

**Автономная некоммерческая организация высшего образования  
"Поволжский православный институт имени Святителя Алексия,  
митрополита Московского"**

Кафедра математики и информатики

Направление подготовки 44.03.01 Педагогическое образование  
Направленность (профиль) Информационные технологии

**БАКАЛАВРСКАЯ РАБОТА**

на тему:

«Разработка учебно-методических материалов по разделу «Алгоритмизация»»

Выполнил студент  
4 курса группы ИТ-401  
очной формы обучения  
Рогожкин Арсений Владимирович

\_\_\_\_\_  
(подпись)

Научный руководитель  
К.п.н., доцент  
Сыротюк Светлана Дмитриевна

\_\_\_\_\_  
(подпись)

**Допустить к защите:**  
Заведующий кафедрой

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(И.О.Ф..)

« \_\_\_ » \_\_\_\_\_ 20 г.

Тольятти  
2021

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

Кафедра математики и информатики

Направление подготовки 44.03.01 Педагогическое образование  
Направленность (профиль) «Начальное образование»

УТВЕРЖДАЮ

Зав. кафедрой математики и информатики

Е.В.Бахусова

(подпись)

(И.О.Ф.)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

**На выполнение бакалаврской работы**

Студент Рогожкин Арсений Владимирович

1. Тема: Разработка учебно-методических материалов по разделу «Алгоритмизация»
2. Срок сдачи законченной бакалаврской работы 1 июня 2020.
3. Содержание работы:

Введение (описать актуальность исследования, выявить проблему, объект исследования, предмет исследования, цель исследовательской работы, гипотезу)

Глава 1 Теоретические основы методического сопровождения раздела «Алгоритмизация» в профильном курсе информатики (осуществить анализ нормативных документов и содержания учебно-методических материалов по разделу «Алгоритмизация» в профильном курсе информатики, рассмотреть электронно-образовательные ресурсы, как средство изучения раздела «Алгоритмизация» в профильном курсе информатики)

Глава 2 Проектирование и разработка электронного практикума по разделу «Алгоритмизация» в профильном курсе информатики (осуществить проектирование структуры и содержания электронного практикума и разработка сценариев его компонентов, разработать этапы реализации электронного практикума: разработка интерфейса пользователя и функционала его компонентов; реализовать электронный лабораторный практикума для проведения внеурочной деятельности)

Глава 3 Оценка эффективности разработанного электронного практикума (дать общую характеристику исследования, осуществить методику проведения и результаты педагогического эксперимента)

## Заключение

4. Ориентировочный перечень графического и иллюстративного материала: таблицы, рисунки (диаграммы, схемы): работа должна включать карту-проект и информационные карты системы занятий с учащимися в виде таблиц, диаграммы для представления результатов эксперимента, иллюстративный материал заданий для учащихся в виде рисунков.

5. Дата выдачи задания «2» февраля 2021г.

Научный руководитель \_\_\_\_\_ Сыротюк Светлана Дмитриевна  
(подпись) (И.О.Ф.)

Задание принял к исполнению \_\_\_\_\_ Рогожкин Арсений Владимирович  
(подпись) (И.О.Ф.)

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

Кафедра математики и информатики

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

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_

(подпись)

(И.О.Ф.)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**КАЛЕНДАРНЫЙ ПЛАН  
выполнения бакалаврской работы**

на тему: Разработка учебно-методических материалов по разделу  
«Алгоритмизация»

Студента группы ИТ-401 Рогожкина Арсения Владимировича

	Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководите ля
1.	Поиск литературы и других источников, их предварительное изучение, подготовка списка источников	31.01. 2021			
2.	Формирование плана исследования, его содержания и структуры	12.02.2021			
3.	Написание разделов ВКР				
	Введение	19.02.2021			
	1 глава	05.03.2021			
	2 глава	02.04.2021			
	3 глава	07.05.2021			
4.	Формирование выводов и практических рекомендаций. Написание заключения	14.05.2021			
5.	Оформление работы	21.05.2021			

6.	Предзащита дипломной работы	04.06.2021			
7.	Исправление замечаний	06.06.2021			
8.	Представление бакалаврской работы на кафедру	07.06.2021			
9.	Получение отзыва от руководителя	08.06.2021			
10.	Получение справки о проценте оригинального текста	08.06.2021			
11.	Подготовка доклада и иллюстративных материалов для защиты	09.06.2021			
12.	Изучение отзыва руководителя. Подготовка ответов на замечания	11.06.2021			

Научный руководитель

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(И.О.Ф.)

Задание принял к исполнению

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(И.О.Ф.)

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
Глава 1 Теоретические основы методического сопровождения раздела «Алгоритмизация» в профильном курсе информатики .....	8
1.1 Анализ нормативных документов по разделу «Алгоритмизация» в профильном курсе информатики .....	8
1.2 Анализ содержания учебно-методических материалов по разделу «Алгоритмизация» в курсе информатики средней школы .....	11
1.3 Электронно-образовательные ресурсы, как средство изучения раздела «Алгоритмизация» в профильном курсе информатики .....	19
Глава 2 Проектирование и разработка электронного практикума по разделу «Алгоритмизация» в профильном курсе информатики .....	26
2.1 Проектирование структуры и содержания электронного практикума и разработка сценариев его компонентов .....	26
2.2 Этапы реализации электронного практикума: разработка интерфейса пользователя и функционала его компонентов .....	30
2.3 Разработка электронного лабораторного практикума для проведения внеурочной деятельности «Весёлая алгоритмика» .....	39
ЗАКЛЮЧЕНИЕ .....	69
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	70

## ВВЕДЕНИЕ

Возрастающий информационный поток приводит к качественному изменению способов получения, преобразования, хранения и использования информации. Необходимость переработки большого объема информации в жатые сроки привела к изменению механизма её восприятия человеком. Внимание современных школьников перестало быть длительным и глубоким, мышление стало клиповым и поверхностным. В результате этого обозначается, по мнению Семеновских Т.В. «явное несоответствие, обновленных внутренних ожиданий обладателей клипового мышления, размеренному ритму образовательных устоев» [1].

При этом как никогда системное и алгоритмическое мышление необходимо человеку для обработки, анализа, структурирования информации. Эти виды мышления лежат в основе многих учебных, ситуационных и профессиональных задач. Развитие указанных видов мышления является образовательным результатом основного базового курса информатики, согласно федеральным государственным образовательным стандартам основного общего образования.

Несмотря на то что существует большое количество методических разработок по разделу «Алгоритмизация», время отводимое на изучение данного раздела на основных уроках информатики недостаточно для полноценного формирования системного и алгоритмического видов мышления. Таким образом, возникает потребность в создании отдельного курса внеурочных занятий, на котором будет формироваться алгоритмическое и системное мышление и при этом будут учитываться когнитивные особенности учащихся современного поколения.

**Объект исследования:** процесс обучения по разделу «Алгоритмизация» в курсе информатики средней школы.

**Предмет исследования:** разработка учебно-методических материалов по разделу «Алгоритмизация».

**Цель исследования:** спроектировать, разработать и показать эффективность электронного практикума по разделу «Алгоритмизация» в профильном курсе информатики.

**Гипотеза исследования** заключается в предположении о том, что освоение учащимися профильного курса информатики по теме «Алгоритмизация» будет проходить эффективнее при условии, если использовать разработанный электронный практикум в процессе обучения по теме «Алгоритмизация».

