URL: <https://www.udemy.com>. – (дата обращения: 15.05.2020). – Текст: электронный.

12. YouTube «Информатик БУ»: сайт. URL: <https://www.youtube.com/channel/UCmUcjDHUkIMhfqBfyHYXYuA>. – (дата обращения: 02.06.2020). – Текст: электронный.

13. YouTube «Информатика. Подготовка к ЕГЭ 2018»: сайт. URL: <https://www.youtube.com/playlist?list=PL66kLi3dt8A5slbxnwIjiYHloR1VGE-UJ>. – (дата обращения: 14.05.2020) – Текст: электронный.

ПРИЛОЖЕНИЯ

Приложение А

Пример конспекта видеоурока с теоретическим материалом

Урок 3. Задание 11. Рекурсивные алгоритмы

Цель видеоурока – сформировать навыки составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python.

Задачи:

1. Обучить учеников задавать выход из рекурсии.
2. Сформировать у обучаемых навык использования определения и записи на языке программирования рекуррентных формул.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции, оператора if, ключевого слова return, определение и конструкцию рекурсии;

Уметь:

использовать конструкции функции и рекурсии, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования простейших

Здравствуйте, ребята. Меня зовут Коновалова Анжелика Денисовна. А мы продолжаем с вами проходить курс «Подготовка к ЕГЭ по информатике с использованием языка программирования Python».

На прошлых уроках мы с вами разобрали теорию для задания 8, а также попробовали решить несколько его примеров. Кроме этого, вы должны были самостоятельно выполнить домашнее задание и получить за это баллы.

Сегодня мы переходим к разбору задания 11. На выполнение данного задания отводится 5 минуты, за правильный ответ ученик получает 1 балл.

Для начала повторим теорию, нужную для выполнения предложенных задач.

Для решения данного задания нам надо знать следующее:

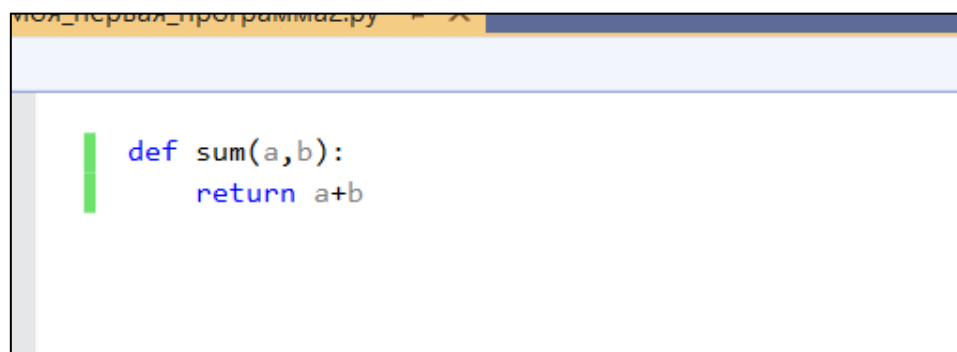
1. создание функции;
2. ключевое слово `return`;
3. условный оператор `if`;
4. рекурсия.

Для начала вспомним, что же такое функция? Функция – это фрагмент именованного программного кода (подпрограмма), к которому можно обратиться из другого места программы.

Существуют некоторые правила для создания **функций в Python**:

1. Блок функции начинается с ключевого слова `def`, после которого следуют название функции и круглые скобки «()».
2. Любые аргументы, которые принимает функция должны находиться внутри этих скобок.
3. После скобок идет двоеточие «:» и с новой строки с отступом начинается тело функции.

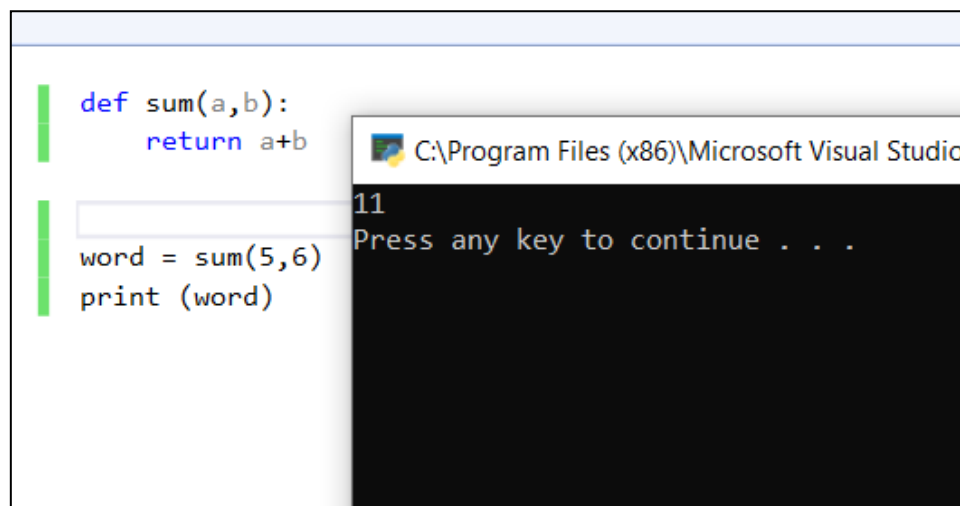
Например, функция может выглядеть следующим образом (рисунок А.1):



```
def sum(a,b):  
    return a+b
```

