

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Алтайский государственный гуманитарно-педагогический  
университет имени В.М. Шукшина»  
(АГГПУ им. В.М. Шукшина)

Институт естественных наук и профессионального образования  
Кафедра математики, физики, информатики

Направление подготовки 44.03.01 Педагогическое образование  
Профиль подготовки Информатика

**Методы обучения учащихся средней школы элементам алгоритмизации  
и программирования на уроках информатики**

Выпускная квалификационная работа

**Допустить к защите**

Зав. кафедрой математики,  
физики, информатики

«\_\_» \_\_\_\_\_ 20\_\_ г.

Захаров П.В.

\_\_\_\_\_  
(подпись)

Выполнил студент

Группы Ф-ЗИ141

Соколов

(фамилия)

Александр Николаевич

(имя, отчество)

\_\_\_\_\_  
(подпись)

Научный руководитель

к. физ-мат. н.

(ученая степень, звание)

Шилинг Г.С.

(Ф.И.О.)

\_\_\_\_\_  
(подпись)

**Оценка**

«\_\_» \_\_\_\_\_ 20\_\_ г.

\_\_\_\_\_  
(подпись председателя ГЭК)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Алтайский государственный гуманитарно-педагогический  
университет имени В.М. Шукшина»  
(АГГПУ им. В.М. Шукшина)

АННОТАЦИЯ

на выпускную квалификационную работу бакалавра

студента Соколова Александра Николаевича группы Ф-ЗИ141

Направление 44.03.01 Педагогическое образование

Профиль (при наличии) Информатика

Тема Методы обучения учащихся средней школы элементам алгоритмизации и программирования на уроках информатики

The object of research is the methods of teaching students in secondary schools

Subject of research: the process of teaching high school students the elements of algorithmization and programming in computer science lessons using the developed methodological recommendations.

Thesis consists of an introduction, two chapters and a conclusion.

The first chapter examines teaching methods, guidelines, and the place in the curriculum of the topic “algorithmization and programming”.

In the second chapter, a methodological recommendation is developed for teaching high school students the elements of algorithmization and programming. Testing is conducted in the form of a pedagogical experiment.

Thesis includes applications with lesson plans and electronic media. The list of sources used consists of 44 items.

Автор ВКР \_\_\_\_\_ Соколов А.Н.  
(подпись) (ФИО)

Руководитель ВКР \_\_\_\_\_ Шилинг Г.С.  
(подпись) (ФИО)

## Оглавление

Введение.....	4
Глава 1. Методика преподавания раздела «Алгоритмизация» в школьном курсе информатики .....	7
1.1 Место предмета в учебном плане.....	7
1.2 Методика обучения «Программированию» и введение понятия «алгоритм».....	8
1.3 Классификация методов обучения.....	19
Вывод по Главе 1.....	29
Глава 2. Разработка и проведение занятий с применением метода видеоурока.....	30
2.1 Роль видеоурока в образовании.....	30
2.2 Каким должен быть видеоурок.....	32
2.3 Программное обеспечение, используемое для создания видеоурока.....	34
2.4 Создание видеоурока на примере CamStudio.....	37
2.5 Разработка уроков и апробация.....	39
Вывод по Главе 2.....	48
Заключение.....	49
Библиографический список.....	50
Приложение 1 Диаграмма результатов исследования.....	51
Приложение 2 Тест для итоговой проверки знаний.....	52
Приложение 3 План-конспект урока "Алгоритмизация и язык программирования Паскаль" 9 класс .....	58

## Введение

Чтобы решать задачи на компьютере, необходимо уметь составлять алгоритм. Решать задачи и составлять план их решения, ставить и доказывать гипотезы опытным путем, определять итоги своей деятельности, анализировать и искать лучшие способы решения задач, путем улучшения и конкретизации созданного алгоритма позволяют определять уровень развития алгоритмического мышления школьников. Следовательно нужно пристальное внимание уделять алгоритмическому мышлению растущему поколению.

Алгоритмическое мышление по ходу жизни растет под действием внешних обстоятельств и под действием дополнительных факторов возможен его более заметный рост. Потребность находить новейшие результативные средства формирования алгоритмического мышления у подростков обусловлена его важностью с целью последующей самореализации личности в современном компьютеризированном мире.

В методической литературе по информатике замечены разнообразные методы развития алгоритмического мышления учеников: осуществление регулярного и направленного использования идей структурного подхода, улучшение уровня мотивации задач, непрерывная интеллектуальная деятельность.

Результативным методом формирования алгоритмического мышления учеников старших классов при изучении темы «алгоритмизация и программирование» считается обучение составлению алгоритмов и их применение при решении задач.

Касательно изучения базового курса дисциплины в 7-9 классах, программирование в 9 классе начало преподаваться в уменьшенном объеме, скатившись до изучения одного языка программирования, однако ключевые конструкции языка были популярны в иных разделах информатики (устройство ПК, логика, электронные таблицы). При беседе с учениками об освоении раздела «Алгоритмизация и программирование» замечено

интересное мнение. Многие думают, что информатику возможно осваивать минуя раздел «Алгоритмизация и программирование», несмотря на то важность темы никто никак не опровергал.

Актуальность: В процессе всего этапа обучения информатики в школе значимость раздела «Алгоритмизация и программирование» претерпела существенные перемены. Согласно определенным причинам: увеличением теоретической базы дисциплины и технического снабжение кабинета информатики, обучение разделу в промежуток с 2003 года по 2018 год существенно уменьшилось. Точнее сказать, снизилось число уроков, отводимых на обучение данной темы в старших классах. Значительная доля времени отводится на обучение тем из раздела «Информационные и коммуникационные технологии». Однако ни смотря на это абсолютно никак не поменялись требования к степени освоения знаний и умений данного раздела программы по информатике, так как он остается фундаментом базовых знаний в этой дисциплине. Поэтому важно правильно подходить к методу обучения данному разделу, чтобы эффективно с меньшими времензатратами преподносить новый материал по данной теме.

Объектом исследования являются методы обучения учащихся средней школы.

Предмет исследования: процесс обучения учащихся средней школы элементам алгоритмизации и программирования на уроках информатики с применением разработанной методической рекомендации.

Целью работы является исследование методов обучения учеников средней школы и разработка методической рекомендации для обучения элементам алгоритмизации и программирования на уроках информатики.

- Задачи: - изучить методическую литературу;
- выделить методические рекомендации по изучении данной темы;
  - рассмотреть основные методы преподавания;
  - выделить современный, предположительно перспективный метод и исследовать его роль в образовании;
  - применить его на практике, провести педагогический эксперимент;

- подвести итоги апробации.

Дипломная работа состоит из введения, двух глав и заключения.

В первой главе проводится исследование методик преподавания, методических рекомендаций и место в учебном плане темы «алгоритмизация и программирование».

Во второй главе разрабатывается методическая рекомендация для обучения учащихся средней школы элементам алгоритмизации и программированию. Проводится апробация в виде педагогического эксперимента.

Дипломная работа включает в себя приложения с план-конспектами уроков и электронный носитель. Список использованных источников состоит из 44 наименований.

Методы исследования: анализ теоретических источников и публикаций по теме, обобщение и интерпретация результатов исследования, формулирование методических рекомендаций, разработка уроков по изученной теме, проведение эксперимента с проведением уроков и классного часа в школе.

Апробация проводилась в краевом государственном бюджетном профессиональном образовательном учреждении «Бийском медицинском колледже».

## **Глава I. Теория и методика преподавания раздела «Алгоритмизация» в школьном курсе информатики**

### **1.1 Место предмета в учебном плане**

Федеральный базисный учебный план для образовательных учреждений Российской Федерации отводит 105 часов для обязательного изучения информатики и информационных технологий на ступени основного общего образования. В том числе в VIII классе – 35 учебных часов из расчета 1 учебный час в неделю и IX классе – 70 учебных часов из расчета 2 учебных часа в неделю. В примерной программе предусмотрен резерв свободного учебного времени в объеме 11 часов (10,5%) для реализации авторских подходов, использования разнообразных форм организации учебного процесса, внедрения современных методов обучения и педагогических технологий, учета региональных условий.

Распределение содержания по годам обучения может быть вариативным, более того, оно может частично осваиваться уже в начальной школе за счет использования компонента образовательного учреждения и регионального компонентов учебного плана (первоначальное знакомство учащихся с информационными технологиями должно пройти в курсах «Окружающий мир» и «Технология» начальной школы). Содержание образовательной области «Информатика и информационно-коммуникационные технологии» осваивается как в рамках отдельного школьного предмета с таким названием, так и в межпредметной проектной деятельности. Не допускается деление предмета на два («Информатику» и «Информационные технологии») при заполнении журналов и аттестационных документов.

Из всего курса на «Алгоритмы и исполнители» выделяется всего 19 часов.

## **1.2 Методика введения понятия «алгоритм» и обучения «программированию»**

Технология обучения информатики устанавливает перед собою следующие цели: установить определенные цели исследования информатики, а кроме того суть надлежащего общеобразовательного предмета и его роль в учебном плане средней школы; создать и порекомендовать школе и педагогу-практику более оптимальные способы и организационные формы преподавания, устремленные на результат установленных целей; проанализировать целый комплекс средств преподавания информатике (учебные пособия, программные ресурсы, тех. ресурсы и т.п.) и выработать советы согласно их использованию в практике деятельности педагога. Основная особенность курса методики преподавания информатики (МПИ) – связь с другими предметами.

Н.В. Софронова пишет, что «преподавание информатики на современном уровне опирается на сведения из различных областей научного знания: биологии (биологические самоуправляемые системы, такие как человек, другой живой организм), истории и обществоведения (общественные социальные системы), русского языка (грамматика, синтаксис, семантика и пр.), логики (мышление, формальные операции, истина, ложь), математики (числа, переменные, функции, множества, знаки, действия), психологии (восприятие, мышление, коммуникации)». [39].

Иной характерной чертой МПИ считается динамичный, меняющийся вид самой информатики и как науки, и как учебного предмета, её непостоянность, непрерывный рост и усовершенствование равно как технических, так и в особенности программных средств. В данных обстоятельствах вынужденным и продуктивным выводом считается наибольший упор на результаты общей дидактики, на определенные методики ближайших дисциплин – математики и физики.

Ещё одна отличительная черта МПИ – взаимосвязь предмета с применением ПК, что имеет значительно большую “самостоятельность”, нежели всякое иное устройство.



Исследование алгоритмизации и программирования разделяется на две стадии, это и является само исследование алгоритмизации, а далее программирования. В многочисленных учебных программах делают упор лишь в освоении алгоритмизации, потому как незначительное число преподавателей информатики обладают надлежащим уровнем подготовки для обучения программирования на том или ином определенном языке программирования.

Обучение алгоритмизации может помочь сформировать у обучающихся алгоритмическое мышление, которое само по себе считается основой для изучения программирования. По этой причине исследование алгоритмизации считается значимой составляющей направления информатики и при преподавании данного курса педагог обязан быть в особенности осмотрителен.

В образовательном стандарте базового курса по информатики и ИКТ выделяют следующую цепочку понятий для изучения:

- алгоритм, свойства алгоритма, способы записи алгоритмов;
- исполнители алгоритмов (назначение, среда, режим работы, система команд);
- компьютер как формальный исполнитель алгоритмов;
- основные алгоритмические конструкции (следование, ветвление, повторение);
- разбиение задачи на подзадачи, вспомогательный алгоритм;
- алгоритмы работы с величинами (тип данных, ввод и вывод данных).

Изучение алгоритмизации начинается с введения понятия алгоритма. Понятие «алгоритм» относится к фундаментальным математическим понятиям, следовательно, его не охарактеризовать более простыми понятиями. Поэтому определение алгоритма в школьных учебниках по информатике отличается большим многообразием. Вот пример некоторых:

В учебнике Семакина И.Г.( *Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.*, Информатика: Учеб. по базовому курсу и др. - М.: Лаборатория Базовых Знаний, 1998.) [37] «алгоритм» определяется как

последовательность команд, управляющих работой какого-либо объекта, и далее дается более строгое определение – понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую от исходных данных к искомому результату.

В учебнике Кушниренко А.Г. (*Кушниренко А.Г., Лебедев Г.В., Сворень Р.А.*. Основы информатики и вычислительной техники: Пробный учеб. для сред, учеб. заведений - М.: Просвещение, 1993.) [19] «алгоритм» определяется как программа, записанная на специальном школьном алгоритмическом языке.

В учебнике Угриновича Н.Д. [40] (Угринович Н.Д. Информатика и информационные технологии. Учебник для 9 класса.– М.: Бинوم, 2009.) «алгоритм» вводится как чёткое описание последовательности действий.

В повседневной жизни дети никак не сталкиваются с данными понятиями буквально, однако они обнаруживают использование алгоритмов в разной работе человека, о чем немаловажно проинформировать детей на первом же занятии и доказать данное примерами. Вводя представление алгоритма, педагогу необходимо заострять интерес обучающихся на этом, то что алгоритм постоянно составляется с ориентацией в исполнителя алгоритма. По моему самое подходящее определение дано в учебнике И.Г. Семакина.

Таким образом одной с отличительных черт направления «Алгоритмизация и программирование» считается его практическая направленность, в таком случае понятие исполнителя алгоритма необходимо внедрять на базе практических примеров из жизни обучающихся. Главным исполнителем в первоначальном моменте обучения темы обязан являться человек. Учащиеся сами обязаны выступить в образе исполнителей не трудных алгоритмов, к примеру изображение окружности при помощи циркуля. В более старших классах задачи можно ставить и сложнее, в плоть до извлечения корня из квадратного уравнения.

Главной чертой исполнителя, с точки зрения управления, считается система команд исполнителя (СКИ). Данное конечное множество команд,

которые понимает исполнитель, т.е. может их осуществлять. С целью ознакомления с СКИ допускается дать учащимся такого рода алгоритм, который они заведомо никак не сумеют исполнить. Уже после этого следует закрепление этого понятия на базе задач нахождения СКИ у разных исполнителей.

СКИ определяет первое свойство алгоритма – понятность, то есть алгоритм должен включать в себя только команды, входящие в СКИ. Он не должен быть рассчитан на принятие исполнителем самостоятельных решений, не предусмотренных составителем алгоритма.

После свойства понятности следует свойство точность. Можно привести несколько примеров алгоритмов, которые выполняются не точно. Например: кулинарный рецепт можно рассматривать как алгоритм для исполнителя-повара по приготовлению блюда. Но если одним из пунктов в нем будет написано: «Положить несколько ложек сахара», то это пример неточной команды. Сколько ложек? Каких ложек (чайных, столовых)? Каждый повар может это понимать по-своему, и результаты будут разными. Пример точной команды: «Положить 2 столовые ложки сахара».

Еще одно свойство, которое отражено в определении алгоритма – конечность. Оно формулируется так: исполнение алгоритма и, следовательно, получение искомого результата должно завершиться за конечное число шагов. Здесь под шагом подразумевается выполнение отдельной команды. В данном случае это свойство отражает ситуации, когда алгоритм «зацикливается» и не дает результата. Такой алгоритм бесполезен, учащиеся должны научиться отличать эти алгоритмы.