**Задачи исследования:**

1. Проанализировать нормативные документы и научную, учебно-методическую литературу и показать определить роль раздела «Алгоритмизация» в профильном курсе информатики, а также сформулировать основные требования к его структуре, содержанию, методам, средствам и формам обучения.
2. Спроектировать структуру и содержание электронного практикума по разделу «Алгоритмизация» для профильных классов общеобразовательных школ.
3. Разработать и выполнить программную реализацию электронного практикума по разделу «Алгоритмизация».

**Методы исследования:** теоретические: анализ, сравнение, обобщение; эмпирические: тестирование, анкетирование, опрос; методы математической обработки данных диагностики.

**Теоретическая значимость:** в данной работе обобщен результат анализа узкоспециализированных и психолого-педагогических



исследований по проблеме разработки учебно-методических материалов по разделу «Алгоритмизация».

## **Глава 1 Теоретические основы методического сопровождения раздела «Алгоритмизация» в профильном курсе информатики**

### **1.1 Анализ нормативных документов по разделу «Алгоритмизация» в профильном курсе информатики**

Согласно федеральным государственным образовательным стандартам основного общего образования (ФГОС ООО), на первое место при обучении информатике в качестве приоритетного образовательного результата выходит - развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами - линейной, условной и циклической [2] является предметными образовательными результатами освоения базового курса информатики.

В методической литературе с учетом новых стандартов выделяют следующий целевые компоненты раздела «Алгоритмизация и программирование»:

- на первом месте развивающий компонент, предполагающий развитие алгоритмического мышления, важнейшего качества современного программиста;
- на втором месте практический компонент, направленный на приобретение обучающимися опыта построения и исследования алгоритмических формальных моделей;
- на третьем месте программистский компонент, позволяющий обучаемому испытать свои силы в программировании, так как

профессия программиста наиболее востребована в современном информационном обществе [2].

В соответствии с примерной образовательной программой ученик должен научиться:

- составлять алгоритмы для решения учебных задач различных типов;
- выражать алгоритм решения задачи различными способами (словесным, графическим, в том числе и в виде блок-схемы, с помощью формальных языков и др.);
- определять наиболее оптимальный способ выражения алгоритма для решения конкретных задач (словесный, графический, с помощью формальных языков);
- определять результат выполнения заданного алгоритма или его фрагмента;
- использовать термины «исполнитель», «алгоритм», «программа», а также понимать разницу между употреблением этих терминов в обыденной речи и в информатике;
- выполнять без использования компьютера («вручную») несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных, записанные на конкретном языке программирования с использованием основных управляющих конструкций последовательного программирования (линейная программа, ветвление, повторение, вспомогательные алгоритмы);
- составлять несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных с использованием основных управляющих конструкций последовательного программирования и записывать их в виде программ на выбранном языке программирования;
- выполнять эти программы на компьютере;

- использовать величины (переменные) различных типов, табличные величины (массивы), а также выражения, составленные из этих величин;
- использовать оператор присваивания;
- анализировать предложенный алгоритм, например, определять какие результаты возможны при заданном множестве исходных значений;
- использовать логические значения, операции и выражения с ними;
- записывать на выбранном языке программирования арифметические и логические выражения и вычислять их значения.

Выпускник получит возможность:

- познакомиться с использованием в программах строковых величин и с операциями со строковыми величинами;
- создавать программы для решения задач, возникающих в процессе учебы и вне ее;
- познакомиться с задачами обработки данных и алгоритмами их решения;
- познакомиться с понятием «управление», с примерами того, как компьютер управляет различными системами (роботы, летательные и космические аппараты, станки, оросительные системы, движущиеся модели и др.);
- познакомиться с учебной средой составления программ управления автономными роботами и разобрать примеры алгоритмов управления, разработанными в этой среде [3].

## **1.2 Анализ содержания учебно-методических материалов по разделу «Алгоритмизация» в курсе информатики средней школы**

Изучение раздела «Алгоритмизация» направлено на формирование у обучающихся алгоритмического мышления. Впервые понятие алгоритмического мышления описал А.П. Ершов и определил его как «умение планировать структуру действий, необходимых для достижения цели, при помощи фиксированного набора средств».

Можно утверждать, что за более чем пятидесятилетний опыт преподавания информатики и программирования в школе накоплен определённый объём научных знаний в области проектирования методик формирующих алгоритмическое мышление.

При изучении раздела «Алгоритмизация» выпускник получит возможность познакомиться с учебной средой составления программ управления автономными роботами и разобрать примеры алгоритмов управления, разработанными в этой среде.

На основании базисного учебного плана в качестве основного варианта изучения информатики в основной школе рассматривается изучение информатики в 7—9 классах с общим количеством часов — 105.

В зависимости от условий, имеющихся в конкретном образовательном учреждении, возможно увеличение количества часов до 175 с целью углублённого изучения предмета или выстраивание непрерывного курса информатики в 5—9 классах (пять лет по одному часу в неделю, общее число часов — 175).

Ниже приводится три варианта возможного распределения учебного времени:

- I вариант — 105 ч, обучение информатике в 7—9 классах;
- II вариант — 175 ч, обучение информатике в 7—9 классах;
- III вариант — 175 ч, обучение информатике в 5—9 классах.

Приказом № 254 от 20 мая 2020 г. утвержден перечень учебников, рекомендованных к использованию в 2020-2021 учебном году, а именно:

Рассмотрим современные учебники и методические пособия, рекомендованные к использованию в школах реализующую ФГОС.

Учебник «Информатика» разработанный авторским коллективом Босовой Л.Л. и Босовой А.Ю. для классов с 5 по 9.

Учебник «Информатика» разработанный авторским коллективом Полякова К.Ю. и Еремина Е.А. для классов с 7 по 9.

Учебник «Информатика» разработанный авторским коллективом Семакина Залоговой Л.А., Русакова С.В. и Шестакова Л.В. для классов с 7 по 9.

Учебник «Информатика» разработанный авторским коллективом Кушниренко А.Г., Леонов А.Г., Зайдельман Я.Н. и Тарасова В.В. для классов с 7 по 9.

Учебник «Информатика» разработанный авторским коллективом Семенов А.Л. и Рудченко Т.А. для классов с 5 по 6.

Авторским коллективом под руководством Л.Л.Босовой предлагается рассмотреть раздел «Алгоритмика» во втором полугодии 6 класса, в течении 10 учебных часов из которых 3 часа отводится на изучение теории и 7 часов на практические занятия. Для изучения предложены следующие темы: «Исполнители вокруг нас», «Формы записи алгоритмов», «Типы алгоритмов», «Управление исполнителем Чертежник». Основные понятия с которыми должны познакомиться учащиеся при изучении раздела «Алгоритмика» по методике Л.Л. Босовой: задача, последовательность действий, алгоритм, исполнитель, формальный исполнитель, система команд исполнителя, автоматизация, блок-схема, программа, линейные алгоритмы, алгоритмы с ветвлениями, алгоритмы с повторениями, исполнитель Чертежник, абсолютное смещение, относительное смещение,

вспомогательный алгоритм, основной алгоритм, цикл n раз. В учебнике по рассматриваемому нами разделу лабораторный практикум не содержится, но при изучении тем содержатся задания, выполнение которых происходит на компьютерных тренажёрах при изучении темы «Управление исполнителем». При работе в среде исполнителя отрабатываются навыки управления следующими героями: «Черепашка», «Кузнечик», «Водолей». Данные тренажёры содержатся в среде «КуМир», которую можно бесплатно скачать с официального сайта Федерального научно центр Научно-исследовательского института системных исследований Российской академии наук. В таблице 1 представлено тематическое планирование по учебнику Босовой Л.Л. «Информатика» для 6 класса.