Рисунок А.1 – Функция

Для вызова функции следует ввести ее имя и добавить скобки (рисунок А.2). Например:

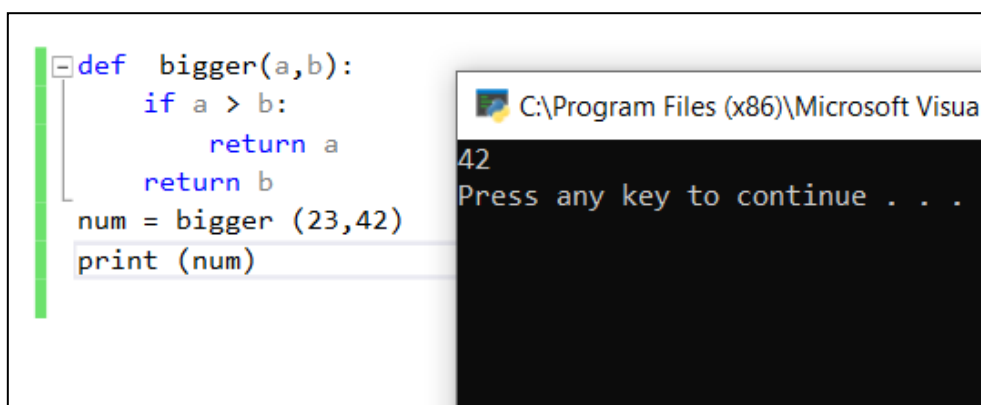


```
def sum(a,b):  
    return a+b  
  
word = sum(5,6)  
print (word)
```

C:\Program Files (x86)\Microsoft Visual Studio\...
11
Press any key to continue . . .

Рисунок А.2 – Вызов функции

Выражение **return** прекращает выполнение функции и возвращает указанное после выражения значение (рисунок А.3). Например:



```
def bigger(a,b):  
    if a > b:  
        return a  
    return b  
num = bigger (23,42)  
print (num)
```

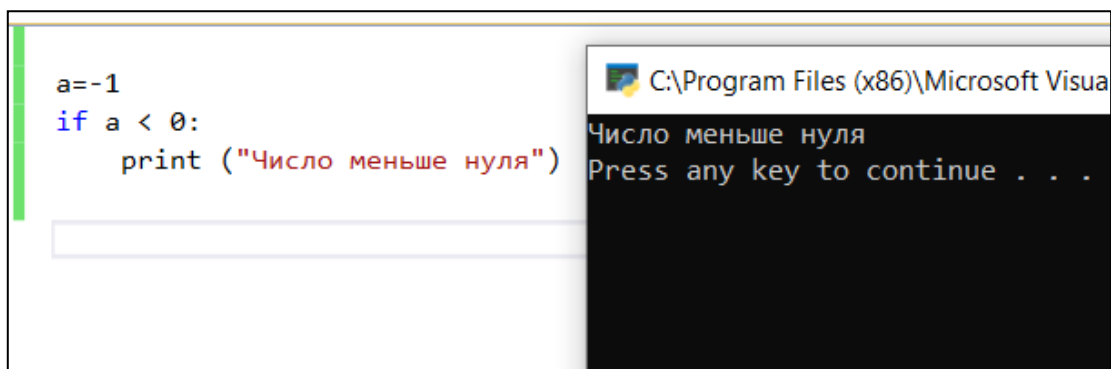
C:\Program Files (x86)\Microsoft Visual...
42
Press any key to continue . . .

Рисунок А.3 – Оператор Return

Условие данной программы состоит в том: если *a* больше чем *b*, то возвращается *b* и прекращает выполняться функция, в конце переменной *num* присваивается результат функции *bigger*.