Далее идет свойство алгоритма - дискретность. «Дискретность состоит в том, что команды алгоритма выполняются последовательно, с точной фиксацией моментов окончания выполнения одной команды и начала выполнения следующей». Требование последовательного выполнения команд заложено в определении алгоритма, но, на мой взгляд, на данном свойстве нужно заострить внимание. Не каждый ребенок сможет выделить его из определения алгоритма.

«Свойство массовости выражается в том, что алгоритм единым образом применяется к любой конкретной формулировке задачи, для решения которой он разработан». От свойства массовости легко перейти к такому понятию как исходные данные. По сути, это свойство можно назвать универсальностью алгоритма по отношению к исходным данным решаемой задачи. Данное свойство не является необходимым свойством алгоритма, а скорее определяет качество алгоритма: универсальный алгоритм лучше не универсального (алгоритм решения частной задачи – тоже алгоритм!). Следует указать учащимся на то, что исполнителю всегда необходимо иметь исходные данные с которыми он будет работать (деньги, продукты, детали, таблицы чисел и т.п.). Например, исполнителю, решающему математическую задачу нужна исходная числовая информация, которая обычно задаётся в условии. Если вам нужно найти номер телефона нужного человека, то исходными данными будут фамилия человека, его инициалы, телефонная книга, а иногда ещё и домашний адрес, так как Ивановых или Петровых с одинаковыми инициалами может оказаться в телефонной книге несколько.

В случае если все без исключения данные свойства выполняются, то исполнитель осуществляет алгоритм формально. Это значит, то что при выполнении алгоритма исполнитель точно следует указаниям и никаких произвольных действий он сделать не способен. Отсюда вытекает вывод о способности формирования автоматических исполнителей. Подобным автоматическим исполнителем по обрабатыванию данных считается эвм. Ученики сами могут назвать подобные автоматические исполнители: роботы, станки с автоматическим управлением, автоматическая стиральная машина и другие.

После того как все свойства алгоритма разобраны следует их закрепить при помощи задач. Для этого полезно рассмотреть с учениками несколько заданий следующего содержания:

- 1) выполнить роль исполнителя: дан алгоритм, формально исполнить его;
- 2) определить исполнителя и систему команд для данного вида работы;

- 3) в рамках данной системы команд построить алгоритм;
- 4) определить необходимый набор исходных данных для решения задачи.

В качестве примера задачи первого типа можно использовать алгоритм игры Баше, рассматриваемый в учебниках. Правила игры определены так: в игре используются 8, 12, 16, 20 предметов. За один ход можно брать от 1 до 4 предметов. Проигрывает тот, кто берет последний предмет. Предлагается алгоритм выигрыша для первого игрока. После того как ученики поиграли в эту игру по тем правилам, что описаны в учебнике, можно предложить им несколько заданий аналитического характера на тему игры Баше. Задания могут быть предложены в качестве домашней работы.

Теперь рассмотрим пример задания второго типа.

Задача: Описать систему команд исполнителя «Геометр», который мог бы выполнять геометрические построения с помощью циркуля и линейки.

Решение. Ученикам знаком класс задач, которые в геометрии называются задачами на построение с помощью линейки, циркуля и карандаша. Полной системой команд для исполнителя «Геометр» является следующий список:

1. Провести отрезок прямой между двумя данными точками.
2. Установить раствор циркуля, равный длине данного отрезка.
3. Установить ножку циркуля в данную точку.
4. Провести окружность.
5. Выделить общие точки двух линий (пересечения или касания).

Необходимо обратить внимание учеников на элементарность каждой команды. Делить их на более простые не имеет смысла.

При построении СКИ ученики должны решать две проблемы: проблема элементарности команд и проблема полноты системы команд. Система команд исполнителя называется полной, если она содержит весь минимально-необходимый набор команд, позволяющий построить любой алгоритм в том классе задач, на который ориентирован исполнитель.

Отрешения предыдущей задачи можно перейти к задачам третьего типа. Оставив исполнитель и СКИ прежними ученикам можно дать такую задачу: «Записать для исполнителя Геометр алгоритм построения окружности, для которой задан её диаметр отрезком АВ».

Данный переход способствует лучшему восприятию задачи, так как ученики уже знакомы с исполнителем и его СКИ.

- Решение:
- установить ножку циркуля в т. А;
  - установить раствор циркуля, равный АВ;
  - провести окружность установить ножку циркуля в т. В;
  - провести окружность;
  - выделить точки пересечения окружностей: т. Сит. D;
  - провести отрезок CD;
  - выделить точку пересечения АВ и CD: т. О;
  - установить ножку циркуля в т. О;
  - установить раствор циркуля, равный ОВ;
  - провести окружность.

С учениками необходимо проанализировать данную задачу на соответствие свойствам алгоритма. Учеников следует подвести к такому выводу: «данный алгоритм удовлетворяет всем основным свойствам: понятности, точности, конечности; благодаря чему может исполняться формально».

Четвертый тип заданий относится к проблеме постановки задач на построение алгоритмов. Для выполнения требуемой работы – решения данной задачи – необходим не только алгоритм, но и полный набор исходных данных. Это могут быть какие-то материальные объекты (например, детали для сборки устройства; продукты для приготовления блюда и пр.) или информация (числовые данные для расчетов). Вот пример задачи на определение полного набора данных.

Задача: Определить полный набор данных для вычисления времени падения кирпича с крыши дома.

Ответ: высота дома, ускорение свободного падения.

*Методические рекомендации при обучении построению алгоритма*

Главной целью раздела алгоритмизации является овладение учащимися структурной методикой построения алгоритмов. Традиционно применяемым дидактическим средством в этом разделе являются учебные исполнители алгоритмов. Главным достоинством учебных исполнителей является: ясность для ученика решаемых задач, наглядность процесса работы в ходе выполнения программы. Как известно, дидактический принцип наглядности является одним из важнейших в процессе любого обучения.

Для того чтобы ученикам было легко работать с учебными исполнителями, они должны удовлетворять следующим условиям:

- это должен быть исполнитель, работающий «в обстановке»;
- этот исполнитель должен имитировать процесс управления некоторым реальным объектом (черепахой, роботом и др.);
- в системе команд исполнителя должны быть все структурные команды управления (ветвления, циклы);
- исполнитель позволяет использовать вспомогательные алгоритмы (процедуры).

Изучая работу любого исполнителя алгоритмов, учителю следует привести его характеристики, совокупность которых называется архитектурой исполнителя. К ним относятся:

- среда, в которой работает исполнитель;
- режим работы исполнителя;
- система команд исполнителя;
- данные, с которыми работает исполнитель.

Обучение программированию лучше организовать в ходе решения задач, подобранных в специально выстроенной последовательности, которая определяется следующими дидактическими принципами:

От простого к сложному – т.е. постепенное усложнение решаемых задач.

-Новизна – каждая задача должна вносить новый элемент знаний – новую команду, новый приём программирования.

-Наследование – решение каждой следующей задачи требует использования знаний, полученных при решении предыдущих.

Чтобы написать для «исполнителя» «алгоритм» используется алгоритмический язык и блок-схемы. С ними можно познакомить на одном уроке, а затем продолжать изучение алгоритмизации и блок-схем совместно с построением алгоритмов на учебных исполнителях. Это поможет изучить основные алгоритмические структуры с теоретической и практической стороны.

Основное достоинство блок-схем – наглядность представления структуры алгоритма. Это достигается изображением блок-схем стандартным способом – сверху вниз.

Алгоритмический язык есть текстовая форма описания алгоритма, которая близка к языку программирования, но как таковым ещё не является, и поэтому не имеет строгого синтаксиса. Для структурирования текста алгоритма в алгоритмическом языке используются строчные отступы. При этом соблюдается правило: все конструкции одного уровня вложенности записываются на одном вертикальном уровне (отступе), а вложенные конструкции смещаются относительно внешней вправо. Это оптимизирует наглядность структуры алгоритма. Поэтому учителю желательно потратить определённое учебное время на формирование навыка правильной записи алгоритма.

После ознакомления с архитектурой исполнителя и способами записи алгоритмов следует приступить к решению задач, соответствующих приведенным выше дидактическим принципам. Только практическая работа на учебных исполнителях помогает освоить построение алгоритмов.

На практических занятиях используются следующие типы задач:

- составление простых линейных алгоритмов;
- составление и использование вспомогательных алгоритмов;
- составление циклических алгоритмов;
- использование ветвлений в алгоритмах;



- использование метода последовательной детализации при составлении сложных алгоритмов.

При разборе этой задачи необходимо обратить внимание учеников на два обстоятельства. Первое: управление учебным исполнителем для достижения поставленной цели будет происходить без обратной связи. В данном случае алгоритм управления будет иметь линейную структуру.

Второе: алгоритм зависит не только от сформулированной цели (искомого результата), но и от исходного состояния исполнителя. Если бы исходное состояние было другим, то был бы другим и алгоритм, несмотря на то, что в результате получается один и тот же рисунок. Для алгоритмов работы «в обстановке» начальное состояние исполнителя является исходным данным задачи. Состояние учебного исполнителя определяется местом его расположения на поле и ориентацией. Результатом же выполнения алгоритма становится не только рисунок (главная цель), но и конечное состояние исполнителя.

Умение использовать вспомогательные алгоритмы необходимо вырабатывать у учеников как можно раньше, уже на примерах линейных алгоритмов. Важнейший прием алгоритмизации и программирования – декомпозиция задачи, т.е. выделение в исходной задаче некоторых более простых подзадач. Алгоритмы решения таких подзадач называются вспомогательными алгоритмами, а реализующие их программы – подпрограммами (процедурами). Таким образом, решение исходной задачи разбивается на несколько алгоритмов: основной алгоритм и вспомогательные алгоритмы. Как правило, в основном алгоритме происходит многократное обращение к вспомогательному алгоритму.

Далее следует изучение циклов. Для составления циклических алгоритмов, следует сначала теоретически подготовить учащихся. Необходимо подробно разобрать циклические алгоритмы при помощи блок схем и алгоритмического языка. И только потом переходить на практику, иначе дети могут не усвоить циклы, и действовать по примерам, не думая о содержании задачи.

Примером задачи на циклы может служить задача на составление алгоритм рисования горизонтальной линии, проведенной от края до края поля. Такая задача привносит в данную тему следующие новые элементы: структурная команда цикла; управление с обратной связью. Обратная связь между объектом управления и управляющей системой заключается в том, что перед выполнением каждого шага проверяется условие «впереди не край?». Если оно истинно, т.е. ответ положительный, то делается шаг, в противном случае выполнение цикла прекращается.

Команда цикла является структурной командой в отличие от простых команд «шаг», «поворот», «прыжок». Структурная команда состоит из нескольких действий: проверка условия, выполнение тела цикла, которое, в свою очередь, может состоять из нескольких команд.

И наконец, изучение основных алгоритмических структур заканчивается ветвлением. Тут можно предложить такую задачу: нарисовать орнамент, состоящий из квадратов, расположенных по краю поля. На примере этой задачи еще раз демонстрируется методика последовательной детализации. Причем, в отличие от предыдущих программ, здесь используется два шага детализации, поскольку в процедуре РЯД содержится обращение к процедуре следующего уровня – КВАДРАТ.

#### *Методика преподавания «программирования»*

Рассмотрим два подхода к изучению языка программирования: формальный и «программирование по образцу». Первый основан на формальном (строгом) описании конструкции языка программирования тем или иным способом и использовании при решении задач только изученных, а, следовательно, понятных элементов языка. При втором же подходе школьникам сначала выдаются готовые программы, рассказывается, что именно они делают, и предлагается написать похожую программу или изменить имеющуюся, не объясняя до конца ряд «технических» или несущественных, с точки зрения учителя, для решения задачи деталей.

Второй подход дает возможность так называемого «быстрого старта», но создает опасность получить полуграмотных пользователей среды

программирования, т.е. людей, которые используют в своей практике достаточно сложные конструкции, но не могут четко объяснить, почему в том или ином случае нужно применять именно их, и как они работают. В результате рано или поздно так же «программисты» сталкиваются с ошибками, исправить которые они просто не в состоянии им не хватает знаний.

При работе с 10 – 11- классами используются в основном первый, формальный подход. При этом некоторыми неформальными умениями эти школьники чаще всего уже обладают. Одна из задач школьной информатики – научить именно формальному подходу, в частности, при применении различных определений. И формальное изучение языка программирования этому немало способствует. Но и без хороших примеров при обучении программированию школьников не обойтись. И чем младше ученики, тем больше примеров необходимо приводить при описании языка (иногда даже заменяя ими строгое определение). Другое дело, что, на наш взгляд, следует добиваться того, чтобы в результате обсуждения примера все его детали оказались понятны школьникам (обязательно нужно объяснить, как и почему это работает, в том числе опираясь на уже изученный формальный материал). В этом случае сильные ученики получают возможность понять все досконально и смогут использовать полученные знания в дальнейшем, а средние - приобретут конкретные навыки и оставят для себя возможность вернуться при необходимости к формальным определениям позже.

«Программирование по образцу» также применяется на занятиях при изучении визуального программирования с использованием объектной модели языка. А именно, мы учим на примерах созданию интерфейсов для оконных приложений прежде, чем знакомится с теорией объектно-ориентированного программирования.

### **1.3 Классификация методов обучения**

Единого понимания и толкования сути самого понятия "метод обучения" еще нет, то, естественно, не прекращаются и дискуссии вокруг

подхода к проблеме классификации этих методов и самой классификации. Отмечу некоторые классификации.

Первое наиболее полное описание системы методов, сложившихся в 60-е годы, дал Е.Я. Голант. [6]. Позже по вопросам классификации методов в структуре общих работ по дидактике выступили М.Н. Скаткин, И.Я. Лернер, Б.П. Есипов, М.А. Данилов и др. Обстоятельный обзор всех классификаций методов провел Ю. К.Бабанский, предложивший и свою классификацию методов.

Е.Я. Голант предложил классификацию по уровню активности учащихся. Он разделил все методы обучения на пассивные и активные в зависимости от степени включенности учащегося в учебную деятельность. К пассивным он отнес те методы, при которых учащиеся только слушают и смотрят (рассказ, лекция, объяснение, экскурсия, демонстрация), а к активным - методы, организующие самостоятельную работу учащихся (работа с книгой, лабораторный метод) (Голант Е. Я. Методы обучения в советской школе. - М., 1957). [6].

Как следующий шаг в русле этого же подхода можно расценивать классификацию по уровню включения в продуктивную (творческую) деятельность, предложенную М.Н. Скаткиным и И.Я. Лернером. Они делят методы обучения в зависимости от уровня их включенности в продуктивную, творческую деятельность и складывающегося ввиду этого характера познавательной деятельности учащихся (Лернер И.Я. Дидактические основы методов обучения. - М.: Педагогика, 1981; Дидактика средней школы / Под ред. М.Н. Скаткина. - М., 1982. -С. 181). [22]. С этой точки зрения ученые выделяют следующие методы.