Таблица 1-Тематическое планирование по учебнику Босовой Л.Л. «Информатика» для 6 класса

Номер урока	Тема урока	Раздел учебника
23	Что такое алгоритм. Работа в среде виртуальной лаборатории «Переправы»	§ 14
24	Исполнители вокруг нас. Работа в среде исполнителя Кузнечик	§ 15
25	Формы записи алгоритмов. Работа в среде исполнителя Водолей	§ 16
26	Линейные алгоритмы. Практическая работа 15 «Создаем линейную презентацию»	§ 17 (1)
27	Алгоритмы с ветвлениями. Практическая работа 16 «Создаем презентацию с гиперссылками»	§ 17 (2)
28	Алгоритмы с повторениями. Практическая работа 17 «Создаем циклическую презентацию»	§ 17 (3)
29	Исполнитель Чертежник. Пример алгоритма управления Чертежником. Работа в среде исполнителя Чертежник	§ 18 (1,2)
30	Использование вспомогательных алгоритмов. Работа в среде исполнителя Чертежник	§ 18 (3)
31	Алгоритмы с повторениями для исполнителя Чертежник. Работа в среде исполнителя Чертежник	§ 18 (4)
32	Обобщение и систематизации изученного по теме «Алгоритмика»	
	Всего часов: 10 часов	

Таким образом, авторы расширили основной базовый курс школьной информатики.

УМК авторского коллектива основывается на двух завершенных и обладающих преемственностью линиях учебников Л. Л. Босовой, А. Ю. Босовой «Информатика. 5–6 классы» и «Информатика. 7–9 классы». Информатика в 7–9 классах является обязательным предметом учебного плана и содержит содержательную линию «Алгоритмы и программирование». Авторский коллектив под руководством Людмилы Леонидовны делит эту содержательную линию на три модуля: «Основы алгоритмизации», «Начало программирования» и «Алгоритмы и программирование». Изучение первых двух модулей она рекомендует рассмотреть в 8 классе, а третий модуль – в 9 классе. Часы на изучение содержательной линии на базовом уровне и углубленном уровне распределены следующим образом (таблица 2):

Таблица 2 -Распределение часов на базовом и углубленном уровне

№	Название темы в программе	Часы по моделям базовой/углубленной		
		7 класс	8 класс	9 класс
3	Алгоритмы и программирования			
3.1	Основы алгоритмизации		11/26	
3.2	Начала программирования		10/16	
3.3	Алгоритмы и программирование			8/20

При рассмотрении модуля «Основы алгоритмизации» авторы предлагают следующий поурочный план (таблица 3):

Таблица 3 - Тематическое планирование по учебнику Босовой Л.Л. «Информатика» для 8 класса на базовом уровне

Номер урока	Тема урока	Параграф учебника, задания РТ
14	Алгоритмы и исполнители	§ 2.1, № 95–110
15	Способы записи алгоритмов	§ 2.2, № 111–114



16	Объекты алгоритмов	§ 2.3, № 115–125
17	Алгоритмическая конструкция следование	§ 2.4, № 126–133
18	Алгоритмическая конструкция ветвление Полная форма ветвления	§ 2.4, № 134–137, 140–146
19	Неполная форма ветвления	§ 2.4, № 138–139
20	Алгоритмическая конструкция повторение. Цикл с заданным условием продолжения работы	2.4, № 147–152
21	Цикл с заданным условием окончания работы	2.4, № 153–157
22	Цикл с заданным числом повторений	§ 2.4, № 158–166, 168
23	Алгоритмы управления	§ 2.5
24	Обобщение и систематизация основных понятий темы «Основы алгоритмизации». Проверочная работа	Глава 2, № 167

В учебно-методический комплекс Л.Л.Босовой входят:

- учебники для 5–6 классов; учебники для 7–9 классов;
- рабочие тетради (в 2 ч) для 5-6 классов; рабочие тетради (в 2 ч) для 7-9 классов;
- сборник задач для 5-6 классов; сборник задач и упражнений для 7–9 классов;
- практикумы для 5–6 классов (КуМир);
- практикум для 7–9 классов;
- сборники самостоятельных и контрольных работ для 5–6 классов;
- сборники самостоятельных и контрольных работ для 7–9 классов;
- комплект плакатов для 5–6 классов;
- комплект плакатов для 7–9 классов;
- методическое пособие для 5–6 классов; методическое пособие для 7–9 классов;
- электронные приложения к учебникам в авторской мастерской Л. Л. Босовой на сайте издательства «БИНОМ».

В рамках изучения предмета информатика реализующегося за счёт часов указанных в базисном учебном плане К.Ю.Полякова предлагает рассмотреть раздел «Алгоритмизация и программирование» в 7-9 классах. На изучение раздела на базовом уровне автор учебника отводит 27 часов, на изучение раздела на углубленном уровне 64 часов. В результате изучения раздела ученики научатся»:

- составлять алгоритмы для решения простых задач в словесной форме, на алгоритмическом языке и на выбранном языке программирования;
- выполнять трассировку алгоритма с использованием трассировочных таблиц;
- программировать несложные линейные, разветвляющиеся и циклические алгоритмы на выбранном языке программирования.

Также рассмотрению подлежит сборник программ по информатике за 7 –9 классы от автора Полякова К.Ю. Данный автор предлагает ряд поурочных планирований. Раздел «Алгоритмизация и программирование» также присутствует в данном материале. Пример поурочного планирования данного раздела в учебнике Полякова К.Ю. «Информатика» для 7-9 классов показан в таблицах 4 – 9 (предложено 2 варианта поурочного планирования: базовый и углубленный).

Таблица 4- Тематическое планирование по учебнику Полякова К.Ю. «Информатика» для 7 класса на базовом уровне

Номер урока	Тема	Часы	Раздел учебника
22	Алгоритмы и исполнители	1	§ 29
23	Формальные исполнители	1	§ 30
24	Способы записи алгоритмов	1	§ 32
25	Линейные алгоритмы	1	§ 33
26	Вспомогательные алгоритмы	1	§ 34
27	Циклические алгоритмы	1	§ 35
28	Циклы с условием	1	§ 37
29	Разветвляющиеся алгоритмы	1	§ 38

30	Ветвления и циклы	1	§ 39
	Всего часов:	9	

Таблица 5-Тематическое планирование по учебнику Полякова К.Ю. «Информатика» для 8 класса на базовом уровне

Номер урока	Тема	Часы	Раздел учебника
13	Программирование. Введение	1	§ 17
14	Линейные программы	1	§ 18
15	Операции с целыми числами	1	§ 18
16	Ветвления	1	§ 19
17	Сложные условия	1	§ 19
18	Цикл с условием	1	§ 20
19	Цикл по переменной	1	§ 20
20	Массивы	1	§ 21
21	Алгоритмы обработки массивов	1	§ 22
22	Поиск максимального элемента	1	§ 22
	Всего часов:	10	

Таблица 6 – Тематическое планирования по учебнику Полякова К.Ю. «Информатика» для 9 класса на базовом уровне

Номер урока	Тема	Часы	Раздел учебника
18	Перестановка элементов массива	1	§ 20
19	Сортировка массивов	1	§ 20
20	Сложность алгоритмов	1	§ 22
21	Как разрабатываются программы?	1	§ 23
22	Процедуры	1	§ 24
23	Функции	1	§ 25
24	Условные вычисления	1	§ 26
25	Обработка больших массивов данных	1	§ 27
	Всего часов:	8	