Теперь рассмотрим условный оператор *if*. Думаю, многие из вас уже знакомы с ним.

Условный оператор *if* содержит в себе логическое условие, и, если это условие истинно, выполнится блок кода, записанный внутри команды *if*. Если логическое условие ложно, то блок кода, записанный внутри команды *if* пропускается, а выполнение кода переходит на следующую после блока *if* строчку кода (рисунок А.4). Например:



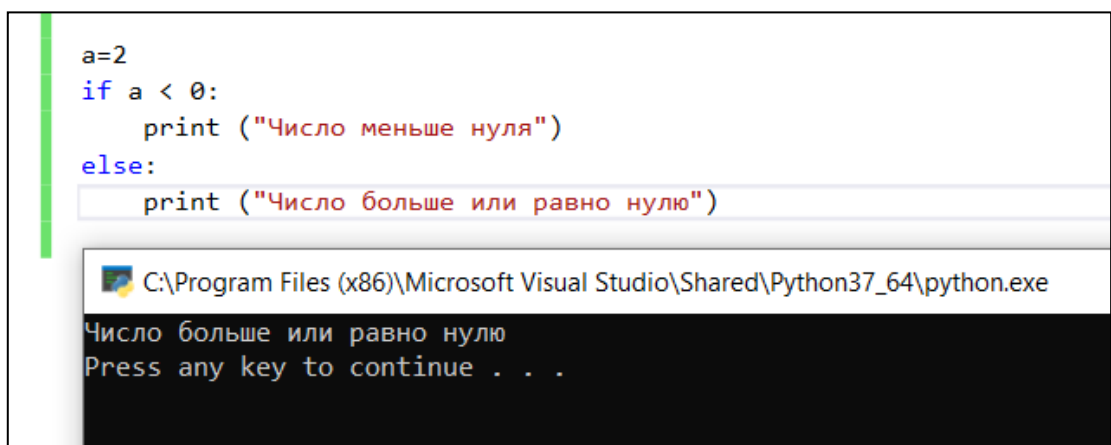
```
a=-1
if a < 0:
    print ("Число меньше нуля")
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe
Число меньше нуля
Press any key to continue . . .
```

Рисунок А.4 – Оператор If

В данной программе задается условие: если «а» меньше нуля, на экран выводится фраза «Число меньше нуля».

В конструкцию **if** может быть добавлена команда **else** (рисунок А.5). Она содержит блок кода, который выполняется, если условие в команде **if** ложно. Например:



```
a=2
if a < 0:
    print ("Число меньше нуля")
else:
    print ("Число больше или равно нулю")
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe
Число больше или равно нулю
Press any key to continue . . .
```

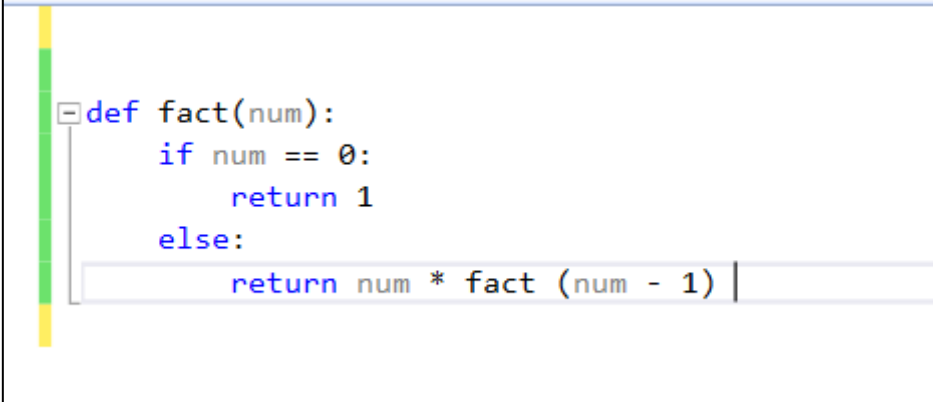
Рисунок А.5 – Конструкция If-else

Теперь условие звучит следующим образом: если «а» меньше нуля, на экран выводится фраза «Число меньше нуля», иначе на экран выводится фраза «Число больше или равно нулю».

И все же в задании 11 на ЕГЭ в первую очередь проверяется знание учащимися приемов использования рекурсивных алгоритмов. Что такое рекурсия?