Объяснительно-иллюстративный, или информационно-рецептивный (рецепция - восприятие), метод. Суть этого метода состоит в том, что учитель сообщает готовую информацию разными средствами, а учащиеся ее воспринимают, осознают и фиксируют в памяти. Сюда относятся такие приемы, как рассказ, лекция, объяснение, работа с учебником, демонстрация.

Репродуктивный метод. Заключается в воспроизведении учеником учебных действий по заранее определенному алгоритму. Используется для приобретения учащимися умений и навыков.

Проблемное изложение изучаемого материала. При работе по этому методу учитель ставит перед учащимися проблему и сам показывает путь ее решения, вскрывая возникающие противоречия. Назначение этого метода состоит в том, чтобы показать образец процесса научного познания. Учащиеся при этом следят за логикой решения проблемы, знакомятся со способом и приемом научного мышления, образцом культуры развертывания познавательных действий.

Частично-поисковый (эвристический) метод. Суть его состоит в том, что учитель расчленяет проблемную задачу на подпроблемы, а учащиеся осуществляют отдельные шаги поиска ее решения. Каждый шаг предполагает творческую деятельность, но целостное решение проблемы пока отсутствует.

Исследовательский метод. В этом случае учащимся предъявляется познавательная задача, которую они решают самостоятельно, подбирая необходимые для этого приемы. Этот метод призван обеспечить развитие у учащихся способностей творческого применения знаний. При этом они овладевают методами научного познания и накапливают опыт исследовательской, творческой деятельности.

Совершенно иной подход был использован Д. О. Лордкипанидзе (Лордкипанидзе, Д.О. Дидактика / Д.О. Лордкипанидзе. – Тбилиси: Изд-во Тбилисского ун-та) [24], предложившими классификацию по источникам получения знаний. Они считали, что на методы обучения наибольшее влияние оказывают те источники, из которых черпают знания учащиеся. На этой основе они выделили три группы методов: словесные, наглядные и практические. При таком подходе каждая группа методов опирается на свой вид источников:

- а) словесные методы - на устное или печатное слово;

б) наглядные методы - на наблюдаемые предметы, явления, наглядные пособия;

в) практические методы - на практические действия, ориентированные на получение знаний, выработку умений и навыков.

Эта классификация оказалась довольно простой. Видимо, поэтому она получила широкое распространение.

Наиболее практичной в дидактическом отношении представляется классификация по дидактическим целям, разработанная М.А. Даниловым (Данилов М. А. Процесс обучения в советской школе. - М., 1960). [6].

Они исходили из того, что если методы обучения представляют собой способы организации учебной деятельности учащихся по решению дидактических целей и задач, то, следовательно, их можно подразделять на следующие группы:

- методы приобретения новых знаний;
- методы формирования умений и навыков и применения знаний на практике;
- методы проверки и оценки знаний, умений и навыков.

Указанная классификация хорошо согласуется с основными задачами обучения и помогает лучшему пониманию их функционального назначения. Правда, в этой классификации не обозначены методы, которые применяются при закреплении (усвоении, запоминании) изучаемого материала, хотя они занимают важное место в процессе обучения.

Интересную организационную классификацию предложил Ю.К. Бабанский (Бабанский Ю.К. Методы обучения в современной общеобразовательной школе – М.: Просвещение, 1985). [1].

Он впервые выделил особую группу методов - методы стимулирования и мотивации учебно-познавательной деятельности. Эти методы ранее как бы не замечали, а между тем они используются каждым работающим в школе учителем и оказывают громаднейшее влияние на эффективность процесса обучения. Поскольку, по мнению Ю.К. Бабанского, любая деятельность в качестве неотъемлемых компонентов имеет три составляющих -

организацию, стимулирование и контроль, то и методы обучения подразделяются на три большие группы:

- Методы организации учебно-познавательной деятельности.
- Методы стимулирования учебно-познавательной деятельности.
- Методы контроля над эффективностью учебно-познавательной деятельности.

Данная классификация соответствует современному пониманию процесса обучения в школе и отражает качественно иной подход к нему. Обучение здесь рассматривается как многогранный педагогический процесс.

### *Современные методики обучения*

#### *Традиционное обучение*

Термин «традиционное обучение» подразумевает, прежде всего, классно-урочную организацию обучения, сложившуюся в XVII в. на принципах дидактики, сформулированных Я.А. Коменским, и до сих пор являющуюся преобладающей в школах мира.

Несомненным достоинством традиционного обучения является возможность за короткое время передать большой объем информации. При таком обучении учащиеся усваивают знания в готовом виде без раскрытия путей доказательства их истинности. Кроме того, оно предполагает усвоение и воспроизведение знаний, и их применение в аналогичных ситуациях. Среди существенных недостатков этого типа обучения можно назвать его ориентированность в большей степени на память, а не на мышление. Это обучение также мало способствует развитию творческих способностей, самостоятельности, активности. Наиболее типичными заданиями являются следующее: вставь, выдели, подчеркни, запомни, воспроизведи, реши по примеру и т.п. Учебно-познавательный процесс в большей степени носит репродуктивный характер, вследствие чего у учащихся формируется репродуктивный стиль познавательной деятельности.

#### *Программированное обучение*

Программированное обучение – это относительно самостоятельное и индивидуальное усвоение знаний и умений по обучающей программе с

помощью специальных средств (учебник, ЭВМ). В традиционном обучении ученик обычно читает, а затем воспроизводит текст учебника, при этом работа по воспроизведению почти никак не управляется, не регламентируется.

Часто учебную информацию ученики получают не от учителя, а из программированного пособия или с дисплея компьютера.

Программированное обучение позволяет ученику получать информацию о результатах усвоения на каждом этапе обучения и осуществлять его корректировку; развивает самостоятельность; дает возможность обучающемуся работать в оптимальном для него ритме, но не стимулирует творчество обучающегося.

#### *Поэтапное формирование умственных действий*

Данная концепция представляет собой детально разработанную систему положений о механизмах и условиях многоплановых изменений, связанных с образованием у человека новых образов, действий, понятий. Широкую известность данная концепция получила благодаря описанию этапов преобразования осваиваемых действий, их переноса во внутренний план.

П. Я. Гальперин выделял шесть этапов формирования умственных действий: 1) формирование мотивационной основы действия; 2) предварительное ознакомление с действием и условиями его выполнения; 3) формирование действия в материальном (или материализованном) виде с развертыванием всех входящих в него операций; 4) отработка действия во внешнем плане как внешне речевого; 5) проговаривание действия "про себя", во внутренней речи; 6) превращение действия во внутренний процесс мышления, осуществление его "в уме".

Считаю, что новый материал усваивается лучше с помощью произвольного запоминания в действии. Усвоение новых знаний и умений при использовании положений рассматриваемой теории происходит гораздо легче, скорее и становится доступным в более раннем возрасте, чем при использовании других форм обучения.



*Развивающее обучение*

Особенностями содержания развивающего обучения по Д.Б. Эльконину и В.В. Давыдову являлись следующие.

– специальное построение учебного предмета, моделирующее содержание и методы научной области, организующее познание учеником генетически исходных, теоретически существенных свойств и отношений объектов, условий их происхождения и преобразования;

– повышение теоретического уровня образования, передача учащимся не только эмпирических знаний и практических умений, но научных понятий, художественных образов, нравственных ценностей;

– содержание развивающего обучения построено на основе теоретических знаний, которые опираются на мысленные преобразования абстракций, отражают внутренние отношения и связи изучаемых объектов и явлений.

– основу системы теоретических знаний составляют содержательные обобщения (наиболее общие понятия науки, выражающие глубинные причинно-следственные закономерности, фундаментальные категории; понятия в которых выделены внутренние связи; теоретические образы, полученные путем мыслительных операций с абстрактными объектами).

Принцип ведущей роли теоретических знаний у Л.В. Занкова отличается от аналогичного принципа в концепции Д.Б. Эльконика и В.В. Давыдова. Центральное место в организации развивающего обучения по Л.В. Занкову занимает работа учащихся по разграничению разных признаков изучаемых объектов и явлений. Разграничение осуществляется в рамках принципов системности и целостности, доминирующий подход – индуктивный. Особое место отводится процессу сравнения, развитию анализирующего наблюдения, способности к выделению разных сторон и свойств явлений, их четкому речевому выражению. Основной мотивацией учебной деятельности является познавательный интерес.

Анализируя концепции развивающего обучения Д. Б. Эльконина, В.В. Давыдова, Л. В. Занкова можно заключить, что источником активности

является не сам обучающийся, а специальным образом организованное обучение. Включение ученика в учебную деятельность, организованную по теоретическому типу делает ее субъектом обучения, однако ученику не предоставляется право самому выбирать способы и формы учебной деятельности.

### *Проблемное обучение*

Проблемное обучение заключается в создании проблемных ситуаций, в осознании, принятии и разрешении этих ситуаций в ходе совместной деятельности обучающихся и учителя, при оптимальной самостоятельности первых и под общим направляющим руководством последнего, а также в овладении учащимися в процессе такой деятельности обобщенными знаниями и общими принципами решения проблемных задач. Принцип проблемности сближает между собой процесс обучения с процессами познания, исследования, творческого мышления.

Проблемное обучение направлено на самостоятельный поиск обучаемым новых знаний и способов действия, а также предполагает последовательное и целенаправленное выдвижение перед учащимися познавательных проблем, разрешая, которые они под руководством педагога активно усваивают новые знания. Следовательно, оно обеспечивает особый тип мышления, глубину убеждений, прочность усвоения знаний и творческое их применение в практической деятельности. Кроме того, оно способствует формированию мотивации достижения успеха, развивает мыслительные способности обучающихся.

Проблемное обучение в меньшей мере чем другие типы обучения применимо при формировании практических умений и навыков; оно требует больших затрат времени для усвоения одного и того же объема знаний по сравнению с другими типами обучения.

### *Знаково-контекстное обучение*

Как и в традиционном обучении, отмечает ученый, учебный материал предъявляется в контекстном обучении в виде учебных текстов как знаковых систем (отсюда «знаково-контекстное обучение») и по-прежнему выступает

как информация, которую нужно усвоить. Но отличительная особенность контекстного обучения в том, что за этой информацией, которая структурирована преимущественно в виде задач и проблемных ситуаций, просматриваются реальные контуры будущей профессиональной деятельности (отсюда знаково-контекстное обучение).

Построение учебного процесса на базе технологии контекстного обучения позволяет максимально приблизить содержание и процесс учебной деятельности студентов к их дальнейшей профессии.

#### *Структура учебной деятельности (Д.Б.Эльконин)*

По мысли Д. Б. Эльконина учебная деятельность - это такая деятельность, в которой воспитывается способность к самоизменению. Частица «само» и указывает на этот качественный скачок в развитии младшего школьника, который может произойти под влиянием школьного обучения. Если переход ребенка к саморазвитию к концу начальной школы произошел, то можно говорить о развивающем характере начального обучения, об обучении, осуществляемом по законам учебной деятельности. А учебная деятельность - это система таких условий обучения, которые делают возможным развитие младшего школьника: появление у него способности к самоизменению.

#### *Кейс технологий*

Кейс методы можно классифицировать в зависимости от того, какой именно творческой работы они требуют. В частности, выделяются методы инцидента, разбора деловой корреспонденции и ситуационного анализа.

Метод инцидента заключается в том, что учащийся должен сам отыскать нужную информацию для принятия решения по обозначенной проблеме. Зачастую в открытом доступе имеются разноречивые оценки того или иного события, явления, проблемы, а это означает, что учащемуся придется сформировать собственную позицию (либо стать на чью-то сторону, либо оставаться сторонним наблюдателем, констатирующим полярность мнений) и на основании этого делать выводы.

Метод разбора деловой корреспонденции предполагает получение кейса с подробным описанием ситуации: пакет документов, помогающих найти выход из сложного положения (в том числе документы, не относящиеся к данной проблеме, чтобы учащиеся могли выбирать нужную информацию) и вопросы, которые позволяют найти решение.

Самым распространенным на сегодняшний день является метод ситуационного анализа, позволяющий глубоко и детально исследовать проблему. Учащемуся предлагается текст с подробным описанием возникшей ситуации и ставится задача, требующая решения. Могут быть также предложены для анализа уже реализованные шаги. В таком случае главной задачей будет определить (путем анализа) их целесообразность.

Естественно, при использовании каждого из перечисленных методов ученики получают также пакет вопросов, на которые им необходимо найти ответы для понимания сути проблемы. Кроме того, кейс технологии предполагают как индивидуальную работу над пакетом заданий, так и коллективную, что развивает умение воспринимать мнение других людей и умение работать в команде.

Продолжать можно бесконечно, так как методов обучения огромное множество, но остановлюсь на методе обучения с использованием видеоурока. В последнее время все более набирает популярность, особенно с развитием дистанционной формы обучения.

Видеоурок – (Wikipedia) это дистанционная форма обучения, с помощью которой можно изменить классическую форму преподавания в вузе, заменив чтение лекций преподавателем у доски на запись этого действия на видеокамеру или выполняемых действий на компьютере с голосом.

## **Вывод по Главе I**

На основании рассмотренного теоретического материала по методам обучения учащихся средней школы элементам алгоритмизации и программирования на уроках информатики было установлено следующее.

Снизилось количество часов уделяемых обучению данной теме, однако не поменялись требования к степени освоения знаний и умений данного раздела программы по информатике, так как он остается фундаментом базовых знаний в этой дисциплине. Поэтому необходимо эффективное усвоение новых знаний учащимися.

Из множества методов обучения мы отметили тот что более заинтересует современных учащихся, а именно интерактивный метод на основе видеоурока.

## **Глава 2. Разработка и проведение занятий с применением метода видеоурока**

### **2.1 Роль видео-урока в образовании**

Обучающее видео (видео-урок) представляет собой обычный показ презентации или демонстрацию работы какой-либо компьютерной программы с голосовым сопровождением. В наш цифровой век потребность в передовых технологиях и методах обучения велика как никогда, и видео-уроки позволяют удовлетворить спрос на качественные знания, оказать весомую помощь, как учителю, так и ученику. Сейчас обучающее видео - это целая отрасль образования.

С помощью видео-уроков можно сделать учебный материал более наглядным. Никакими изобразительными средствами, кроме компьютерного видео, Вы не нарисуете в динамике процесс построения графика функции путем геометрических преобразований (разве что мелом на доске, но это будет не так красиво). Если в какой-то задаче по математике сказано о 10 яблоках, то условие можно наглядно продемонстрировать, не принося реальные яблоки в класс, а просто нарисовав их на мониторе ПК. Таких примеров можно приводить бесконечное множество, но суть одна - путем увеличения наглядности повышается уровень учебных достижений школьников и студентов.

Видео-уроки - это прекрасный инструмент для самоподготовки. Если вы учитель или преподаватель, и вы хотите сэкономить время и силы при подготовке к урокам, лекциям – видео-уроки облегчат вам жизнь как нельзя кстати! Потому что вы можете раздать свое обучающее видео студентам или ученикам, и предоставить им возможность самим прослушивать медиа лекции, готовиться к экзаменам. При этом к вам они будут обращаться, только если что-то не поняли. В случае если видео сделано качественно, такой подход принесет только пользу.