Таблица 7 – Тематическое планирования по учебнику Полякова К.Ю. «Информатика» для 7 класса на углубленном уровне

Номер урока	Тема	Часы	Раздел учебника
34	Алгоритмы и исполнители	1	§ 29
35	Формальные исполнители. Черепаха	1	§ 30
36	Исполнители: Шифровальщик	1	§ 30
37	Оптимальные программы	1	§ 31
38	Способы записи алгоритмов	1	§ 32
39	Блок-схемы алгоритмов	1	§ 32

40	Линейные алгоритмы	1	§ 33
41	Вычислительные задачи	1	§ 33
42	Вспомогательные алгоритмы	1	§ 34
43	Циклические алгоритмы	2	§ 35
45	Контрольная работа	1	
46	Переменные	1	§ 36
47	Процедуры с параметрами	1	§ 36
48	Циклы с условием	1	§ 37
49	Разветвляющиеся алгоритмы	1	§ 38
50	Ветвления и циклы	1	§ 39
51	Контрольная работа	1	
	Всего часов:	26	

Таблица 8 – Тематическое планирования по учебнику Полякова К.Ю. «Информатика» для 8 класса на углубленном уровне

Номер урока	Тема	Часы	Раздел учебника
25	Программирование. Введение	1	§ 17
26	Линейные программы	1	§ 18
27	Операции с целыми числами	1	§ 18
28	Операции с вещественными числами	1	§ 18
29	Случайные и псевдо- случайные числа	1	§ 18
30	Контрольная работа	1	§ 18
31	Ветвления	1	§ 19
32	Сложные условия	1	§ 19
33	Логические переменные	1	§ 19
34	Проект: экспертная система	1	§ 19
35	Цикл с условием	1	§ 20
36	Алгоритм Евклида	1	§ 20
37	Обработка потока данных	1	§ 20
38	Циклы с постусловием	1	§ 20
39	Циклы по переменной	1	§ 20
40	Циклы: практикум	1	§ 20
41	Контрольная работа	1	§ 20
	Всего часов:	23	

Таблица 9 – Тематическое планирования по учебнику Полякова К.Ю. «Информатика» для 9 класса на углубленном уровне

Номер урока	Тема	Часы	Раздел учебника
31	Символьные строки	1	§ 19
32	Операции со строками. Поиск	1	§ 19
33	Преобразования «строка-число»	1	§ 19

34	Перестановка элементов массива	1	§ 20
35	Линейный поиск в массиве	1	§ 20
36	Сортировка массивов	1	§ 20
37	Матрицы (двухмерные массивы)	1	§ 21
38	Контрольная работа	1	
39	Сложность алгоритмов	1	§ 22
40	Как разрабатываются программы?	1	§ 23
41	Процедуры	1	§ 24
42	Рекурсивные процедуры	1	§ 24
43	Функции	1	§ 25
44	Контрольная работа	1	§ 22
	Всего часов:	15	

Работа с исполнителями является важной обучающей ступенью, так как служит пропедевтикой к изучению программирования. Босова Л.Л. использует исполнители, которые необходимы школьникам на итоговом государственном экзамене - ОГЭ, но мировой опыт не использует данные среды.

Программная среда Scratch получила широкое распространение в зарубежных образовательных практиках, поскольку она первоначально и задумывалась разработчиками в качестве среды для обучения. Знание школьниками данной среды открывает возможность участвовать им в олимпиадах, что в свою очередь повышает их мотивацию к изучению программирования на углубленном уровне.

### **1.3 Электронно-образовательные ресурсы, как средство изучения раздела «Алгоритмизация» в профильном курсе информатики**

Понятие «электронные образовательные ресурсы» (ЭОР) объединяет в себе целый комплекс образовательных средств, которые разрабатываются и воспроизводятся на основе компьютерных технологий. Также можно встретить термин «ЦОР», то есть цифровые образовательные

ресурсы - то есть те, которые создаются на основе цифровых технологий и могут быть воспроизведены с помощью цифровых устройств. Основная цель использования ЭОР на занятиях - вывести образовательный процесс на новый уровень, который так необходим современным школьникам. Во-первых, электронные средства обучения представляют любую информацию в более наглядной форме и дают учащимся наиболее полное представление об изучаемых объектах и явлениях. Во-вторых, они обладают большим мотивационным потенциалом: дети любят учиться с помощью современного оборудования, самостоятельно изучать определенные темы, проверять себя и получать обратную связь. Наконец, электронный ресурс имеет большие возможности для организации большого объема данных - поэтому ЭОР может предоставить ученику гораздо больше информации, чем традиционные источники, при этом вся текстовая, визуальная и аудиоинформация будет компактно размещена на одном цифровом устройстве. Использование электронных образовательных ресурсов в учебном процессе является обязательной частью работы современного педагога. Крайне сложно проводить занятия в соответствии с ФГОС без использования современных методов и средств обучения.

В самом общем случае ЭОР включает обучающие видео и аудиозаписи, для которых достаточно домашнего магнитофона или CD-плеера. Наиболее современные и эффективные для создания ЭОР воспроизводятся на компьютере. Именно на таких ресурсах мы и сосредоточим наше внимание.

Итак, здесь и далее мы рассмотрим электронные образовательные ресурсы, требующие компьютерного воспроизведения. Самый простой ЭОР-текстуальный. Они отличаются от книг главным образом

оформлением текстов и иллюстраций-материал представлен на экране компьютера, а не на бумаге.

ЭОР следующей группы также основан на тексте, но имеет существенные различия в текстовой навигации. Мы читаем страницы книги последовательно, тем самым осуществляя так называемую линейную навигацию. В то же время в учебном тексте часто встречаются выражения или ссылки на другую часть того же текста. В таких случаях книга не очень удобна: приходится искать объяснения где-то еще и перелистывать много страниц.

В ЭОР это можно сделать гораздо удобнее: ввести неизвестное выражение и сразу же получить его определение в небольшом дополнительном окне или сразу же изменить содержимое экрана при вводе так называемого ключевого слова (или фразы). По сути, ключевое слово является аналогом строки известного содержания книги, но эта строка не помещается на отдельной странице (содержании), а вставляется в основной текст.

В этом случае текстовая навигация нелинейна (вы выводите фрагменты текста в произвольном порядке, определяемом логической связностью и вашим собственным желанием). Такой текстовый продукт называется гипертекстом.

Третий уровень ЭОР - это ресурсы, состоящие исключительно из визуального или звукового фрагмента. Формальные отличия от Книги здесь очевидны: для печатного издания кино, анимация (мультфильм) или звук невозможны.

Но с другой стороны, стоит отметить, что такие ЭОР по сути ничем не отличаются от аудио / видео продуктов, воспроизводимых на домашнем CD-плеере.

Наиболее существенными, с принципиальными отличиями являются так называемые мультимедийные ЭОР. Это самые мощные и интересные продукты для образования и заслуживают особого внимания.

Английское слово multimedia означает «много способов». В нашем случае это представление учебных объектов самыми разными способами, то есть с помощью графики, фотографий, видео, анимации и звука. Другими словами, используется все, что человек способен воспринимать с помощью зрения и слуха.