Рекурсией в программировании называется ситуация, в которой функция вызывает саму себя. Классическим примером рекурсии может послужить функция вычисления факториала числа.

Факториалом числа, например, 4 является произведение всех натуральных (целых) чисел от 1 до 4. То есть, $1 * 2 * 3 * 4$. **Рекурсивная функция вычисления факториала на языке Python будет выглядеть так (рисунок А.6):**

A screenshot of a code editor showing a Python function definition for a recursive factorial. The code is as follows:

```
def fact(num):  
    if num == 0:  
        return 1  
    else:  
        return num * fact (num - 1) |
```

The code is displayed in a monospaced font with syntax highlighting: 'def' is blue, 'if' and 'else:' are blue, 'return' is blue, and the function name 'fact' is blue. The cursor is at the end of the last line. The editor has a vertical scrollbar on the left with a green and yellow bar.

Рисунок А.6 – Рекурсивная функция

Факториал нуля равен единице. В конце программы результат произведения `num` и результата, возвращенного функцией `fact (num - 1)`, возвращается.

Таким образом, мы кратко рассмотрели теорию для задания 11. Перейдите к просмотру следующего видео, где мы попробуем решить несколько примеров.

Приложение Б

Пример конспекта видеоурока с практическим материалом

Урок 3. Задание 11. Рекурсивные алгоритмы

Цель видеоурока – закрепить навыки составления рекурсивных алгоритмов и программ, реализующих циклические алгоритмические конструкции на языке программирования Python при решении типовых заданий из теста ЕГЭ.

Задачи:

1. Закрепить у учеников умение задавать выход из рекурсии.
2. Закрепить у обучаемых навык использования определения и записи на языке программирования рекуррентных формул.
3. Разобрать примеры типовых заданий из теста ЕГЭ.

Изучив данный видеоурок, ученик должен:

Знать:

понятие функции, оператора `if`, ключевого слова `return`, определение и конструкцию рекурсии;

Уметь:

использовать конструкции функции и рекурсии, строить простейшие алгоритмы и разрабатывать программы на языке Python для решения типовых заданий ЕГЭ;

Владеть:

методами и навыками разработки, отладки и тестирования простейших программ с использованием функции и рекурсии.

Продолжаем урок 3 «Задание 11. Рекурсивные алгоритмы».

В прошлом видеоролике мы разобрали теорию к данному заданию. Попробуем закрепить знания на практике, решим несколько примеров данной задачи.

Пример 1.

Дано следующее задание:

Ниже записаны две рекурсивные функции: F и G. Чему будет равно значение, вычисленное при выполнении вызова F (7)?

```
def F(n):  
    if n > 2:  
        return F(n-1) + G(n-2)  
    else: return 1  
def G(n):  
    if n > 2:  
        return G(n-1) + F(n-2)  
    else: return 1
```

Решение:

Разберем, какие действия выполняются в данном задании. Обратите внимание, что в примере используются две функции, следовательно, для каждая функция будет иметь свою блок-схему, а всего их будет три. Попробуем зарисовать блок-схемы (рисунок Б.1):

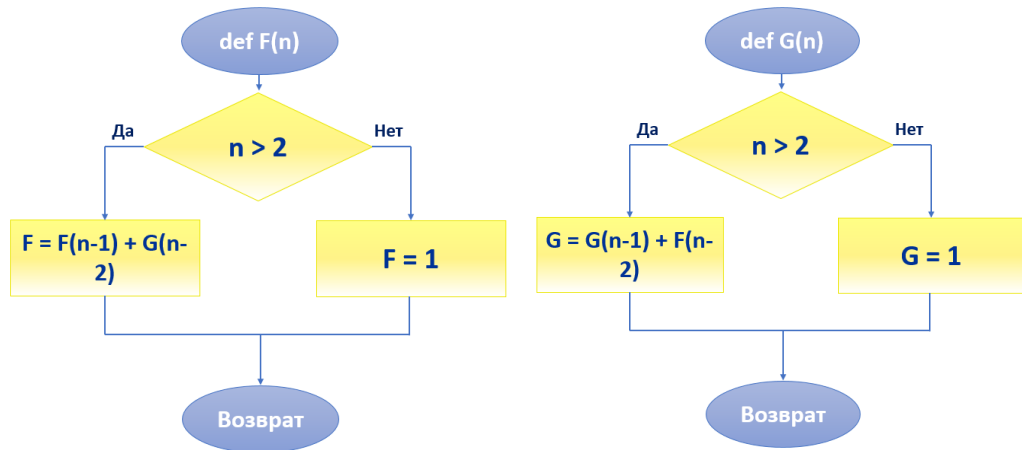


Рисунок Б.1 – Блок-схемы рекурсивных функций

Перед вами блок-схемы, где представлены тела функций. Но чтобы они начали выполняться, следует сделать вызов функции. Для это мы добавим в программу следующую строку: **print (F(7))**. Основная блок-схема будет выглядеть следующим образом (рисунок Б.2):