Конечно, применение этого медиа контента не сводится к тому, что учитель пришел, включил видеоурок и все время ученики смотрят или делают вид, что смотрят. Во время показа всегда можно и нужно делать

паузы, обсудить увиденное, задать вопросы обучающимся. Нужно, чтобы дети сами могли делать выводы из увиденного. Время демонстрации не должно превышать 10-15 минут, далее обязательно должна быть смена вида деятельности.

Желательно, чтобы ученики скопировали объяснение и домой – если все же остались непонятные моменты, всегда можно посмотреть урок еще раз.

Сам видеоурок может быть представлен в разнообразных формах, например, среди основных типов уроков можно выделить следующие три:

- урок введения (объяснения) нового материала;
- урок закрепления знаний, умений и отработки навыков;
- урок обобщения и контроля знаний, умений и навыков.

На уроках объяснения нового материала видеоуроки помогают наглядно и доходчиво изложить материал. С ними изменяется компонент обучения – получение информации. Целью данного типа урока является овладение обучающимися новым материалом. Кроме этого, на уроке, в ходе изучения нового материала, идет и работа по упорядочиванию и закреплению ранее усвоенного. Невозможно изучать новый материал, не вспоминая, не анализируя, не опираясь на уже пройденный материал, не применяя его при выводах каких-то новых положений. Видеоуроки могут использоваться и при дистанционном образовании, дающем возможность ученику и его родителям знакомиться с лекционным материалом, выполнять задания, предлагаемые автором видеоуроков, самостоятельно, что весьма актуально для временно нетрудоспособных учеников, учащихся на домашнем обучении, болеющих или находящихся в отъезде.

Теперь о технической стороне. Средняя наполняемость классов в нашей школе около 10 человек. Тогда можно показать урок, используя компьютер и проектор. Также для этого хороша и интерактивная доска. Вот лишь некоторые преимущества использования видеоуроков:

1. Экономия времени. Заранее подготовленный видеоролик позволяет экономить время урока, за счет чего повышается плотность урока.

2. Наглядность и интерактивность. Благодаря этому учащиеся активно работают на уроке. Повышается концентрация внимания, улучшается понимание и запоминание материала.

3. Формируются УУД у обучающихся, что очень важно в связи с переходом на ФГОС нового поколения.

4. Во внеурочное время создание учебного ролика (видеоурока) может быть итогом проекта, исследовательской работы и т.д.

## **2.2. Каким должен быть видеоурок**

Почему возникает такой вопрос? А потому что, когда начинаешь использовать видеоуроки на практике, мы сталкиваемся с рядом проблем. Поэтому у людей, которые разрабатывают или собираются разрабатывать собственные видеоуроки, появляются эти вопросы:

- Сколько времени оптимально должен длиться видеоурок?
- Нужна ли в видеоуроке встроенная физкультминутка?
- Какая музыка должна сопровождать физкультминутку?
- Урок лучше делать для фронтального или индивидуального просмотра?
- Нужно ли в самом видеоуроке задавать домашнее задание?
- Нужно ли делать отдельные видеофрагменты для повторения?
- Нужно ли встраивать в видеоуроки вопросы или даже тесты, которые будут обеспечивать обратную связь?

Получив в ходе своей ВКР определенный опыт работы с использованием видеоуроков, хочется высказать свое мнение по этому вопросу. Видеоурок по информатике не должен превышать 10–14 минут, потому что, в этом случае чаще всего не хватает время на практику, да и внимание учащихся столь не удерживается. Но здесь можно выделить возрастные рекомендации. Например, для учащихся помладше нужно делать видеоуроки короче, для старших классов можно разделить теорию и практику, поэтому видеоурок можно сделать немного длиннее. Хотя программа построена таким образом, что это не всегда удается. Дело в том,



что очень много материала мы должны дать и останавливаемся перед выбором или убрать часть материала или потратить больше времени на объяснение. Но оптимально время видеоурока по нашим выводам не более 10–14 минут.

Физкультминутки можно убрать или установить возможность выбора в видеоуроке. Т. е. урок останавливается в определенном моменте и предлагается продолжить просмотр или выполнить упражнения физкультминутки. Но физкультминутка не всегда удобна, потому что неизвестно когда она должна быть. Ведь просмотр видеоурока можно сделать и вначале и в конце урока. Поэтому я пришел к выводу, что физкультминутку нужно выделить как отдельный фрагмент урока, т. е. исключить её из самого видеоурока. Можно сделать отдельное меню с физкультминутками и включать их по мере необходимости, а можно и вовсе просто провести физкультминутку без каких либо заставок, т.к. отдыхать во время них особенно должны глаза. Музыка всем нравится разная, поэтому этот выбор так же должен остаться индивидуальным. Вывод физкультминутку убираем из видеоуроков.

Урок лучше делать для фронтального или индивидуального просмотра? На этот вопрос однозначного ответа нет. Как показала практика, когда мы даем видеоурок фронтально, т. е. демонстрируем его с учительского компьютера, то мы полностью управляем этим процессом и обеспечиваем сохранность времени на уроке. Если же видеоурок давать для индивидуального просмотра, то мы получаем большой плюс — учащиеся изучают тему с учетом индивидуальных особенностей, т. е. могут прокрутить и просмотреть какой-нибудь непонятный момент, включить его для повторения во время выполнения практики и т. д. Но здесь мы рискуем потерять время и возникает техническая особенность — нужны наушники на каждый компьютер.

Домашнее задание из видеоуроков лучше убрать. Т.к. при изменении программы или домашнее задание в видеоуроке не нужно учебников урок становится не совсем понятным, если вы его вовремя не выключите. Без

него видеоурок становится более универсальным, т. е. вы его сможете давать для любого класса и в любом учебном году. А домашнее мы можем дать на доске или, что еще лучше и удобнее, дать его в виде сообщения с помощью программы NetOpSchool. Это сообщение появляется поверх всех окон на компьютерах учащихся. Учащимся не нужно всматриваться в написанное на доске домашнее задание, а учитель экономит время на написание этого домашнего задания, просто скопировав фрагмент текста из плана урока.

Было бы неплохо сделать небольшие видеофрагменты для каждой темы, для того что бы у учащихся была возможность самостоятельно повторить материал и не ждать ответа учителя.

Для обеспечения обратной связи мы можем воспользоваться секретами формата swf и встроить в свои видеоуроки вопросы или даже тесты, которые будут задаваться по ходу урока. Это будет очень полезно для видеоуроков, которые мы предоставим для индивидуального просмотра, а так же можно будет использовать при фронтальном предоставлении материала, передав управление одному из учащихся либо самостоятельно спрашивая мнение всего класса. Таким образом, мы сможем поддержать должное внимание при объяснении материала. Но с другой стороны подобные видеоуроки по информатике разрабатываются несколько сложнее и дольше.

Научиться делать видеоуроки может каждый. Это не сложнее чем создать презентацию, просто нужно получить определённые знания.

### **2.3 Программное обеспечение, используемое для создания видеоурока**

С помощью видеоуроков можно решить даже самые сложные задачи, которые трудно понять, читая текст, особенно не разбираясь в терминологии, ведь весь процесс представляется в наглядном видео.

Наиболее распространенные и часто используемые программы:

1. Camtasia Studio - мощная утилита для записи изображения с экрана в видео-файлы различных форматов, имеется возможность редактирования видео, есть встроенные Macromedia Flash (SWF) и видео проигрыватели. Camtasia захватывает действия и звуки в любой части Windows-систем и

сохраняет в файл стандарта AVI. Сделанное при помощи Camtasia Studio видео можно экспортировать в один из поддерживаемых программой форматов - AVI, SWF, FLV, MOV, WMV, RM, GIF, CAMV. Кроме того, на основе любого видео может быть скомпилирован исполнительный exe-файл, который будет содержать встроенный проигрыватель. Camtasia Studio позволяет накладывать ряд эффектов, умеет работать с отдельными кадрами, облегчает запись, редактирование и опубликование высокоточного, сжатого видео для тренировки, образования, дистанционного обучения, решений технической поддержки, демонстраций продукта, торговых презентаций и т.д.

2. Программа Screen2exe предназначена для захвата изображений с экрана монитора и сохранения их в виде последовательности изображений в \*.exe файл. Проще говоря, она делает скриншоты с экрана и воспроизводит их в той же последовательности из \*.exe. Вы можете выбрать область захвата (по умолчанию вся площадь экрана будет захвачена) и количество FPS (кадров в секунду) - 1,2,4 fps и "так быстро, как только возможно"(30 fps max). Запись можно остановить/приостановить (F10/F9). Все клики мышью обозначаются графически при записи. Также возможно сохранять звук. После остановки записи, Вам предложат оптимизировать запись, внести изменения или сохранить без изменений. При оптимизации, удаляются идентичные кадры (то есть те в которых при записи не происходило никаких действий), что существенно уменьшает размер файла. Изменения, которые Вы можете внести после записи: укоротить запись, добавить всплывающий текст (только латиница), добавить картинки, сфокусироваться на определённой области. Файл можно сохранить в: наилучшем качестве, хорошем 16bit, нормальном - 4096 цветов, простом - 256 цветов, бесцветном (Grayscale). Вы также можете видеть размер конечного файла, он будет изменяться в зависимости от выбранного качества. Просмотр файлов, созданных в Screen2exe (\*.exe), возможен и без установки самой программы.

3. Jing может не только снимать видео, но и делать скриншоты. Есть возможность добавлять текстовые поля, стрелки или прямоугольники для выделения нужных моментов.

4. Программа может отправлять результаты записи на Screencast.com, Flickr, Twitter и прочие веб сервисы. Если есть микрофон, звук также можно записать.

5. Werbineria очень простая в использовании "open source" программа для записи видео. Запись ведется в формате .AVI с возможностью дальнейшей конвертации в .FLV. С помощью горячих кнопок можно легко начать\прекратить запись. Также в программе есть интересная функция объединения двух источников видео (например, веб-камера и запись с экрана).

6. Wink программа для создания уроков и презентаций, которая также может делать скриншоты и добавлять объяснительные поля, кнопки, названия и т.д. Поддерживает множество форматов .EXE, .FLV, .HTML, прочие.

7. UltraVNC Screen Recorder основан на программе CamStudio, но больше внимания уделяет продуктивности. Поддерживает аудио и скриншоты.

8. BB FlashBack Express бесплатная версия более продвинутой программы. Может записать с источника видео, экрана, аудио. Результат можно экспортировать в AVI или FLV. Имеется поддержка быстрой загрузки видео на популярные видео сервисы.

9. TipCam, кроме стандартной записи видео и аудио, также имеет следующие особенности:

- поддерживает AVI и FLV
- видео можно перезаписывать
- начало и конец можно вырезать
- видео имеет функцию зума для увеличения определенных элементов
- можно рисовать произвольные фигуры во время записи.

## 2.4 Создание видеоролика на примере CamStudio

Camtasia является очень простой в использовании программой, к тому же бесплатной в использовании. Camtasia Studio включает в себя четыре утилиты: Camtasia MenuMaker, Camtasia Player, Camtasia Theater и Camtasia Recorder. Для работы со всеми этими утилитами служит главный интерфейс программы. Camtasia Recorder Главная утилита пакета - Camtasia Recorder. Эта программа предназначена для захвата звука и изображения. По умолчанию утилита использует кодек TSCC, однако, его можно изменить на любой компрессор, установленный в системе. Лучше использовать H.264 он быстрый и при сжатии не теряет так в качестве как другие и платформа Youtube использует его же, при заливании роликов туда, они меньше теряют в качестве. Захваченное видео может содержать разнообразные пояснительные надписи. На сегодняшний день версий программы Camtasia Studio имеется большое количество, лично мне очень нравится шестая версия. Очень удобная и функциональная. В интернете можно найти бесплатные учебники по работе с этой программой.

Вся работа по созданию видеоролика состоит из 3-х частей. 1. Запись видеоролика с экрана монитора. 2. Редактирование видеоролика. 3. Сохранение видеоролика на компьютере в нужном формате. Как создать видео ролик с помощью Powerpoint и Camtasia Studio?

Если Вам необходимо создать видеоролик, то вот один из способов это сделать. Вначале воспользуемся таким программным средством, как Powerpoint от Microsoft. Эта программа позволяет создавать видеопрезентации, которые представляют собой наборы слайдов. Слайды затем запускаются последовательно на экране дисплея, и могут транслироваться в автоматическом режиме от начала и до конца. Если такую презентацию записать с помощью программы Camtasia, то получается вполне приличный видеоролик. Который затем можно загрузить на сайт типа YouTube, и который могут смотреть многие тысячи людей. Текст можно сопровождать подходящими картинками. Выбираем один из предустановленных шаблонов, или просто чистый формат листа, и начинаем

оформлять наши слайды. При этом следует иметь в виду, что среди всех шрифтов нам лучше всего подходят шрифты Sans serif – например, Arial. Если выбрать размер букв меньше, чем 14 pt, то они будут плохо различимы на экране при просмотре видеоролика. Формат видео, который используется на YouTube, составляет 640x480 пикселей. Поэтому необходимо размеры наших слайдов привести в соответствие с этими числами. В Powerpoint заходим во вкладку «Дизайн» и жмем на «Параметры страницы». Однако размеры страницы даны в сантиметрах. Выражаем требуемые размеры в сантиметрах. Числу 640 соответствует 22.58 см, а числу 480 – 16.93 см. Эти числа мы указываем в полях «Ширина» и «Высота». Ориентация при этом альбомная. Нажимаем «ОК». Теперь размеры слайдов приведены в соответствие с размером экрана.

Весь процесс можно разделить на три этапа: 1. Запись видеоурока в Camtasia Recorder 2. Редактирование (постобработка) в Camtasia Studio Editor 3. Кодирование в видеофайл. Во-первых, в углу картинки может быть отображен System Stamp - различная информация, которая включает в себя дату и время создания клипа, имя компьютера и т.д. System Stamp может постоянно находиться в углу экрана или демонстрироваться через определенные промежутки времени. Во-вторых, к отснятому видео можно добавить watermark (водяной знак). Обычно вотермарки представляют собой фирменный логотип создателя и используются для защиты авторских прав. Для создания водяного знака можно использовать любое изображение в форматах BMP, GIF или JPG. Еще один тип эффекта, который может быть применен к видео, - Caption (сопроводительная надпись). После того, как захват завершен, программа предложит сразу же открыть полученный видеофайл в основном окне Camtasia Studio. Сделанное при помощи Camtasia Studio видео можно экспортировать в один из поддерживаемых программой форматов - AVI, SWF, FLV, MOV, WMV, RM, GIF, CAMV. Кроме того, на основе любого видео может быть скомпилирован исполнительный exe-файл, который будет содержать встроенный проигрыватель. Лучше экспортировать

обычно в SWF формат, тогда вес получается меньше. Для просмотра нужен Flash Player.