Сегодня термин «мультимедиа» используется достаточно широко, поэтому важно понимать, к чему именно он относится. Например, известный мультимедийный плеер называется мультимедийным плеером, потому что он может чередовать воспроизведение фотографий, видео, аудиозаписей и текста. Но в то же время каждый продукт, который в настоящее время воспроизводится, является «одним медиа» («два медиа» можно назвать только голосовым видео).

То же самое можно сказать и о «мультимедийной коллекции»: в общем, коллекция является мультимедийной, но каждый отдельно используемый элемент не является мультимедийным.

Говоря о мультимедийном ЭОР, мы имеем в виду возможность одновременного воспроизведения на экране компьютера и звучания определенного набора объектов, представленных различными способами. Конечно, речь не идет о бессмысленной путанице, все представленные объекты логически связаны, подчинены определенной дидактической идее, и изменение одного из них вызывает соответствующие изменения в других. Такое связанное множество объектов справедливо называют «сценой». Использование театрального термина вполне оправдано, так как чаще всего в мультимедийном ЭОР представлены фрагменты реальной или воображаемой реальности.



Степень адекватности представления фрагмента реального мира определяет качество мультимедийного продукта. Высшим выражением является «виртуальная реальность», в которой используются мультимедийные компоненты высочайшего качества для человеческого восприятия: трехмерный визуальный ряд и стереозвук.

Новое поколение ЭОР представляет собой открытую образовательную модульную мультимедийную систему (ОМС).

В своей простейшей форме это продукты электронного обучения, которые решили три основные проблемы современной ЭСМ.

Первая проблема заключалась в том, что электронные учебные ресурсы, распространяемые в интернете, были в основном текстовыми и графическими. Понятно, что электронная копия учебника не принесет никакой пользы ученику, а работа с многочисленными информационными ресурсами не характерна для школы, в отличие от университета.

Кроме того, на любом этапе обучения процесс обучения далеко не исчерпывается получением информации, необходимо также обеспечить практическую подготовку и аттестацию (желательно на основе предмета).

Очевидно, что для решения этих задач необходим ЭОР с интерактивным мультимедийным контентом, но распространение таких продуктов в глобальной сети столкнулось с серьезными техническими трудностями.

В новом поколении ЭОР была решена проблема сетевого доступа к высоко интерактивному мультимедийному контенту. Другими словами, все пять новых педагогических инструментов могут быть использованы в этих продуктах.

Вторая технологическая проблема тесно связана с решением первой. На сегодняшний день интерактивные мультимедийные продукты выпускаются на CD-ROM, причем каждый производитель использует свои

собственные программные решения, методы загрузки и пользовательские интерфейсы. Часто это приводило к тому, что обучение работе с диском требовало почти такой же тщательности и времени, как и для учебного контента.

ЕСМ нового поколения - сетевая продукция, выпускаемая разными производителями в разное время и в разных местах. Таким образом, программное обеспечение воспроизведения и пользовательский интерфейс были унифицированы. В результате для ЭОР НП была решена проблема независимости методов хранения, поиска и использования ресурса от производителя, времени и места производства.

Третья проблема специфична для образования. На протяжении многих лет было заявлено, что компьютер обеспечит личностное образование. В педагогической практике понятие индивидуальных образовательных траекторий учащихся используется уже давно.

На самом деле необходимость разных подходов к обучению разных учеников очевидна, но в классной системе это практически невозможно. Однако даже в нынешней бинарной системе «учитель – класс» учителя все равно разные, каждый из них хочет учить по-своему. Соответственно, ЭОР должен допускать создание авторских учебных курсов.

В традиционных условиях преподаватель полностью свободен в использовании различных информационных ресурсов (учебники различных издательств, методические материалы, научные публикации...) и ограничен в плане практических занятий (например, лабораторно-селекционные комплексы-это, увы, утопия). Поэтому, когда первые серьезные учебные продукты появились на CD-ROM, их недостатки сразу же приписали жесткой роли учебного курса. Учитель хотел бы что-то изменить, но очень много разных специалистов занимаются созданием

интерактивного мультимедийного контента, которого, конечно же, в школе нет.

В новом поколении ЭОР также решается проблема создания авторского образовательного курса и индивидуальных образовательных траекторий для учащихся.

Таким образом, ЭОР, как средство изучения темы «Алгоритмизация» в профильном курсе информатики является эффективным средством обучения, способствующем более глубокому усвоению теоретического материала и отработке практических навыков у учащихся. В качестве разрабатываемого ЭОР будет рассмотрен электронный практикум. Во второй главе рассмотрим проектирование электронного практикума по разделу «Алгоритмизация».

## **Глава 2 Проектирование и разработка электронного практикума по разделу «Алгоритмизация» в профильном курсе информатики**

### **2.1 Проектирование структуры и содержания электронного практикума и разработка сценариев его компонентов**

При изучении раздела «Алгоритмизация» в школе перед учителем стоят следующие *задачи*:

- 1) сформировать у обучающихся системное мышление;
- 2) сформировать у обучающихся алгоритмическое мышление.

Таким образом, главной целью обучающегося, при освоении раздела «Алгоритмизация» в школьном курсе является овладеть структурной методикой построения алгоритмов, в том числе и построения алгоритмов для формальных исполнителей. Главным достоинством учебных формальных исполнителей является: ясность для ученика решаемых задач, наглядность процесса работы в ходе выполнения программы.

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

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

Перед началом использования любого исполнителя алгоритмов, учителю следует описать его характеристики, совокупность которых называется архитектурой исполнителя. К ним относят:

- 1) среда, в которой работает исполнитель;

- 2) режим работы исполнителя;
- 3) система команд исполнителя;
- 4) данные с которыми работает исполнитель.

Обучение составлению кода для программы исполнителя лучше организовать в ходе решения задач, подобранных с применением следующих дидактических принципов:

- 1) от простого к сложному (т.е. постепенное усложнение рассматриваемых задач);
- 2) новизна (т.е. каждая задача рассматриваемая в классе должна носить новый элемент знаний);
- 3) наследование (решение каждой новой задачи требует использование знаний полученных ранее).

Для чёткого представления структуры основных видов алгоритмов необходимо использовать блок-схемы, они дадут дополнительную наглядность и помогут легче понять теоретический материал. С ними можно познакомиться на одном уроке (в том числе и на обычном уроке информатики), а затем продолжать изучение алгоритмизации и блок-схем совместно с построением алгоритмов на учебных исполнителях. Такой подход поможет изучить основные алгоритмические структуры с теоретической и практической стороны.

На практических занятиях необходимо использовать следующие типы задач:

- 1) построение линейных алгоритмов;
- 2) использование разветвляющихся алгоритмов;
- 3) циклические алгоритмы разных видов.

При разработке алгоритма для конкретного исполнителя необходимо обратить внимание учеников на следующее обстоятельство: алгоритм зависит не только от сформулированной цели, но и от исходного

положения исполнителя. Если бы исходное состояние было другим, то был бы другим и алгоритм (или часть алгоритма). Таким образом, исходное состояние исполнителя является исходными данными задачи. Состояние исполнителя определяется местом его расположения на поле и его ориентацией. Результатом его работы становится не только «рисунок движения», но и конечное состояние исполнителя.

Учебные программы внеурочной деятельности, как правило, рассчитаны на реализацию в течение периода обучения полгода, год. Это ставит учителя разрабатывающего программу в конкретные временные рамки, а именно, если программа реализуется 1 раз в неделю в течении 0,5 года, то необходимо запланировать деятельность с обучающимися на 17 учебных часов. Спроектируем программу внеурочной деятельности исходя из выше описанных особенностях изучения алгоритмов для исполнителей и представим результат в таблице №10.