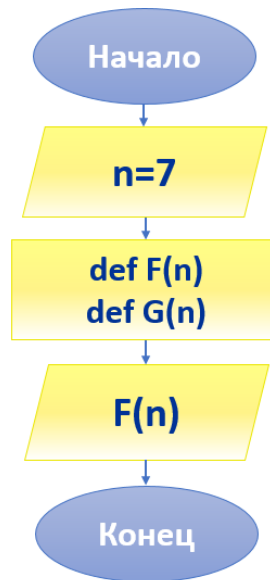


Рисунок Б.2 – Блок-схема программы

В программу вводится $n=7$. Давайте распишем, что выводит функция, начиная с 1:

$F(1) = 1;$
 $F(2) = 1;$
 $F(3) = G(1) + F(2) = 2;$
 $F(4) = G(2) + F(3) = 1 + 2 = 3;$
 $F(5) = F(4) + G(3) = 3 + 2 = 5;$
 $F(6) = F(5) + G(4) = 5 + 3 = 8;$
 $F(7) = F(6) + G(5) = 8 + 5 = 13.$

Ответ: 13.

Проверка: Сделаем проверку в Visual Studio (рисунок Б.3):

```

def F(n):
    if n > 2:
        return F(n-1) + G(n-2)
    else: return 1
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 1
print(F(7))
  
```

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python
 13
 Press any key to continue . . .

Рисунок Б.3 – Проверка решения

Пример 2.

Дано следующее задание:

Ниже записан рекурсивный алгоритм F. Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F (5)?

```
def F(n):  
    if n > 2:  
        return F(n-1) + F(n-2)  
    else: return n
```

Решение:

Обратите внимание, что в данной задаче используется только одна функция. Следовательно, блок-схем будет только 2 (рисунок Б.4):

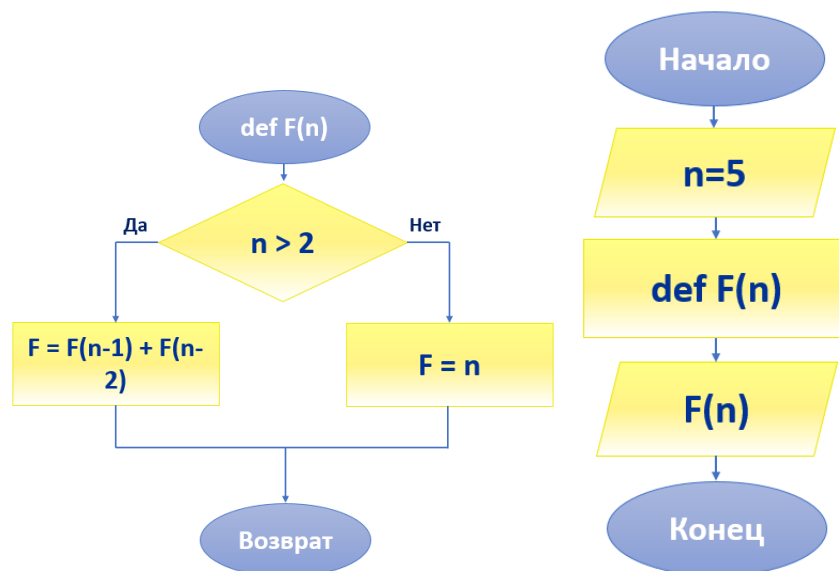


Рисунок Б.4 – Блок-схемы программы

Но чтобы функция начала выполняться, добавим вызов функции. Для это мы напишем в программе следующую строку: **print (F(5))**.

В программу вводится n=5. Значение, вычисленное алгоритмом при вызове F(5) равно:

$$F(5) = F(4) + F(3) = F(3) + F(2) + F(2) + F(1) = F(2) + F(1) + 2 + 2 + 1 = 8.$$

Ответ: 8.