## 2.5 Разработка уроков

Методическое обоснование урока: В современном информационном мире важным для человека становится умение оперативно и качественно работать с информацией, привлекая для этого современные средства и методы. Одними из важных компетенций определяемых ФГОС нового поколения являются информационные компетенции. Урок информатики с применением интерактивных методов обучения позволяют влиять на формирование и развитие информационных компетенций обучающихся, которые сегодня имеют большое значение и в профессиональной деятельности. Представленная разработка урока информатики основана на применении интерактивных методов обучения. Использование технологии приводит к повышению мотивации обучающихся, стимулированию их познавательной активности и творчества. Преподаватель только управляет ходом урока, вовлекая обучающихся в активную деятельность. Уроки разделены на этапы, каждый из которых имеет свои цели и приёмы. В процессе занятий обучающиеся самостоятельно формулируют тему и цели урока, делятся своими ожиданиями от урока, выполняют творческую работу, связанную с их будущей профессией, самостоятельно проверяют и оценивают себя. Методическая рекомендация предназначена для учителей информатики, работающих в общеобразовательных школах, учреждениях СПО.

В первом уроке на тему "Алгоритмизация и язык программирования Паскаль" в 9 классе (электронный носитель\папка: «Урок1\_класс9») использовалось видео для актуализации знаний перед изучением новой темы.

В параллельном классе же преподаватель информатики провел тот же урок только без использования видеурока, а актуализировал знания устно, в беседе с классом.

Во втором уроке на тему «Общие сведения о языке программирования Паскаль» в 9 классе (электронный носитель\ папка: «Урок2\_класс 9»)

использовался интерактивный метод (видеоурок) для изучения нового материала.

В параллельном классе, преподаватель информатики вел урок с использованием презентации (папка «Урок 2 класс 9»).

На первых этапах уроков предусмотрен организационный момент, настрой на работу и формирование навыков самоорганизации.

На втором этапе – постановка цели на урок. На этом этапе урока развиваются навыки общения со сверстниками и учителем в процессе деятельности (технология сотрудничества – переход от педагогики требований к педагогике отношений) – коммуникативные УУД, формируется – личностные УУД, вырабатывается умение ставить учебную задачу, называть цель формулировать тему. Мы вместе с учениками поставили цель и сформулировали тему урока.

В самом начале урока, ученики более концентрированы, поэтому актуально выполнение актуализации знаний, полученных на предыдущих уроках. В этом этапе происходит диалог учителя с учениками. Ученики закрепляют знания из предыдущих уроков и плавно переходят к этому уроку.

На этапе объяснения нового материала, мы показывали 10 минутные видеоуроки. Делали паузы и беседовали с учениками, если что то было не понятно. После показа ролика просили повторить учеников, то что они в нем узнали. Далее решали практические задания.

У каждого урока должен быть итог, здесь мы подводим итог всего урока, узнаем, что было интересного и необычного в нем. Выставлял оценки за работу.

Учебная работа на уроках была разнообразной: опрос, беседа, смотр видеоурока, групповая работа.

Темп работы во время урока спокойный. Характер общения с учащимися доброжелательный, создан нужный для работы микроклимат. Психологическая атмосфера поддерживается непринужденной беседой, разговором, обсуждением.



Итоги уроков: все цели, задачи, этапы уроков выполнены. В течение всех уроков учащиеся работали активно. Мне было легко вести уроки, ученики активно включались в работу. Внимание учеников как и ожидалось легко привлекли видеоуроки.

После изучения материала ученикам была предложена контрольная работа.

### **План – конспект урока информатики 9 класс.**

#### **Урок на тему: «Общие сведения о языке программирования Паскаль»**

##### *Планируемые образовательные результаты:*

- предметные – знание общих сведений о языке программирования Паскаль (история возникновения, алфавит и словарь, используемые типы данных, структура программы);

- метапредметные – умения анализа языка Паскаль как формального языка;

- личностные – представление о программировании как сфере возможной профессиональной деятельности.

##### *Решаемые учебные задачи:*

1) рассмотрение вопросов, касающихся общей характеристики языка программирования Паскаль;

2) знакомство с алфавитом и словарем языка Паскаль;

3) рассмотрение некоторыми простыми типами данных в языке Паскаль;

4) рассмотрение структуры программы на языке Паскаль;

5) рассмотрение процесса выполнения оператора присваивания.

##### *Основные понятия, изучаемые на уроке:*

- язык программирования;

- программа;

- алфавит;

- служебные слова;

- типы данных;

- структура программы;
- оператор присваивания.

*Используемые на уроке средства ИКТ:*

- персональный компьютер (ПК) учителя, мультимедийный проектор, экран;
- ПК учащихся.

*Электронные образовательные ресурсы*

- презентация «Общие сведения о языке программирования Паскаль»;
- ресурсы федеральных образовательных порталов:
  - 1) анимация «Типы величин»,
  - 2) анимация «Команда присваивания».

*Особенности изложения содержания темы урока*

1. Организационный момент (1 минута)

Приветствие учащихся, сообщение темы и целей урока.

2. Повторение (4 минуты)

Обсудить итоги зачета и сообщить оценки.

3. Изучение нового материала (20 минут)

Новый материал излагается в сопровождении презентации или видеоматериала «Общие сведения о языке программирования Паскаль» и анимаций «Типы величин», «Команда присваивания».

1 слайд — название презентации;

2 слайд — ключевые слова;

- язык программирования
- программа
- алфавит
- служебные слова
- типы данных
- структура программы
- оператор присваивания

3 слайд — паскаль;

Языки программирования — это формальные языки, предназначенные для записи алгоритмов, исполнителем которых будет компьютер. Записи алгоритмов на языках программирования называются программами.

Существует несколько тысяч языков программирования. Мы с вами познакомимся с языком программирования Паскаль, который был разработан в 70-х годах прошлого века Никлаусом Виртом (Швейцария).

Никлаус Вирт (род. в 1934 г.) - швейцарский учёный, специалист в области информатики, один из известнейших теоретиков в области разработки языков программирования, профессор информатики (компьютерных наук). Разработчик языка Паскаль и ряда других языков программирования.

4 слайд — алфавит языка (схема);

Алфавит языка программирования Паскаль - набор допустимых символов, которые можно использовать для записи программы.

- латинские прописные буквы (A, B, C, ..., X, Y, Z);

- латинские строчные буквы (a, b, c, ..., x, y, z);

- арабские цифры (0, 1, 2, ..., 7, 8, 9);

- специальные символы (знак подчёркивания; знаки препинания; круглые, квадратные и фигурные скобки; знаки арифметических операций и др.).

5 слайд — алфавит языка (схема);

В алфавит языка Паскаль включены неделимые элементы (составные символы).

:= (знак операции присваивания);

>= и <= (знаки < и >);

(\* и \*) (начало и конец комментария).

6 слайд — словарь языка (таблица);

В языке существует также некоторое количество различных цепочек символов, рассматриваемых как единые смысловые элементы с фиксированным значением. Такие цепочки символов называются служебными словами.

## Служебное слово языка Паскаль      Значение служебного слова

and	и
array	массив
begin	начало
do	выполнить
else	иначе
for	для
if	если
of	из
or	или
procedure	процедура
program	программа
repeat	повторять
then	то
to	до (увеличивая до)
until	до (до тех пор, пока)
var	переменная
while	пока

7 слайд — алфавит и словарь языка.

Имена (констант, переменных, программ и других объектов) - любые отличные от служебных слов последовательности букв, цифр и символа подчеркивания, начинающиеся с буквы или символа подчеркивания.

Прописные и строчные буквы в именах не различаются. Длина имени может быть любой.

8 слайд — простые типы данных (таблица);

В языке Паскаль используются различные типы данных. Мы будем пользоваться некоторыми из так называемых простых типов данных:

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer	- 32 768... 32 768	2 байта со знаком
Вещественный	real	$\square(2.9 * 10^{-39} \dots 1.7 * 10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт

Строковый string Последовательность символов длиной меньше 255 1  
байт на символ

логический boolean True и False 1 байт

В вещественном числе целая часть от дробной отделяется точкой, при этом перед точкой и после неё должно быть, по крайней мере, по одной цифре. Пробелы внутри числа недопустимы.

Просмотр и обсуждение анимации «Типы величин».

9 слайд — структура программы на языке Паскаль (схема);

В программе, записанной на языке Паскаль, можно выделить:

- 1) заголовок программы;
- 2) блок описания используемых данных;
- 3) блок описания действий по преобразованию данных (программный блок).

Заголовок программы состоит из служебного слова `program` и имени программы. После имени программы ставится точка с запятой.

10 слайд — раздел описания переменных (схема);

Блок описания данных состоит из раздела описания констант (`const`), раздела описания переменных (`var`) и некоторых других разделов. В разделе описания переменных указываются имена используемых в программе переменных и их тип. Имена переменных одного типа перечисляются через запятую, затем после двоеточия указывается их тип; описание каждого типа заканчивается точкой с запятой. Ниже приведён пример раздела описания переменных:

```
var i,j: integer; x: real; a: char;
```

`integer` — основной, но не единственный тип для работы с целочисленными данными. Дополнительную информацию по этому вопросу вы можете найти в справочниках по программированию на языке Паскаль.

В 9 классе мы ограничимся рассмотрением разделов описания констант и переменных, оставив рассмотрение других разделов для старшей школы.

11 слайд — общий вид программы;

Программа может не иметь заголовка; в ней может отсутствовать блок описания данных. Обязательной частью программы является программный блок. Он содержит команды, описывающие алгоритм решения задачи. Программный блок начинается со слова `begin` и заканчивается словом `end` с точкой.

Общий вид программы:

```

program <имя программы>;
  const <список постоянных значений>;
  var <описание используемых переменных>;
begin <начало программного блока>
  <оператор 1>;
  <оператор 2>;
  ...
  <оператор n>
end.

```

Операторы — языковые конструкции, с помощью которых в программах записываются действия, выполняемые над данными в процессе решения задачи.

Точка с запятой служит разделителем между операторами, а не является окончанием соответствующего оператора.

Перед оператором `end` точку с запятой ставить не нужно.

12 слайд — оператор присваивания;

Основное преобразование данных, выполняемое компьютером, - присваивание переменной нового значения, что означает изменение содержимого области памяти.

Общий вид оператора:

```
<имя переменной>:=<выражение>
```

Просмотр и обсуждение анимации «Команда присваивания».

13 слайд — выполнение оператора присваивания (схема);

Рассмотрим процесс выполнения операторов присваивания на следующем примере:

```
a:=10;
```

```
b:=5;
```

```
s:=a+b
```

При выполнении оператора `a:=10` в ячейку оперативной памяти компьютера с именем `a` заносится значение 10; при выполнении оператора `b:=5` в ячейку оперативной памяти компьютера с именем `b` заносится значение 5. При выполнении оператора `s:=a+b` значения ячеек оперативной памяти с именами `a` и `b` переносятся в процессор, где над ними выполняется операция сложения. Полученный результат заносится в ячейку оперативной памяти с именем `s`.

14 слайд — самое главное.

- Паскаль - универсальный язык программирования, получивший своё название в честь выдающегося учёного Блеза Паскаля.

- Типы данных в языке Паскаль: целочисленный (Integer), вещественный (Real), символьный (Char), строковый (String), логический (Boolean) и другие.

- В программе, записанной на языке Паскаль, можно выделить:

1) заголовок программы;

2) описание используемых данных;

3) описание действий по преобразованию данных (программный блок).

- Общий вид программы:

```
program <имя программы>;
```

```
  const <список постоянных значений>;
```

```
  var <описание используемых переменных>;
```

```
begin
```

```
  <оператор 1>;
```

```
  <оператор 2>;
```

```
  ...
```

```
  <оператор N>
```

```
end.
```

Вопросы и задания

15 слайд – вопросы и задания;

Вопросы 1-11 к параграфу 4.1

Если позволяет время выполнить задания в РТ № 172-175.

4. Практическая часть (15 минут)

В практической части урока знакомим учащихся со средой программирования PascalABC.NET. Скачать ее можно по ссылке на сайте (<http://pascalabc.net/>)

Продемонстрировать ученикам основные приемы работы в этой среде.

5. Подведение итогов урока. Сообщение домашнего задания.

Выставление оценок (4 минуты)

16 слайд — опорный конспект;

17 слайд — Д/з.

Домашнее задание.

## **Вывод по Главе 2**

Используя свой опыт в создании видео и монтажа, была разработана методическая рекомендация для создания видеоуроков.

Выделены положительные стороны таких уроков. Рассмотрен необходимый набор программ и оборудования для их создания и проведения. На примере CamStudio рассмотрели создание видеоурока.

Провели апробацию в Бийском медицинском колледже.

На основании поставленного педагогического эксперимента подтвердили успешность нашей разработки.



## **Заключение**

В заключении проведенного нами исследования можно сделать следующие основные выводы по теме.

Методов обучения множество и необходимо грамотно подходить к их использованию. Мы выбрали, набирающий популярность в последнее время, один из интерактивных методов, метод видеоурока. Определили его основные преимущества. С уменьшением часов на изучение раздела «Алгоритмизация и программирование» этот метод по моему мнению нужно использовать, он экономит время затраты, позволяет создать свою учебную базу в интернете, куда может зайти ученик и посмотреть видеоурок в любой момент, что положительно скажется на изучении нового материала и повторении старого без отвлекания учителя.

В своей работе мы разработали рекомендации по созданию видеоуроков, выделили необходимые технические и программные средства, показали на примере одной из бесплатных программ (CamStudio).

Провели на практике исследование, как повлияет использование видеоуроков при изучении нового материала. Были взяты два параллельных класса, где в одном из них все преподавалось стандартно учителем, а в другом та же тема, но с применением видеоуроков.

После был проведен тест (Приложение 2). И по результатам отмечено, что класс, где был использован метод видеоурока, справился с контрольной лучше. Что показано на диаграмме в (Приложении 1).

## Библиографический список

- 1) *Бабанский, Ю.К.* Методы обучения в современной общеобразовательной школе / Ю.К. Бабанский. – М.: Просвещение, 1985. – 208 с.
- 2) *Бочкин А.И.* Методика преподавания информатики / А.И. Бочкин. – Минск: Выш. школа, 1998. – 431 с.
- 3) Воскресная компьютерная школа [Электронный ресурс] / Система программирования Pascal ABC. – Режим доступа: <http://sunschool.math.rsu.ru>.
- 4) *Гейн А.Г., Линецкий Е.В, Сапир М.В., Шолохович В.Ф.* Информатика: Учеб. для 8-9кл. сред. шк. [Текст]: книга / Гейн А.Г., Линецкий Е.В, Сапир М.В., Шолохович В.Ф. - М.: Просвещение, 1994. – 848с.
- 5) *Гейн А.Г., Сенокосов А. И., Шояхович В.Ф.* Информатика: Классы 7-9. [Текст]: книга / Гейн А.Г., Сенокосов А. И., Шояхович В.Ф. - М.: Дрофа, 1998.– С. 23-27.
- 6) *Голант, Е.Я.* Методы обучения в советской школе / Голант Е.Я.– Москва : ГУПИ Министерства просвещения РСФСР, 1957. – 151 с.
- 7) *Гусева А.И.* Учимся информатике. Задачи и методы их решения. [Текст] / А.И. Гусева – Санкт-Петербург: Диалог-МИФИ, 2001 г.384 с.
- 8) *Данилов М. А.* Процесс обучения в советской школе/ Данилов М. А. Процесс обучения в советской школе. - М., 1960 – 167 с.
- 9) *Еремин О.Ф.* Методическое пособие по программированию на языке Pascal ABC / О. Ф. Еремин. – М.: Моздок, 2009. – 49 с.
- 10) *Ершов А-П., Монахов В.М.* Основы информатики и вычислительной техники. [Текст]: Пробное учеб. пособие для сред. учеб, заведений: В 2 ч. / Под ред. А-П. Ершова и В.М.Монахова, - М.: Просвещение, 1985 – 1986.