Таблица 10 – Тематическое программы внеурочной деятельности «Весёлая алгоритмика»

Тема	Количество часов
История программирования. Формальные и неформальные исполнители.	1
Машина Тьюринга	2
Наглядные алгоритмы из блоков.	2
Роботы среди людей. Перспективы и проблемы.	2
Лабораторная работа №1. Знакомство со средой Scratch. Внешний вид среды, поля. Алгоритмы. Линейные алгоритмы.	2
Лабораторная работа №2. Ветвление.	2
Лабораторная работа №3. Циклические алгоритмы. Цикл с параметром.	2
Лабораторная работа №4. Циклический алгоритм с предусловием и постусловием.	2
Лабораторная работа №5. Музыка чисел.	2
<b>ИТОГО:</b>	<b>17</b>

Проектирование электронного практикума можно разделить на несколько этапов:

Постановка цели и задач электронного практикума, а также определение ожидаемых результатов и критерии оценки качества разработки.

Разработка модели электронного практикума: определение назначения электронного практикума, его типа, структуры и требований к его разработке и использованию.

Разработка сценариев компонента электронного практикума: взаимодействия обучающихся, реализации обратной связи с педагогом и получение последнего информации о результатах обучения.

Реализация электронного практикума: разработка методических указаний для учителей по использованию ресурса в процессе обучения [23, 26].

Целью разработанного данного курса является формирование навыков самоанализа и систематизации знаний, которые получают учащиеся в процессе обучения.

Для подведения промежуточных результатов обучения учащихся, следует подобрать средства измерения и оценки знаний.

Также следует определить основные требования к разработке электронного практикума. Электронный практикум должны содержать основную информацию и сведения, которые интересно представлены, но не слишком длинные и монотонные (в пределах от 3 до 15 минут, т. к. существует психологическая трудность усвоения информации при долгой и однообразной деятельности). Они должны доступно разъяснять трудные моменты, новые термины и понятия, а также соответствовать уровню знаний обучающихся.

Практические работы и интерактивные упражнения в свою очередь должны соответствовать своей теме, а также уровню знаний учащихся. Задания должны быть посвящены повторению и закреплению того теоретического материала, который был ранее пройден.

Таким образом, после проектирования и разработки электронного практикума, можно приступить к следующему этапу – реализации программного проекта.

## **2.2 Этапы реализации электронного практикума: разработка интерфейса пользователя и функционала его компонентов**

Практикум – учебное пособие с практическими заданиями, упражнениями, лабораторными работами, помогающее освоить учебную дисциплину практически.

Электронный практикум– это интегрированное средство, выступающее как компонент поддержки учебного процесса и включающее практические занятия и упражнения для усвоения пройденного теоретического материала [27, 29].

Практикум дополнен теоретической информацией, задания в котором выполнены по шагам и с объяснением действий. Суть практикума – наглядно показать все действия, чтобы в дальнейшем обучаемый мог самостоятельно проделать ту же работу.

При разработке электронного практикума необходимо учитывать следующие критерии [11]:

1. Научность содержания – обеспечение возможности построения содержания учебной деятельности с учетом основных принципов



педагогике, психологии, кибернетики, теории высшей нервной деятельности.

2. Обеспечение мотивации – стимулирование постоянной высокой мотивации обучаемых, подкрепляемой целенаправленностью, активными формами работы, высокой наглядностью, своевременной обратной связью, причем мотивация не должна идти за счет интереса к самому компьютеру.

3. Целенаправленность – обеспечение обучающегося постоянной информацией о ближайших и отдалённых целях обучения, степени достижения целей; стимуляция тех видов познавательной активности обучающихся, которые необходимы для достижения основных учебных целей.

4. Наличие входного контроля – диагностика обучаемого перед началом работы с целью обеспечения индивидуализации обучения, а также оказания требуемой первоначальной помощи.

5. Креативность – обеспечение подготовки специалистов с творческим потенциалом, способных самостоятельно ставить и решать проблемы; исключение таких нежелательных последствий компьютеризации, как чрезмерная алгоритмизация мыслительной деятельности и пассивность мышления.

6. Индивидуализация обучения – содержание учебного предмета и трудность учебных задач должны соответствовать возрастным возможностям и индивидуальным особенностям обучаемых и строиться с учётом их уже приобретенных знаний и умений.

7. Обеспечение систематической обратной связи – обратная связь должна быть педагогически оправданной, не только сообщать о допущенных ошибках, но и содержать информацию достаточную для их устранения.

8. Педагогическая гибкость – программа должна позволять обучаемому самостоятельно принимать решения о выборе стратегии обучения, характере помощи, последовательности и темпе подачи учебного материала; должна быть обеспечена возможность доступа к ранее пройденному учебному материалу, выхода из программы в любой ее точке.

9. Открытость – возможность реализации любого способа управления учебной деятельностью, выбор которого обусловлен, с одной стороны, теоретическими воззрениями разработчиков обучающей программы, а с другой – целями обучения.

10. Возможность возврата назад – при самостоятельной работе должна быть предусмотрена отмена обучающимся ошибочных действий.

11. Структурная целостность – знания должны быть представлены в виде укрупнённых дидактических единиц, сохраняющих логику, главные идеи и взаимосвязи изучаемого материала.

12. Наличие развитой системы помощи – система помощи должна быть многоуровневой, педагогически обоснованной, достаточной для того, чтобы решить задачу и усвоить способ ее решения.

13. Наличие интуитивно понятного дружелюбного интерфейса – обучаемый не должен затрачивать значительных усилий на освоение основных приёмов работы в программе, обеспечение возможностей изменения шрифтов учебного текста, цвета фона.

Из требований, предъявляемых к электронному практикуму можно выделить:

- наглядность представленной информации;
- возможность возврата;
- пошаговая структура заданий;
- наличие заданий для самоконтроля.

При создании электронного практикума в соответствии с основными требованиями, можно выделить следующие элементы:

1. Аннотация, краткая характеристика которая включает в себя назначение электронного практикума, педагогический адрес, и описание программного обеспечения.

2. Методические указания, содержат рекомендации по использованию электронного практикума.

3. Практический блок, состоит из лабораторных работ или уроков.

4. Блок контроля, состоит из контрольных заданий и вопросов, промежуточных и итоговых тестов.

Структуру практического занятия можно представить в трех основных частях:

- описание;
- основной блок;
- заключительный блок.

В описании, как правило, формулируется тема работы, устанавливаются цели и задачи, так же излагаются теоретические основы, для формирования у обучаемого представления о практической работе.

Тема работы должна четко указывать на предмет практического изучения. Цель работы должна отражать ее познавательно-практическую направленность. Задачи работы должны указывать на умения и навыки, которые приобретаются студентом при выполнении работы.

В основном блоке содержится набор заданий и объяснения к их выполнению для выполнения целей и задач, поставленных в данном уроке.

Заключительный блок состоит из контрольных вопросов, индивидуальных заданий, промежуточных и итоговых тестов. Они предназначены для самопроверки надежности знаний и умений, приобретенных в ходе выполнения практической работы.

При разработке электронного практикума так же необходимо уделить особое внимание интерфейсу продукта.

Интерфейс – аппаратно-программные средства, обеспечивающие графическое отображение и обмен информацией между человеком и компьютером [29].

Также требования к пользовательскому интерфейсу условно могут быть разбиты на две группы [19-22]:

- требования к внешнему виду пользовательского интерфейса и формам взаимодействия с пользователем;
- требования по доступу к внутренней функциональности системы при помощи пользовательского интерфейса.