Проверка: Сделаем проверку в Visual Studio (рисунок Б.5):

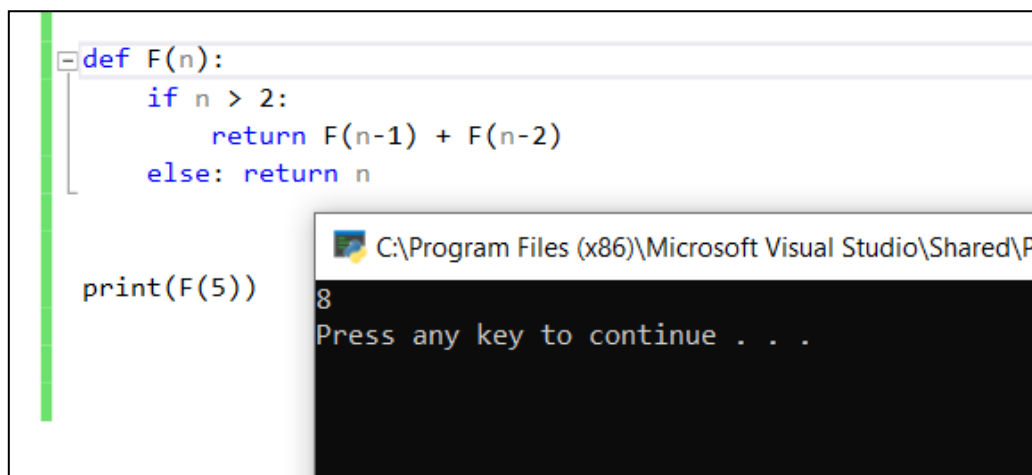
The image shows a code editor window with a Python script. The script defines a function F(n) that returns the sum of the two preceding Fibonacci numbers if n is greater than 2, and returns n otherwise. Below the function definition, there is a call to print(F(5)). An output window is overlaid on the bottom right, showing the number 8 and the prompt 'Press any key to continue . . .'. The path of the output window is 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\P'.

Рисунок Б.5 – Проверка решения

Вы с вами разобрали решения нескольких примеров задания 11. Попробуйте самостоятельно выполнить домашние задания и прислать решение для проверки.

На этом наш урок заканчивается. В следующем видео мы кратко повторим теорию к заданию 19 из ЕГЭ по информатике и попробуем прорешать некоторые его варианты.

Спасибо за внимание и увидимся на следующем уроке!

Приложение В

Пример домашнего задания

Самостоятельно выполните задания, приведенные ниже.

11.1. Ниже записаны две рекурсивные функции: F и G. Чему будет равно значение, вычисленное при выполнении вызова F (5)?

```
def F(n):
    if n > 2:
        return F(n-1) + G(n-1) + F(n-2)
    else: return n
def G(n):
    if n > 2:
        return G(n-1) + F(n-1) + G(n-2)
    else: return n+1
```

11.2. Ниже записаны две рекурсивные функции (процедуры): F и G. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F (11)?

```
def F(n):
    if n > 0:
        G (n - 1)
def G(n):
    print ("*")
    if n > 1:
        F (n - 3)
```

11.3. Ниже записан рекурсивный алгоритм F. Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F (6)?

```
def F(n):
    if n > 2:
        return F(n-1) + F(n-2) + F(n-3)
    else:
        return n
```

11.4. Ниже записана рекурсивная функция (процедура) F. Что выведет программа при вызове F(4)?

```
def F(n):
    print (n)
    if n > 2:
        F (n - 1)
```

$$F(n - 2)$$

$$F(n - 3)$$

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

$$F(1) = 1; F(2) = 1;$$

$$F(n) = F(n - 2) * (n - 1), \text{ при } n > 2.$$

Чему равно значение функции $F(7)$?

Приложение Г

Пример интерактивного упражнения

Задание «Блок-схема» (рисунок Г.1)

Ссылка: <https://learningapps.org/10403479>.

Формулировка задания: Постройте правильный порядок элементов блок-схемы.