- 11) *Ершова А.П., Монахова В.М.* Изучение основ информатики и вычислительной техники: Пособие для учителей [текст]: книга / Под ред. Ершова А.П., Монахова В.М. - М.: Просвещение, 1985. - Ч. 1,2.
- 12) *Заборовский Г.А.* Информатика: уч. пособие для 9-го кл. / Г. А. Заборовский, А.И. Лапо, А.Е. Пупцев. – Минск: Нар. Асвета, 2009. – 191 с.
- 13) *Звонкий А.К., Кулаков А.Г., Ландо С.К.*, Алгоритмика: 5-7 классы: Учеб. и задачник для общеобразоват. учеб. заведений [Текст]: книга / Звонкий А.К., Кулаков А.Г., Ландо С.К. и др. - М.: Дрофа, 1996. – С. 59-63.
- 14) *Златопольский Д.М.* Интеллектуальные игры в информатике. [Текст] / Д.М. Златопольский – Москва: БХВ-Петербург, 2004 г.- 390 с.
- 15) Информационный канал [Sub@scribe.ru](mailto:Sub@scribe.ru) [Электронный ресурс] / Программирование на Паскале. Новая методика обучения. – Режим доступа: <http://subscribe.ru>.
- 16) *Кузнецов Л.А., Лпитова И. В.* Основы информатики. 8-9 кл [Текст]: Учеб. для общеобразоват. учеб. заведений. / Кузнецов Л.А., Лпитова И. В.. – М.: Дрофа, 1999. - 44с.
- 17) *Кузнецов А.А., Самовольнова Л.Е.*, Программы общеобразовательных учреждений. Информатика [Проект федерального компонента Государственного образовательного стандарта начального общего, основного общего и среднего (полного) образования. Образовательная область «Информатика»]// ИНФО. - 1997.-№1.-С.3-11.
- 18) *Кулаков А. Г., Ландо С. К.* Алгоритмика. 5-7 кл.: Методика, рекомендации для учителя. Решение задач. [Текст]: книга / Кулаков А. Г., Ландо С. К. – М.: Дрофа, 1997. - 94с.
- 19) *Кушниренко А.Г., Лебедев Г.В., Сворень Р.А.* Основы информатики и вычислительной техники. [Текст]: Пробный учеб. для сред, учеб. заведений / Кушниренко А.Г., Лебедев Г.В., Сворень Р.А. - М.: Просвещение, 1993. . – 14-39с.
- 20) *Лапчик М.П.* Методика преподавания информатики [Текст] / М.П. Лапчик – М.: Мир, 2003. – 440с

- 21) *Лапчик М.Л.*. Вычисления. Алгоритмизация. Программирование [Текст]: Пособие для учителя / Лапчик М.Л. – М.: Просвещение, 1988. – 27-55с.
- 22) *Лернер И.Я.* Дидактические основы методов обучения. - М.: Педагогика, 1981; Дидактика средней школы / Под ред. М.Н. Скаткина. - М., 1982. – 181 с.
- 23) *Летчик М. П.*. Информатика и информационные технологии в системе общего и профессионального образования. [Текст]: монография / *Летчик М. П.* – Омск: Изд-во Ом. гос. пед. ун-та, 1999.
- 24) *Лордкипанидзе, Д.О.* Дидактика / *Д.О. Лордкипанидзе.* – Тбилиси: Изд-во Тбилисского ун-та, 1985. – 306 с.
- 25) *Макарова Н.В.* Информатика и ИКТ. 10 класс. Базовый уровень. [Текст] / *Н.В. Макарова* – Санкт-Петербург: Питер, 2008 г.- 256 с.
- 26) *Макарова Н.В.* Информатика и ИКТ. 11 класс. Базовый уровень. [Текст] / *Н.В. Макарова* – Санкт-Петербург: Питер, 2008 г.- 224 с.
- 27) *Малев В.В.* Общая методика преподавания информатики / *В. В. Малев.* – Воронеж: ВГПУ, 2005. – 271 с.
- 28) *Мациевский С.В.* Учебное пособие [Текст] / *С.В. Мациевский, С.А. Ишанов, С.В. Клевцур* – Калининград: Изд-во КГУ, 2003. – 140 с.
- 29) *Миняйлова Е.Л.* Информатика: 9 класс: учебный курс / *Е. Л. Миняйлова, Д. А. Вербовиков, Н. Р. Коледа.* – Минск: Аверсэв, 2009. – 172 с.
- 30) *Михалевич В.С., Каныгин Ю.М., Гриценко В.И.* Информатика – новая область науки и практики Кибернетика. Становление информатики. [Текст] / *В.С. Михалевич, Ю.М. Каныгин, В.И. Гриценко* – М.: Наука, 1986. - 221с.
- 31) *Немцова Т.И.* Практикум по информатике. [Текст] / *Т.И. Немцова, Ю.В. Назарова* – Москва: Форум, Инфра-М, 2006 г.- 320 с.
- 32) *Панкратова Л.П.* Контроль знаний по информатике: тесты, контрольные задания, экзаменационные вопросы, компьютерные проекты. [Текст] / *Л.П. Панкратова, Е.Н. Челак* – Москва: БХВ-Петербург, 2004 г.- 448 с.

33) *Пейперт С.* Дети, компьютер и плодотворные идеи [текст]: статья / Пейперт С – Пер. с англ. - М.: Мир, 1990.

34) *Потапов Ю. В.* Учебно-методическое пособие. [Текст] / Ю.В. Потапов, И.Л. Фукс, -Томск: Томский гос. ун-т, 2001. – 48 с.

35) Программы для средних общеобразовательных учебных заведений. Основы информатики и вычислительной техники. - М.: Просвещение, 1992.

36) *Сафронов И.К.* Задачник-практикум по информатике [Текст] / И.К. Сафронов – Санкт- Петербург: БХВ-Петербург, 2002 г.- 432 с.

37) *Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.*, Информатика: Учеб. по базовому курсу [Текст]: книга / Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В. - М.: Лаборатория Базовых Знаний, 1998. – 47с.

38) *Совертков П.И.* Занимательное компьютерное моделирование в элементарной математике. Учебное пособие. [Текст] / П.И. Совертков – Москва: Гелиос АРВ, 2004 г.- 384 с.

39) *Софронова Н.В.* Теория и методика обучения информатике [Текст] / Софронова Н.В. Учебное пособие. — М.: Высшая школа, 2003. — 186 с.

40) *Угринович Н.Д.* Информатика и информационные технологии. Учебник для 9 класса. [текст] / Угринович Н.Д. – М.: Бином, 2009. 156 с.

41) *Ушаков Д.М.* Паскаль для школьников / Д.М. Ушаков, Т.А. Юркова. – СПб.: Питер, 2010. – 256 с.

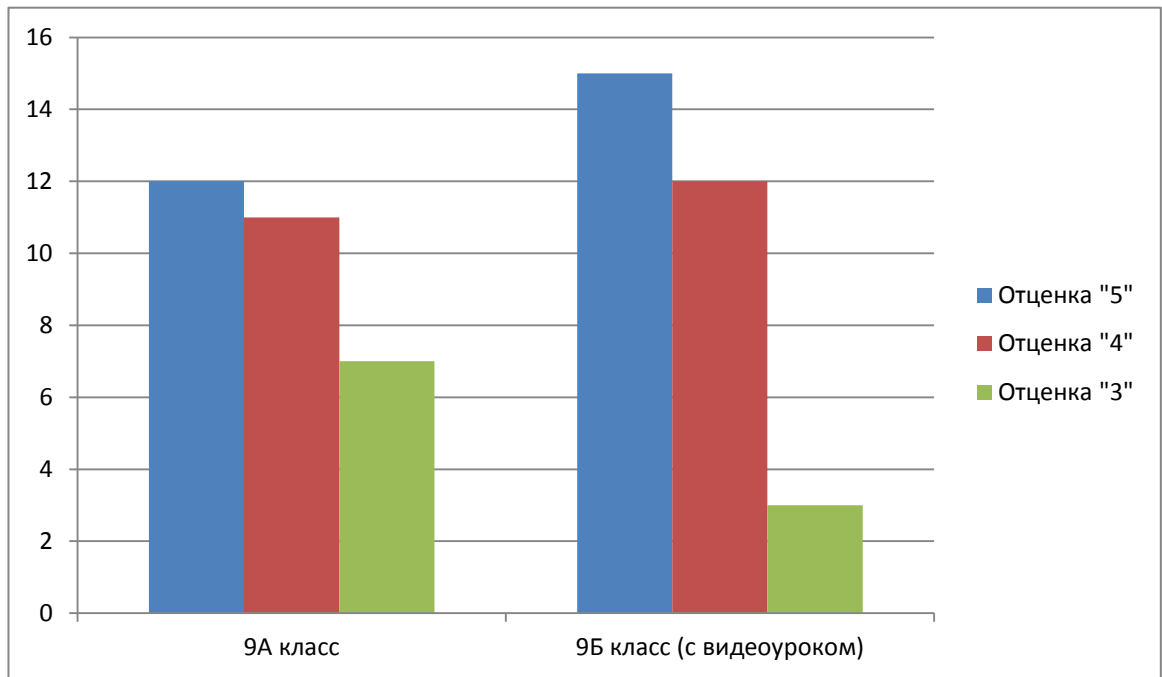
42) Фестиваль педагогических идей «Открытый урок» [Электронный ресурс] / «Алгоритмизация и программирование» в базовом курсе школы. – Режим доступа: <http://festival.1september.ru>.

43) *Шаунукова, Л.З.* Информатика: Учеб- пособие для 10 - 11 кл. общеобразовательных учреждений [Текст]: книга / Шаунукова, Л.З. – М.: Просвещение, 2000. - 133с.

44) *Шауцукова Л.З.* Информатика: Учеб. пособие для 10– 11 кл. общеобразоват. учреждений [Текст] / Л.З. Шауцукова. – М.: Просвещение, 2000.- 342с.

45) 3. Закон РФ «Об образовании» Федеральный закон от 29.12.2012 №273-Ф «Об образовании в Российской Федерации».

## Приложение 1.



## Приложение 2

### АЛГОРИТМЫ

#### ТЕСТ

1. Алгоритм - это

1. правила выполнения определенных действий;
2. ориентированный граф, указывающий порядок выполнения некоторого набора команд;
3. описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов;
4. набор команд для компьютера;
5. протокол вычислительной сети.

2. Алгоритм называется линейным, если

1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
2. ход его выполнения зависит от истинности тех или иных условий;
3. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
4. он представим в табличной форме;
5. он включает в себя вспомогательный алгоритм.

3. Алгоритм называется циклическим, если

1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
2. ход его выполнения зависит от истинности тех или иных условий;
3. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
4. он представим в табличной форме;
5. он включает в себя вспомогательный алгоритм.

4. Алгоритм включает в себя ветвление, если

1. он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
2. ход его выполнения зависит от истинности тех или иных условий;
3. его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
4. он представим в табличной форме;
5. он включает в себя вспомогательный алгоритм.

5. Свойством алгоритма является:

1. результативность;
2. цикличность;
3. возможность изменения последовательности выполнения команд;
4. возможность выполнения алгоритма в обратном порядке;
5. простота записи на языках программирования.

6. Свойство алгоритма, заключающиеся в том, что каждое действие и алгоритм в целом должны иметь возможность завершения, называется

1. дискретность;
2. детерминированность;
3. конечность;
4. массовость;
5. результативность.



7. Свойство алгоритма, заключающиеся в том, что алгоритм должен состоять из конкретных действий, следующих в определенном порядке, называется

1. дискретность;
2. детерминированность;
3. конечность;
4. массовость;
5. результативность.

8. Свойство алгоритма, заключающиеся в отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях, называется

1. дискретность;
2. детерминированность;
3. конечность;
4. массовость;
5. результативность.

9. Свойство алгоритма, заключающиеся в том, что один и тот же алгоритм можно использовать с разными исходными данными, называется

1. дискретность;
2. детерминированность;
3. конечность;
4. массовость;
5. результативность.

10. Свойство алгоритма, заключающиеся в том, что любое действие должно быть строго и недвусмысленно определено в каждом случае, называется

1. дискретность;
2. детерминированность;
3. конечность;
4. массовость;
5. результативность.

11. Выберите верное представление арифметического выражения  $\frac{x+3y}{5xy}$  на алгоритмическом языке:

1.  $x + 3y / 5xy$
2.  $x + 3*y / 5*x*y$
3.  $(x + 3y) / 5xy$
4.  $(x + 3*y) / (5*x*y)$
5.  $x + 3*y / (5*x*y)$

12. Алгоритм, записанный на «понятном» компьютеру языке программирования, называется

1. исполнителем алгоритмов;
2. программой;
3. листингом;
4. текстовкой;
5. протоколом алгоритма.

### КЛЮЧ

1	2	3	4	5	6	7	8	9	10	11	12
3	3	1	2	1	3	1	5	4	2	4	2

## Приложение 3

### Урок "Алгоритмизация и язык программирования Паскаль" 9 класс

#### Цели

##### *познавательные:*

- Углубление и систематизация знаний, учащихся по основным понятиям: алгоритм, свойства алгоритма, исполнитель, блок-схема;
- научить обрабатывать алгоритмы, записывать программы на языке Паскаль;
- стимулирование интереса учащихся к изучаемой на уроке теме и предмету "Информатика" в целом;

##### *развивающие:*

- развитие алгоритмического и логического мышления учащихся;
- развитие умения применять полученные знания при решении задач различной направленности;
- развитие памяти, внимания;
- развитие познавательного интереса;
- активизация взаимодействия между учащимися, развитие навыков групповой работы;

##### *воспитательные:*

- воспитание у учащихся самостоятельности, коллективизма, ответственности за себя и других членов коллектива.
- развивать умение отстаивать свою точку зрения, аргументировано вести диалог с одноклассниками.

#### **Оборудование урока:**

- Ноутбуки;
- Мультимедийный проектор;
- Интерактивная доска.

#### Использованные программы

- MS PowerPoint

## I. Организационный момент.

Учитель сообщает тему и цели урока.

## II. Актуализация знаний.

Мы изучаем информатику с 5 класса. На уроках мы узнали многое об информации, устройствах компьютера, алгоритмах. Заинтересовало то, что в нашей повседневной жизни нас окружают алгоритмы, любой человек выполняет свои действия по порядку, раздумывая, правильно ли он поступает.

В настоящее время слово “алгоритм” является одним из важнейших понятий науки информатики и имеет следующее определение: Алгоритм - это описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

1. Включается видеоурок с приостановками и пояснениями по следующим пунктам.