Общими требованиями к пользовательскому интерфейсу являются [11]:

- унифицированность;
- дружелюбность;
- лаконичность;
- гибкость.

После того, как необходимый теоретический, практический и материал для проверки знаний готовы, можно преступать к загрузке готового курса на выбранную платформу.

В качестве программно-методической поддержки раздела «Алгоритмизация» был разработан сайт –электронный практикум «Алгоритмизация» на основе CMS WordPress. CMS WordPress — это система управления контентом, по английски сокращенно CMS, расшифровка Content Management System. Система управления контентом– это программа, которая устанавливается на хостинг, для упрощения администрирования и управления сайтом. Хостинг – хранилище, на

котором располагаются файлы сайта. Тот же компьютер, но подключенный к интернету и работающий 24 часа в сутки.

Электронный образовательный ресурс расположен по адресу <https://algoritm401.wordpress.com/>

На рисунке 1 представлена главная страница сайта.

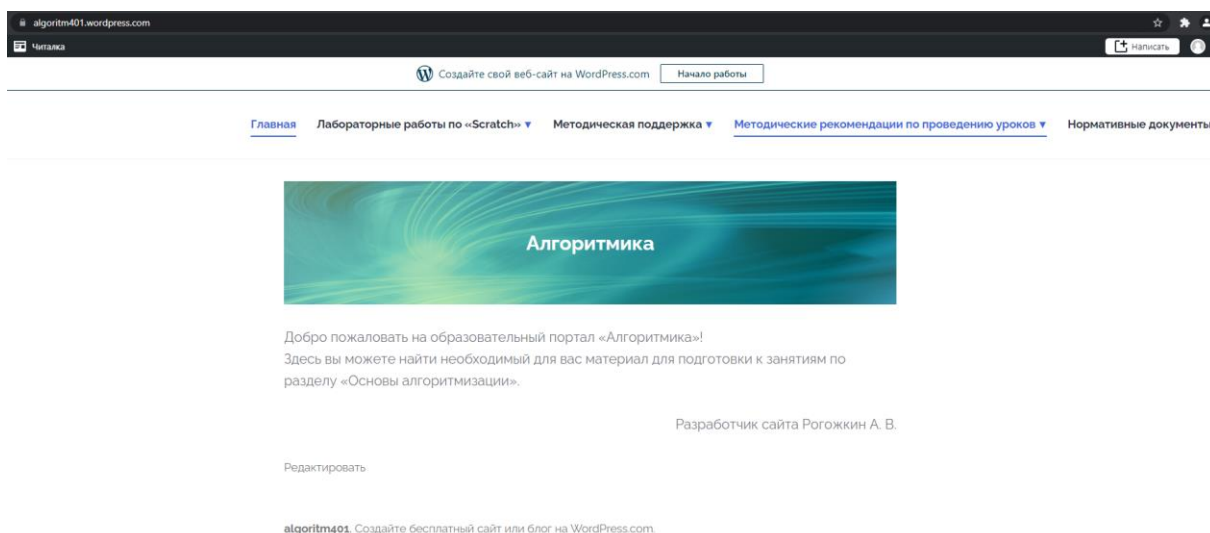


Рисунок 1- Главная страница

Верхнее меню состоит из 4 разделов: «Главная», «Методические рекомендации по проведению уроков», «Методическая поддержка», «Нормативные документы», «Обратная связь».

Страница «Методические рекомендации по проведению уроков» содержит основную информацию по каждому уроку (цель, задачи, электронно-образовательные ресурсы). Для удобства рекомендации к урокам расположены на отдельных страницах (рисунки 2).

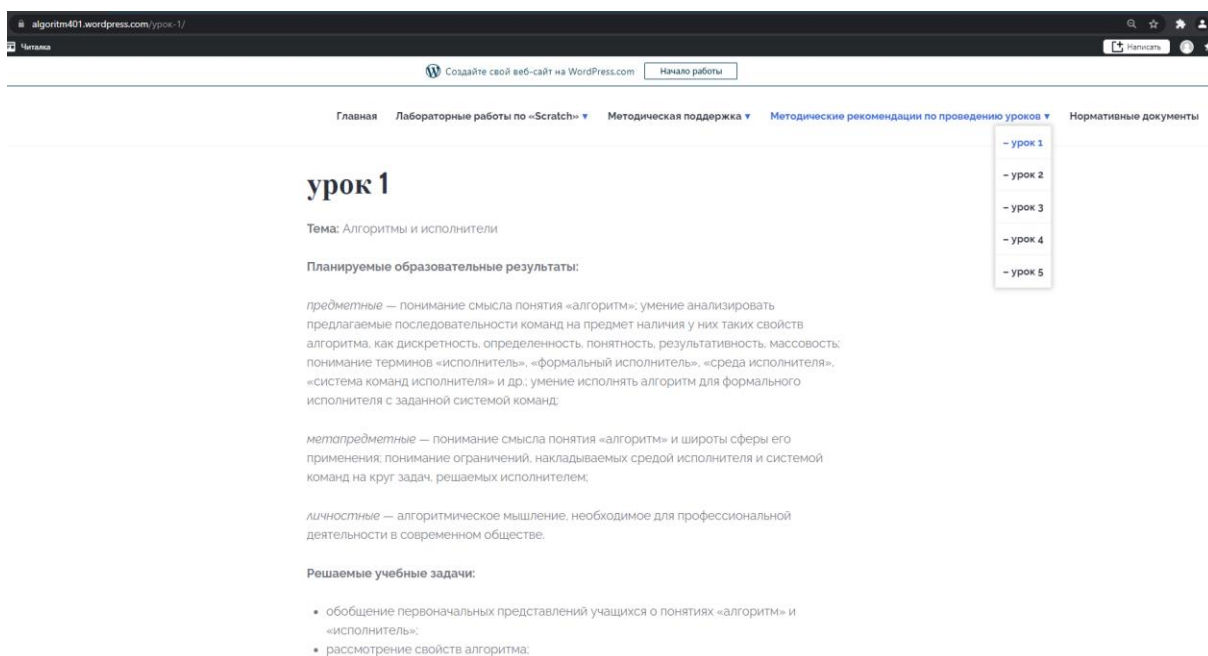


Рисунок 2- Фрагмент урока по теме №1 «Алгоритмы и исполнители»

Страница «Методическая поддержка» содержит разделы (рисунок 4):

- Практические работы.
- Презентации.
- Самостоятельная работа

На странице «Практические работы» представлены практические работы (Рисунок 4).