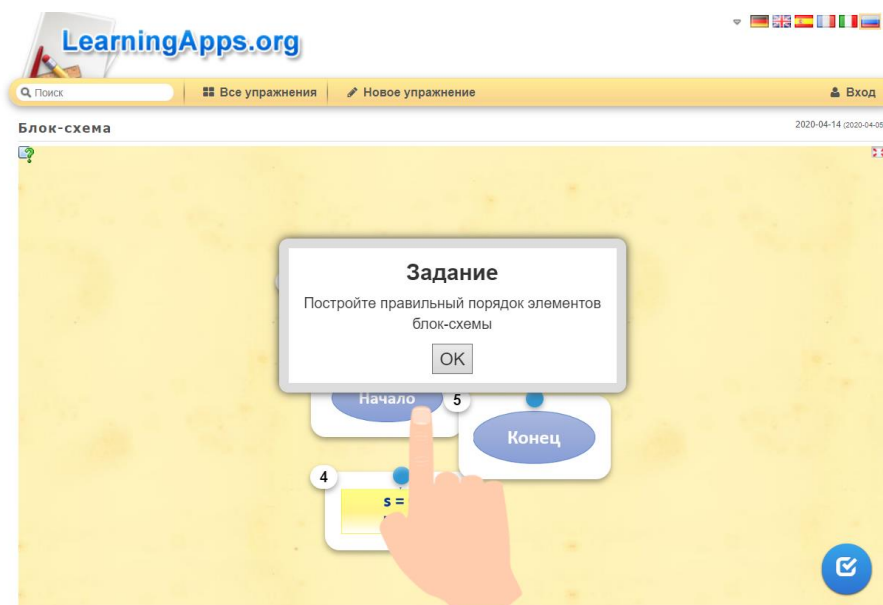


Рисунок Г.1 – Задание «Блок-схема»

Цель: закрепление умения составлять блок-схемы

Инструкция по выполнению: внимательно прочитайте задание и нажми «ОК» в диалоговом окне. Восстановите правильный порядок блоков. После нажмита на галочку в правом нижнем углу, чтобы проверить правильность выбранного ответа. Если весь столбец приобрел зеленый цвет, то вы выполнили задание верно, если столбец имеет красный цвет, то решение не верно.

Приложение Д

Типовые задания из бланка ЕГЭ по информатике (раздел «Алгоритмизация и программирование»)

Задание 8. Запишите число, которое будет напечатано в результате выполнения программы.

```
s = 42
n = 1
while s > 0:
    s = s - 5
    n = n + 3
print(n)
```

Задание 11. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

```
def F(n):
    if n > 2:
        return F(n-1)+ G(n-2)
    else: return 1
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 1
```

Чему будет равно значение, вычисленное при выполнении вызова F(7)?

Задание 19. Значения элементов двумерного массива A[1..100,1..100] задаются с помощью следующего фрагмента программы:

```
for i in range(1, 101):
    for k in range(1, 101):
        if i == k:
            A[i][k] = 1
        else:
            A[i][k] = -1
```

Чему равна сумма элементов массива после выполнения этого фрагмента программы?

Задание 20. Укажите наибольшее десятичное число, при вводе которого на экране сначала напечатается 3, а затем 6.

```
x = int(input())
L = 0
```

```

M = 0
while x > 0 :
    L = L+1
    if (x % 2) != 0:
        M = M + x % 8
    x = x // 8
print(L)
print(M)

```

Задание 21. Напишите в ответе наибольшее значение входной переменной k , при котором программа выдаёт тот же ответ, что и при входном значении $k = 49$.

```

def f(n):
    return n*n*n
def g(n):
    return 2*n + 1
k = int(input())
i = 1
while f(i) < g(k):
    i += 1
print (i)

```

Задание 24. На обработку поступает последовательность из четырёх целых чисел. Нужно написать программу, которая выводит на экран сумму всех отрицательных чисел последовательности и максимальное число в последовательности.

Известно, что вводимые числа по абсолютной величине не превышают 1000. Программист написал программу неправильно.

```

mx = 0
s = 0
for i in range(1, 5):
    x = int(input())
    if x < 0:
        s = x
    if x > mx:
        mx = x
print(s)
print(mx)

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности - 5 2 -4 3.

2. Приведите пример такой последовательности, содержащей хотя бы одно неотрицательное число, что, несмотря на ошибки, программа печатает правильный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Задание 25. Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите алгоритм, позволяющий найти и вывести максимальное значение среди двузначных элементов массива, не делящихся на 3. Если в исходном массиве нет элемента, значение которого является двузначным числом и при этом не кратно трём, то выведите сообщение «Не найдено».

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

```
# допускается также
# использовать две
# целочисленные переменные j, max
a = []
```

```
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Free Pascal 2.4) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Задание 27. Дан набор из N целых положительных чисел. Из этих чисел формируются все возможные пары (парой считаются два элемента, которые находятся на разных местах в наборе, порядок чисел в паре не учитывается), в каждой паре вычисляются сумма и произведение элементов. Необходимо определить количество пар, у которых сумма нечётна, а произведение делится на 3.