2. Вспомним основные свойства алгоритма:

**Дискретность** (от лат. discretus — разделённый, прерывистый, раздельность) (алгоритм должен состоять из конкретных действий, следующих в определенном порядке);  
**Детерминированность** (от лат. determinate – определенность, точность) (любое действие должно быть строго и недвусмысленно определено в каждом случае);  
**Конечность** (каждое действие и алгоритм в целом должны иметь возможность завершения);  
**Массовость** (один и тот же алгоритм можно использовать с разными исходными данными);  
**Результативность** (отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях).

**Понятность** алгоритма означает, что алгоритм должен содержать только те команды, которые входят в набор команд, который может выполнить конкретный исполнитель;

**Точность** алгоритма означает, что каждая команда должна пониматься однозначно.

3. Виды алгоритмов:

- **Линейный алгоритм** (описание действий, которые выполняются однократно в заданном порядке);
- **Циклический алгоритм** (описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие);
- **Разветвляющийся алгоритм** (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий);
- **Вспомогательный алгоритм** (алгоритм, который можно использовать в других алгоритмах, указав только его имя).

**Массив** — это пронумерованная последовательность величин одинакового типа, обозначаемая одним именем. Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом. Каждое из значений, составляющих массив, называется его *компонентой* (или *элементом* массива).

Если за каждым элементом массива закреплен только один его порядковый номер, то такой массив называется *линейным*. Вообще количество индексов элементов массива

определяет *размерность* массива. По этому признаку массивы делятся на одномерные (линейные), двумерные, трёхмерные и т.д.

На практике наиболее распространены следующие **формы представления алгоритмов**:

В устной форме.

В письменной форме на естественном языке.

В письменной форме на формальном языке.

Для более наглядного представления алгоритма широко используется графическая форма – **блок-схема**, которая составляется из стандартных графических объектов.

В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура.

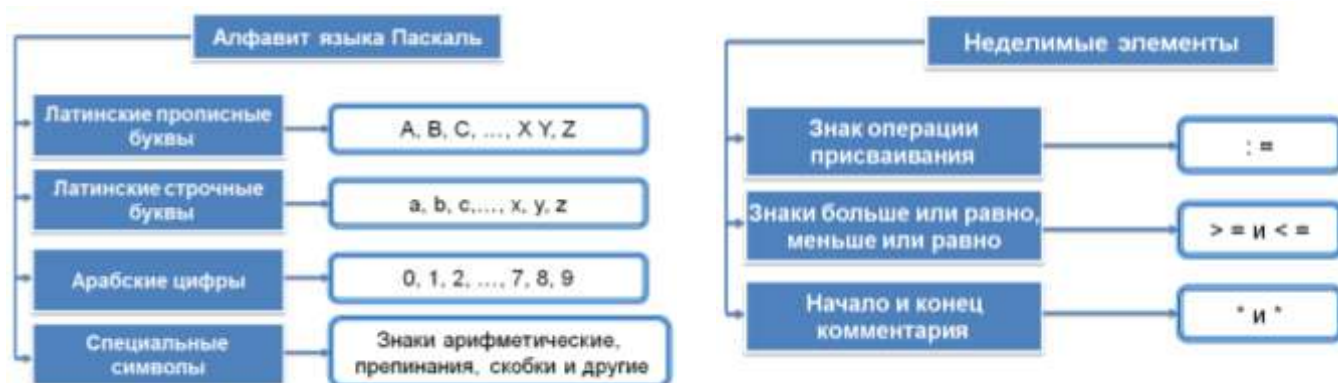
**Языки программирования** - это формальные языки, предназначенные для записи алгоритмов, исполнителем которых будет компьютер.

Записи алгоритмов на языках программирования называются **программами**.

**Язык Паскаль** – универсальный язык программирования.

### Алфавит языка

Алфавит языка программирования Паскаль - набор допустимых символов, которые можно использовать для записи программы.



### Словарь языка

На листочках даны служебные слова языка Паскаль. Нужно написать значения этих слов.

Используем структуру **РЕЛЛИ ТЭЙБЛ**. Два участника по плечу поочередно записывают свои ответы на одном листе бумаги.

Службное слово языка Паскаль	Значение служебного слова
and	и
array	массив
begin	начало
do	выполнить
else	иначе
for	для
if	если
of	из
or	или
procedure	процедура
program	программа
repeat	повторять
then	то
to	до (увеличивая до)
until	до (до тех пор, пока)
var	переменная
while	пока

Операции над целыми числами в языке Паскаль:

Операция	Обозначение	Тип результата
Сложение	+	integer
Вычитание	-	integer
Умножение	*	integer
Получение целого частного	div	integer
Получение целого остатка деления	mod	integer
Деление	/	real

**Амперсанд (&)** (иногда — **амперсенд**; от **англ. ampersand**) — знак, заменяющий слово «и».

### Этапы решения задач на компьютере



### Задача о пути торможения автомобиля

Водитель автомобиля, движущегося с некоторой постоянной скоростью, увидев красный свет светофора, нажал на тормоз. После этого скорость автомобиля стала уменьшаться каждую секунду на 5 метров. Требуется найти расстояние, которое автомобиль пройдёт до полной остановки.

#### Первый этап

Дано:

$v_{0x}$  - начальная скорость;

$v_x$  - конечная скорость (равна нулю);

$a_x$  - ускорение (равно -5 м/с<sup>2</sup>)

Требуется найти: расстояние, которое пройдёт автомобиль до полной остановки.

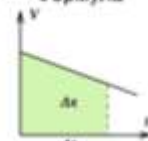


### Задача о пути торможения автомобиля

#### Второй этап

В данной ситуации мы имеем дело с прямолинейным равноускоренным движением тела. Формула для перемещения при этом имеет вид:

$$s_x = \frac{v_{0x}(v_x - v_{0x})}{a_x} + \frac{a_x}{2} \left( \frac{v_x - v_{0x}}{a_x} \right)^2$$



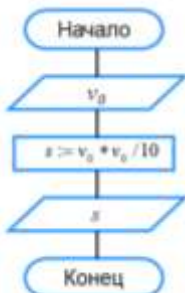
Упростим эту формулу с учётом того, что конечная скорость равна нулю:

$$s_x = \frac{v_{0x}^2}{2a_x} \quad \text{При } a_x = -5 \text{ м/с}^2 \text{ получим: } s_x = \frac{v_{0x}^2}{10}$$

### Задача о пути торможения автомобиля

#### Третий этап

Представим алгоритм решения задачи в виде блок-схемы:



### Задача о пути торможения автомобиля

#### Четвёртый этап

Запишем данный алгоритм на языке программирования Паскаль:

```

program n_2;
  var v0, s: real;
begin
  writeln('Вычисление длины пути торможения автомобиля');
  write('Введите начальную скорость (м/с): ');
  readln(v0);
  s := v0*v0/10;
  writeln('До полной остановки автомобиль пройдёт', s:8:4, ' м.')
end.
  
```

## Задача о пути торможения автомобиля

### Пятый этап

Протестировать составленную программу можно, используя ту информацию, что при скорости 72 км/ч с начала торможения до полной остановки автомобиль проходит 40 метров.

Выполнив программу несколько раз при различных исходных данных, можно сделать вывод: чем больше начальная скорость автомобиля, тем большее расстояние он пройдет с начала торможения до полной остановки.



**III. Решение задач на алгоритмы.** Используем структуру КОНЭРС. Это обучающая структура на взаимодействие УЧЕНИК – УЧЕНИК, для развития КОММУНИКАЦИИ И СОТРУДНИЧЕСТВА, «углы», обучающая структура, в которой ученики распределяются по разным углам в зависимости от выбранного им варианта ответа. Направляясь к углам класса, ученики осознают, что существует разнообразие точек зрения по данной проблеме. Обучающая структура КОНЭРС призвана для развития коммуникации и сотрудничества, своего собственного мышления, учит ценить и принимать разность точек зрения и идей. Учитель объявляет 3 угла: «линейный алгоритм», «массив чисел», «циклический алгоритм». Перед учащимися лежат карточки №1. Нужно определить к какому алгоритму относится задание, подойти к выбранному углу, обсудить, решить задачи. Учитель опрашивает учащихся из разных углов. Затем решения и ответы вводятся через сканер и каждая группа видит решения через интерактивную доску.

Задание для 1 угла. **Линейный алгоритм, записанный на алгоритмическом языке** (задание 8 из ОГЭ)

1. а) В программе «:=» обозначает оператор присваивания, знаки «+», «-», «\*» и «/» — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики. Определите значение переменной b после выполнения алгоритма:

**a := 1**

**b := 4**

**a := 2\*a + 3\*b**

**b := a/2\*b**

В ответе укажите одно целое число — значение переменной b.

**Решение.**

Выполним программу:

$a := 1$

$b := 4$

$a := 2*a + 3*b = 2 + 12 = 14$

$b := a/2*b = 7 \cdot 4 = 28.$

б) В программе «:=» обозначает оператор присваивания, знаки «+», «-», «\*» и «/» — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики. Определите значение переменной  $b$  после выполнения алгоритма:

**$a := 4$**

**$b := 10$**

**$a := b - a*2$**

**$b := 24/a*4$**

В ответе укажите одно целое число — значение переменной  $b$ .

**Решение.**

Выполним программу:

$a := 4$

$b := 10$

$a := b - a*2 = 10 - 8 = 2$

$b := 24/a*4 = 12 \cdot 4 = 48.$

в) В программе «:=» обозначает оператор присваивания, знаки «+», «-», «\*» и «/» — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики. Определите значение переменной  $b$  после выполнения алгоритма:

**$a := 4$**

**$b := 5$**

**$a := b + 15$**



$b := 100/a*4$

В ответе укажите одно целое число — значение переменной  $b$ .

**Решение.**

Выполним программу:

$a := 4$

$b := 5$

$a := b + 15 = 20$

$b := 100/a*4 = 20.$

Задание для 2 угла. Простейший циклический алгоритм, записанный на алгоритмическом языке (задание 9 из ОГЭ)

1.а) Запишите значение переменной  $s$ , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг		
нач		Var s,k: integer;
цел s, k	DIM k, s AS INTEGER	Begin
s := 0	s = 0	s := 0;
нц для k от 3 до 8	FOR k = 3 TO 8	for k := 3 to 8 do
s := s + 7	s = s + 7	s := s + 7;
кц	NEXT k	writeln(s);
вывод s	PRINT s	End.
кон		

**Решение.**

Цикл «for k := 3 to 8 do» выполняется шесть раз. Каждый раз переменная  $s$  увеличивается на 7. Поскольку изначально  $s = 0$ , после выполнения программы получим:  $s = 7 \cdot 6 = 42$ .

б) Определите, что будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг		
нач	DIM k, s AS INTEGER	Var s, k: integer;
цел s, k	s = 100	Begin
s := 100	FOR k = 1 TO 9	s := 100;
нц для k от 1 до 9	s = s - 5	for k := 1 to 9 do
s := s - 5	NEXT k	s := s - 5;
кц	PRINT s	write (s);
вывод s	END	End.
кон		

### Решение.

Цикл «for k := 1 to 9 do» выполняется девять раз. Каждый раз переменная s уменьшается на 5. Поскольку изначально  $s = 100$ , после выполнения программы получим:  $s = 100 - 9 \cdot 5 = 55$ .

в) Определите, что будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг	s = 2	Var s, k: integer;
нач	FOR k = 1 TO 5	Begin
цел s, k	s = s*2	s := 2;
s := 2	NEXT k	for k := 1 to 5 do
нц для k от 1 до 5	PRINT s	s := s*2;
s := s*2	END	write (s);
кц		End.

ВЫВОД s		
КОН		

**Решение.**

Цикл «for k := 1 to 5 do» выполняется пять раз. Каждый раз переменная s умножается на 2. Поскольку изначально  $s = 2$ , после выполнения программы получим:  $s = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 64$ .

Задание для 3 угла. **Циклический алгоритм обработки массива чисел, записанный на алгоритмическом языке** (задание 10 из ОГЭ)

1. а) В таблице Dat хранятся данные измерений среднесуточной температуры за 10 дней в градусах (Dat[1] — данные за первый день, Dat[2] — за второй и т. д.). Определите, какое число будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг		Var k, m: integer;
нач	DIM Dat(10) AS INTEGER	Dat: array[1...10] of integer;
целтаб Dat[1:10]	DIM k,m AS INTEGER	Begin
цел k, m	Dat(1) = 2:     Dat(2) = 5	Dat[1]:= 2;     Dat[2] := 5;
Dat[1]:= 2	Dat(3) = 8:     Dat(4) = 5	Dat[3]:= 8;     Dat[4] := 5;
Dat[2]:= 5	Dat(5) = 4:     Dat(6) = 2	Dat[5]:= 4;     Dat[6] := 2;
Dat[3]:= 8	Dat(7) = 0:     Dat(8) = 3	Dat[7]:= 0;     Dat[8] := 3;
Dat[4]:= 5	Dat(9) = 4:     Dat(10) =         5	Dat[9]:= 4;     Dat[10] := 5;
Dat[5]:= 4	m = 0	m := 0;
Dat[6]:= 2	FOR k := 1 TO 10	for k :=         1 to         10 do
Dat[7]:= 0	IF Dat(k)>m THEN	if Dat[k] > m then
Dat[8]:= 3	m = Dat[k]	begin
Dat[9]:= 4	ENDIF	m := Dat[k]
Dat[10] := 5	NEXT k	end;
m := 0	PRINT m	writeln(m);

нц для к от 1 до 10 если Dat[k] > m то m := Dat[k] все КЦ вывод m КОН		End.
--	--	------

**Решение.**

Программа предназначена для нахождения максимального значения среднесуточной температуры. Проанализировав входные данные, приходим к выводу, что ответ 8.

б) В таблице Dat хранятся данные измерений среднесуточной температуры за 10 дней в градусах (Dat[1] — данные за первый день, Dat[2] — за второй и т. д.). Определите, какое число будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг	DIM Dat(10) AS INTEGER	Var k, m: integer;
нач	DIM k,m AS INTEGER	Dat: array[1...10] of integer;
целтаб Dat[1:10]	Dat(1) = 2:    Dat(2) = 5	Begin
цел k, m	Dat(3) = 7:    Dat(4) = 5	Dat[1]:= 2;    Dat[2] := 5;
Dat[1]:= 2	Dat(5) = 4:    Dat(6) = 2	Dat[3]:= 7;    Dat[4] := 5;
Dat[2]:= 5	Dat(7) = 0:    Dat(8) = 3	Dat[5]:= 4;    Dat[6] := 2;
Dat[3]:= 7	Dat(9) = 4:    Dat(10) =        5	Dat[7]:= 0;    Dat[8] := 3;
Dat[4]:= 5	m = 10	Dat[9]:= 4;    Dat[10] := 5;
Dat[5]:= 4	FOR k := 1 TO 10	m := 10;
Dat[6]:= 2	IF Dat(k) < m THEN	for k :=        1 to        10 do
Dat[7]:= 0	m = Dat[k]	if Dat [ k] < m then
Dat[8]:= 3	ENDIF	begin

<pre> Dat[9]:= 4 Dat[10]      := 5 m := 10 нц для к от 1 до     10 если Dat[к] &lt; m то m := Dat[к] все КЦ вывод m КОН </pre>	<pre> NEXT k PRINT m </pre>	<pre> m := Dat[k] end; writeln(m); End. </pre>
--	-----------------------------	--

**Решение.**

Программа предназначена для нахождения значения минимального элемента массива Dat. Проанализировав входные данные, приходим к выводу, что ответ 0.

в) В таблице Dat представлены данные о количестве голосов, поданных за 10 исполнителей народных песен (Dat[1] — количество голосов, поданных за первого исполнителя, Dat[2] — за второго и т. д.). Определите, какое число будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг	DIM Dat(10) AS INTEGER	Var k, m: integer;
нач	DIM k,m AS INTEGER	Dat: array[1...10] of integer;
целтаб Dat[1:10]	Dat(1) = 56: Dat(2) = 70	Begin
цел к, м	Dat(3) = 20: Dat(4) = 41	Dat[1]:= 56; Dat[2] := 70;
Dat[1]:= 56	Dat(5) = 14: Dat(6) = 22	Dat[3]:= 20; Dat[4] := 41;
Dat[2]:= 70	Dat(7) = 30: Dat(8) = 12	Dat[5]:= 14; Dat[6] := 22;
Dat[3]:= 20	Dat(9) = 65: Dat(10) = 35	Dat[7]:= 30; Dat[8] := 12;
Dat[4]:= 41	m = 0	Dat[9]:= 65; Dat[10] := 35;

<pre>Dat[5]:= 14 Dat[6]:= 22 Dat[7]:= 30 Dat[8]:= 12 Dat[9]:= 65 Dat[10] := 35 m := 0 нц для к от 1 до   10 если Dat[к] &gt; 25 то   m := m+1 все КЦ вывод m КОН</pre>	<pre>FOR k := 1 TO 10   IF Dat(k) &gt; 25 THEN     m =m+1   ENDIF NEXT k PRINT m</pre>	<pre>m := 0; for k := 1 to 10 do   if Dat[k] &gt; 25 then     begin       m := m+1     end;   writeln(m); End.</pre>
--	--	--

### Решение.

Программа предназначена для определения количества исполнителей, за которых подали более 25 голосов. Проанализировав входные данные, приходим к выводу, что таких исполнителей было шесть.

### IV. Практическая работа.

Целая часть – div, остаток – mod.

$$z := x \text{ div } y$$

**x** - число, которое будем делить на **y** (делимое)

**y** - число, на которое будем делить число **x** (делитель)

**z** - результат целочисленного деления (целочисленное частное)

Таким образом, вот такая запись  $(55 / 6)$  нацело = 9 в результате использования **оператора div** будет выглядеть так

$$z := 55 \text{ div } 6$$

**z** будет равно 9. Помните! При использовании оператора **div** дробная часть будет отброшена!

А сейчас поговорим о делении с остатком. Оно не особо отличается и главным здесь является то, что в результате отбрасывается как раз целая часть. То есть  $(40 / 6)$  с остатком = 4,  $(10 / 3)$  с остатком = 1,  $(22 / 5)$  с остатком = 2 и т.д. В паскале для этого есть **оператор mod**. Записывается он точно так же.

$$z := x \text{ mod } y$$

**x** - число, которое будем делить на **y** (делимое)

**y** - число, на которое будем делить число **x** (делитель)

**z** - остаток

Например  $(40 / 6)$  с остатком = 4 с оператором mod будет такой

$$z := 55 \text{ mod } 6$$

И как результат получим  $z=1$ .

Задание 2. Используем структуру **РЕЛЛИ ТЭЙБЛ**. Два участника по плечу поочередно записывают свои ответы на одном листе бумаги. Это примеры короткого алгоритма на языке программирования. (задание 20 из ОГЭ).

Дописать: (первым начинает тот ученик, который стоит первым по списку).

1. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число — количество чисел, кратных 4.

Решение.

```
var n, s, k, g: integer;
begin
s:=0;
readln(n);
for k:=1 to n do
begin
readln(g);

if _____ then

s:=_____ ;

end;
writeln(s);
end.
```

Ответ:  $(g \text{ mod } 4 = 0)$                        $s+1;$

Через сканер на экране появляются ответы одной пары. Проверяем, выявляем ошибки, делаем выводы.

2. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6.

Решение.

```
var n, s, k, g: integer;
begin
s:=0;
readln(n);
for k:=1 to n do
begin
readln(g);

if _____ then

s:=_____

end;
writeln(s);
end.
```

Ответ:  $(g \bmod 6 = 0)$   $s+1$ ;

## V. Тестирование.

Первая группа учащихся (не сдающие ОГЭ) на ноутбуках выполняют тест в Excel, вторая группа (сдающие ОГЭ по информатике) составляют программу к задаче 20.

### Задание 1

*Вопрос:*

Алгоритм - это ...

*Выберите один из 5 вариантов ответа:*

- 1) правила выполнения определенных действий
- 2) ориентированный граф, указывающий порядок исполнения некоторого набора команд
- 3) понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей
- 4) набор команд для компьютера
- 5) протокол вычислительной сети

### Задание 2



Вопрос:

Укажите наиболее полный перечень способов записи алгоритмов:

Выберите один из 5 вариантов ответа:

- 1) словесный, графический, псевдокод, программный
- 2) словесный
- 3) графический, программный
- 4) словесный, программный
- 5) псевдокод

### **Задание 3**

Вопрос:

Суть такого свойства алгоритма как *результативность* заключается в том, что:

Выберите один из 5 вариантов ответа:

- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов)
- 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд
- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа
- 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату
- 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма

### **Задание 4**

Вопрос:

Суть такого свойства алгоритма как *массовость* заключается в том, что:

Выберите один из 5 вариантов ответа:

- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов)
- 2) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату
- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа
- 4) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд

5) способ решения задачи определён однозначно в виде последовательности шагов. На любом шаге не допускаются никакие двусмысленности или неопределённости

### **Задание 5**

*Вопрос:*

Суть такого свойства алгоритма как *дискретность* заключается в том, что:

*Выберите один из 5 вариантов ответа:*

- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов)
- 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд
- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа
- 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату
- 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма

### **Задание 6**

*Вопрос:*

Суть такого свойства алгоритма как *понятность* заключается в том, что:

*Выберите один из 5 вариантов ответа:*

- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов)
- 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд
- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа
- 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату
- 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма

### **Задание 7**

*Вопрос:*

Суть такого свойства алгоритма как *детерминированность* заключается в том, что:

*Выберите один из 5 вариантов ответа:*

- 1) способ решения задачи определён однозначно в виде последовательности шагов. На любом шаге не допускаются никакие двусмысленности или неопределённости
- 2) записывая алгоритм для конкретного исполнителя можно использовать лишь те команды, что входят в систему его команд
- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа
- 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определённому результату
- 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма

### **Задание 8**

*Вопрос:*

Алгоритм называется линейным:

*Выберите один из 5 вариантов ответа:*

- 1) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- 2) если ход его выполнения зависит от истинности тех или иных условий
- 3) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
- 4) если он представим в табличной форме
- 5) если он включает в себя вспомогательный алгоритм

### **Задание 9**

*Вопрос:*

Алгоритм называется циклическим:

*Выберите один из 5 вариантов ответа:*

- 1) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- 2) если ход его выполнения зависит от истинности тех или иных
- 3) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
- 4) если он представим в табличной
- 5) если он включает в себя вспомогательный алгоритм

### **Задание 10**

*Вопрос:*

Алгоритм включает в себя ветвление, если:

Выберите один из 5 вариантов ответа:

- 1) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- 2) если ход его выполнения зависит от истинности тех или иных
- 3) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
- 4) если он представим в табличной форме
- 5) если он включает в себя вспомогательный алгоритм

**Ответы:**

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

Критерии оценивания:

1 б. за каждое задание, «5» - 10 б, «4» - 8-9 б, «3» - 5-7 б, «2» - меньше 5 б.

Учащиеся, сдающие ОГЭ по информатике, пишут программу на языке Паскаль.

**20.2** Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 7 и оканчивающихся на 2. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: сумму всех чисел, кратных 7 и оканчивающихся на 2.

**Пример работы программы:**

Входные данные	Выходные данные
112 24 42 49 22	154

0	
---	--

**Решение.**

Program z1;

Var a, s: integer;

Begin

s := 0;

readln(a);

while a <> 0 do

begin

if (a mod 7 = 0) and (a mod 10 = 2) then

s := s + a;

readln(a);

end;

writeln(s);

End.

Для проверки правильности работы программы необходимо использовать следующие тесты:

	Входные данные	Выходные данные
1	21 12 112 182 0	294
2	36 0	0
3	112 0	112

**Дополнительно:**

**Простой линейный алгоритм для формального исполнителя** (задание 14 из ОГЭ)

Один ученик выполняет работу с сайта <http://inf.reshuoge.ru/test?id=679300&nt=True&pub=False>, используя интерактивную доску, остальные работают по карточкам.

**Задание 1**

У исполнителя Умножитель две команды, которым присвоены номера:

1. умножь на 2
2. прибавь 3

Первая из них умножает число на 2, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 4 числа 62, содержащий не более 5 команд. В ответе запишите только номера команд.

**Решение.**

Умножение на число обратимо не для любого числа, поэтому, если мы пойдём от числа 62 к числу 4, то однозначно восстановим программу. Полученные команды будут записываться справа налево. Если число некратно 2, то отнимаем 3, а если кратно, то делим на 2:

$62 / 2 = 31$  (команда 1);  
 $31 - 3 = 28$  (команда 2);  
 $28 / 2 = 14$  (команда 1);  
 $14 / 2 = 7$  (команда 1).  
 $7 - 3 = 4$  (команда 2).

Запишем последовательность команд в обратном порядке и получим ответ: 21121.

Ответ: 21121.

**Задание 2**

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат
2. прибавь 3

Первая из них возводит число на экране во вторую степень, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 2 числа 55, содержащий не более 5 команд. В ответе запишите только номера команд.

**Решение.**

Пойдём от конца к началу, будем извлекать корни и вычитать тройку, пока не получим из числа 55 число 2, а затем инвертируем порядок команд

$55 - 3 = 52$  (команда 2);  
 $52 - 3 = 49$  (команда 2);  
 $\sqrt{49} = 7$  (команда 1);  
 $7 - 3 = 4$  (команда 2).  
 $\sqrt{4} = 2$  (команда 1).

Искомая последовательность команд 12122.

**Задание 3**

У исполнителя Делитель две команды, которым присвоены номера:

1. раздели на 2
2. вычти 3

Первая из них уменьшает число на экране в 2 раза, вторая уменьшает его на 3. Исполнитель работает только с натуральными числами. Составьте алгоритм получения из числа 76 числа 5, содержащий не более 5 команд. В ответе запишите только номера команд.

**Решение.**

Поскольку Делитель работает только с натуральными числами и число 76 — чётное, первая команда должна быть 1. Из числа 38 число 5 можно получить последовательностью команд 1212. Следовательно, искомый алгоритм: 11212.

**Задание 4**

У исполнителя Делитель две команды, которым присвоены номера:

1. раздели на 2
2. прибавь 1

Первая из них уменьшает число на экране в 2 раза, вторая увеличивает его на 1. Исполнитель работает только с натуральными числами. Составьте алгоритм получения из числа 23 числа 4, содержащий не более 5 команд. В ответе запишите только номера команд.

**Решение.**

Поскольку Делитель работает только с натуральными числами и число 23 — нечётное, первая команда должна быть 2. Из числа 24 число 4 можно получить последовательностью команд 1112. Следовательно, искомый алгоритм: 21112.

**Задание 5**

У исполнителя Делитель две команды, которым присвоены номера:

1. раздели на 2
2. прибавь 1

Первая из них уменьшает число на экране в 2 раза, вторая увеличивает его на 1. Исполнитель работает только с натуральными числами. Составьте алгоритм получения из числа 89 числа 24, содержащий не более 5 команд. В ответе запишите только номера команд.

**Решение.**

Поскольку Делитель работает только с натуральными числами и число 89 — нечётное, первая команда должна быть 2. Из числа 90 число 24 можно получить последовательностью команд 1212. Следовательно, искомый алгоритм: 21212.

**VI. Домашнее задание.**

## Вариант 1

**1. Решение квадратного уравнения**

Квадратное уравнение имеет вид  $ax^2 + bx + c = 0$ , где  $a \neq 0$ . Требуется найти корни этого уравнения. Найдём дискриминант по формуле  $D = b^2 - 4ac$ .

Если дискриминант  $D > 0$ , то уравнение имеет 2 корня:  $x_1 = \frac{-b + \sqrt{D}}{2a}$ ;  $x_2 = \frac{-b - \sqrt{D}}{2a}$ ,

если дискриминант  $D = 0$ , то уравнение имеет 1 корень:  $x = \frac{-b}{2a}$ ,

если дискриминант  $D < 0$ , то уравнение не имеет корней.

Задание:

1. Составьте алгоритм решения квадратного уравнения.
2. Ответьте на вопросы:
  - а) в каком школьном предмете вы определяли алгоритмы?
  - в) как проявляются свойства алгоритма:
    - конечность;
    - дискретность;
    - массовость;
    - понятность
 на вашем примере?

## Вариант 2

**20.2** Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 7 и оканчивающихся на 1. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность). Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести одно число: сумму всех чисел, кратных 7 и оканчивающихся на 1.

**Пример работы программы:**

Входные данные	Выходные данные
21	112
14	
31	
28	
91	
0	



**Решение.**

**20.2** Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Паскаль:

```

var a, s: integer;

begin
s:= 0;

readln(a);

while a<>0 do

begin

if (a mod 7 = 0) and (a mod 10 = 1)

then s := s + a;

readln(a);

end;

writeln(s)

end.

```

Возможны и другие варианты решения. Для проверки правильности работы программы необходимо использовать следующие тесты:

№	Входные данные	Выходные данные
1	21 91 0	112
2	70 0	0
3	11 0	0

	10	
4	15	0
	0	

- Написать синквейн к словам «Компьютер, алгоритм».

Алгоритм написания синквейна.

**1-я строка.** Кто? Что? 1 существительное.

**2-я строка.** Какой? 2 прилагательных.

**3-я строка.** Что делает? 3 глагола.

**4-я строка.** Что автор думает о теме? Фраза из 4 слов.

**5-я строка.** Кто? Что? (Новое звучание темы). 1 существительное.

Примеры:

1. Алгоритм
2. точный, понятный.
3. составлять, действовать, выполнять
4. строгое исполнение последовательности действий
5. план

1. Компьютер
2. мощный, рабочий
3. работает, помогает, затягивает
4. хороший помощник для человека
5. ЭВМ

VII. Рефлексия. Итог, оценивание.

Подведи итог своей работы.

ФИ \_\_\_\_\_

1. Мне было интересно работать на уроке.  Да.  Нет.

2. Мне было легко выполнять задания.  Да.  Нет.

3. Мне было трудно выполнять задания.  Да.  Нет.

4. Труднее всего было выполнить задание \_\_\_\_\_