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел N в k раз время работы программы увеличивается не более чем в k раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает одного килобайта и не увеличивается с ростом N .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, — 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, — 2 балла.

Вы можете сдать одну или две программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет бóльшая из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию

Описание входных и выходных данных.

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N строк записано одно натуральное число, не превышающее 100.

Пример входных данных:

4
1
2
3
4

Пример выходных данных для приведённого выше примера входных данных:

2

Из четырёх чисел можно составить 6 пар. В данном случае условиям удовлетворяют две пары: (2, 3) и (3, 4). Суммы чисел в этих парах (5 и 7) нечётны, а произведения (6 и 12) делятся на 3. У всех остальных пар как минимум одно из этих условий не выполняется.

Приложение Е

Тест с использованием авторской методики Полякова К.Ю.

1. Что будет выведено на экран в результате выполнения фрагмента программы

```
a = 5
b = 3
print ( "a=Z(", b, ")", sep="" )
```

2. Запишите оператор для вывода значений переменных a=5 и b=3 в следующем формате:

5 + 3 =?

3. Что будет выведено на экран в результате выполнения фрагмента программы

```
a = 5
b = 3
print ( "Z(", a, ")=(", a+b, ")", sep="" )
```

4. Чему будет равна переменная «с» после выполнения этой программы:

```
a = 20
b = 3
b = a // b + b
c = a % b + a
```

5. Чему будет равна переменная «с» после выполнения этой программы:

```
a = 26
b = 6
b = a // b
c = a % (b + 1)
```

6. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
if a > 5:
    a = 12
```

7. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10;
if a > 5: a += 12
else: a -= 7
if a > 5: a += 12
else: a -= 7
```

8. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
b = 5
if not (a < 5):
    a = 5
```

9. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
b = 5
if a > 1 and a < b:
    a -= 5
if a > 1 and a == b:
    a -= 5
```

10. Какое число нужно написать вместо многоточия, чтобы цикл выполнялся ровно 3 раза?

```
i = 2
while i <= ...:
    print ( "Привет!" )
    i += 1
```

11. Чему будет равно значение переменной «а» после выполнения этого цикла?

```
i = 4
a = 12
while i < 5:
    a += i
    i += 1
```

12. Чему будет равно значение целой переменной «а» после выполнения этого фрагмента программы?

```
a = 10;
for i in range(2):
    a -= 1
```

13. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
print ( 4, end="" )
for i in range(2,1,-1):
    print ( 2*i, end="" )
```

14. Отметьте все правильные утверждения о списках (массивах) в языке Python.

- элементы списка могут быть разных типов
- все элементы списка должны быть одного типа
- элементы могут нумероваться с единицы
- элементы всегда нумеруются с нуля
- размер списка может меняться во время работы программы

15. Какой индекс имеет последний элемент массива A?

```
A = [8]*6
```

16. Требуется заполнить массив именно так:

```
X = [1 3 5 7 9 11]
```

Какой оператор надо поместить в тело цикла вместо многоточия?

```
X = [0]*6
for k in range(6):
```

...

- X[k] = k
- X[k] = 2*k
- X[k] = 2*k - 1
- X[k] = 2*k + 1
- X[k] = 2*(k + 1)

17. Задан массив X из N элементов. Какой оператор надо поставить вместо многоточия, чтобы найти сумму всех элементов массива в переменной S?

```
S = 0
for k in range(N):
    ...
```


18. Задан массив X из N элементов. Какой оператор надо поставить вместо многоточия, чтобы найти номер максимального элемента массива в переменной M?

```
M = 0
for k in range(1,N):
    if X[k] > X[M]:
        ...
```

19. Что будет выведено на экран после выполнения программы?

```
s = "123"
s = s + "0" + s;
s = s + s;
print ( s )
```

20. Что будет выведено на экран после выполнения этой программы?

```
s = "12345"
n = len(s) + len("456")
print ( n )
```

Приложение Ж

Анкета

Ваше имя: _____

1. Как Вы в целом оцениваете проведенный тренинг?

- Очень хорошо
- Хорошо
- Удовлетворительно
- Неудовлетворительно
- Затрудняюсь ответить

2. Был ли Вам понятен предоставленный материал?

- Да
- Нет

3. Было ли Вам интересно принимать участие в обучении?

- Да
- Нет

4. Был ли предложенный материал полезен для Вас?

- Да
- Нет

5. Что бы Вы посоветовали улучшить?

Благодарю за заполнение анкеты!