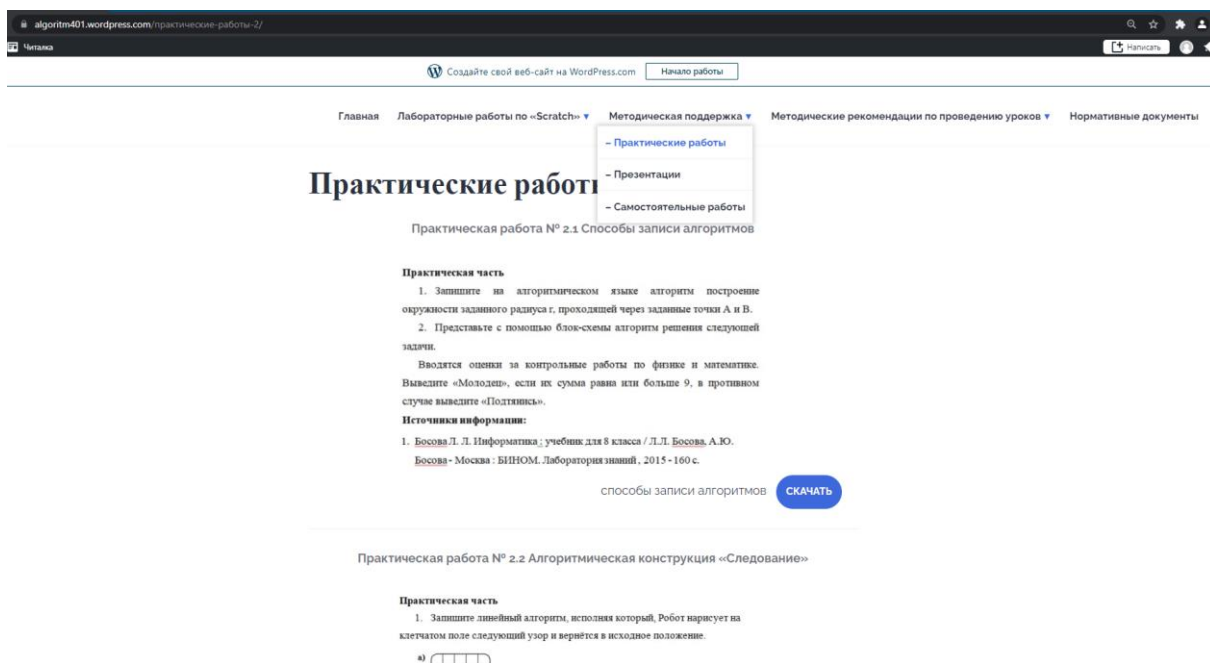


Рисунок 4- Страница «Практические работы»

На странице «Презентации» (рисунок 6) расположены презентации к каждому уроку.

Для проверки домашнего задания на странице «Самостоятельные работы» размещены приложения, тесты, кроссворд (рисунки 7).

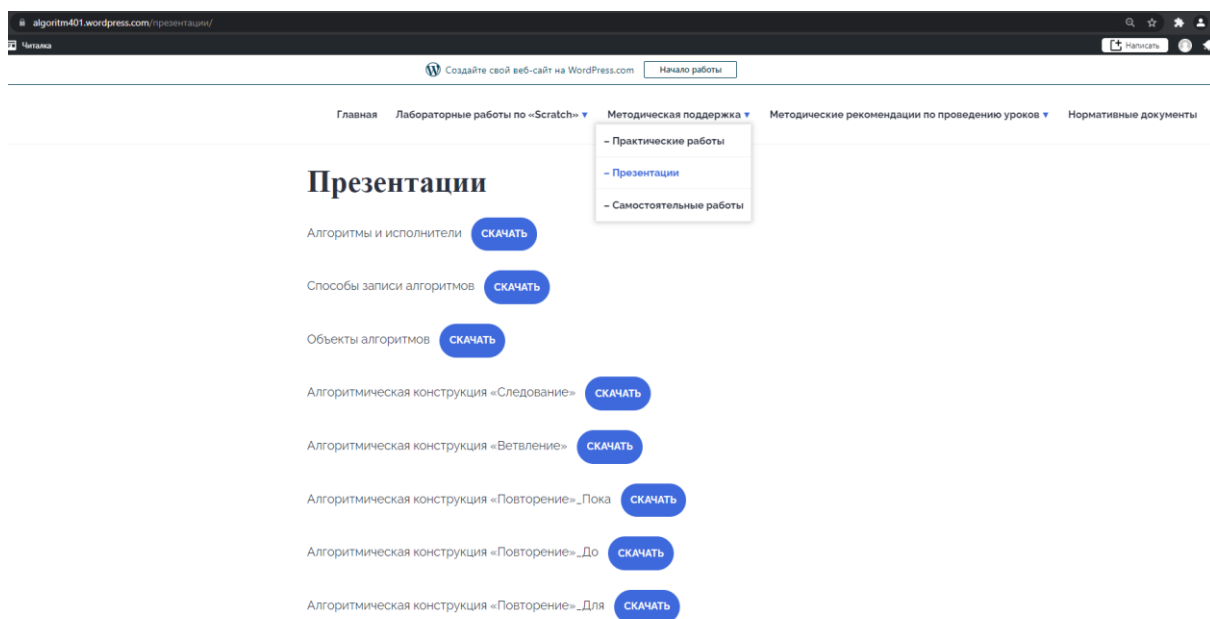
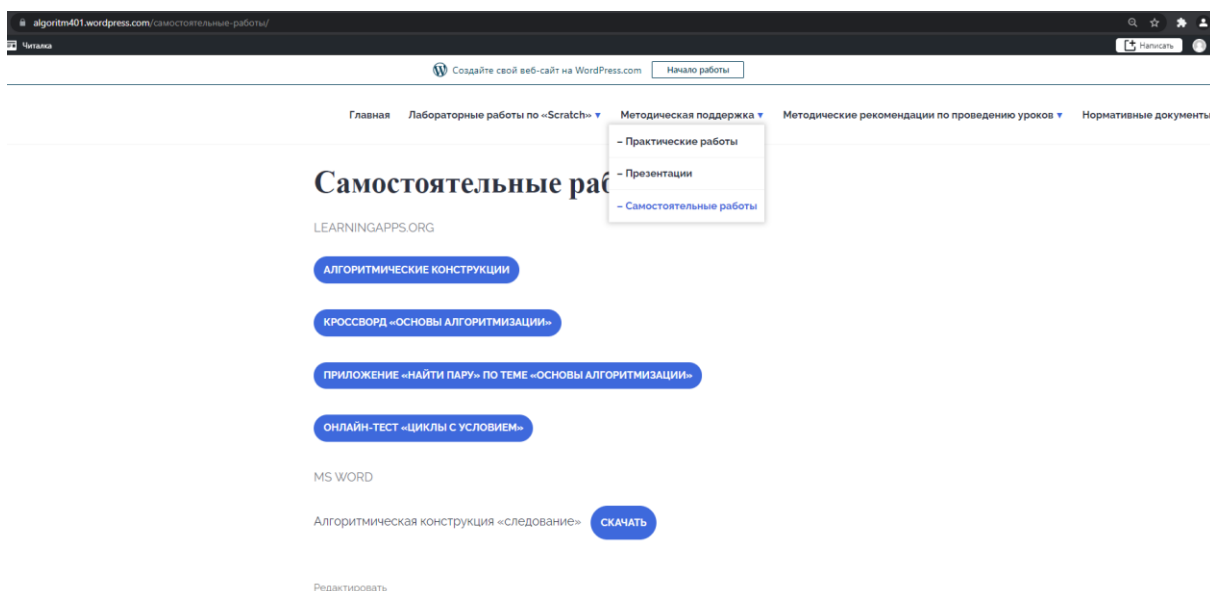


Рисунок 6 – Страница «Презентации»



### Рисунок 7-страница «Самостоятельные работы»

Самостоятельные работы разработаны с помощью:

- текстового редактора MS Word [18];
  - программы компьютерного тестирования знаний MyTestX [29];
  - сайта LearningApps.org — приложение Web 2.0 для поддержки обучения и процесса преподавания с помощью интерактивных модулей.
- Педагогическая апробация проводилась во время практики в лицее №57 г. Тольятти. Разработанный комплект электронных образовательных ресурсов использовался на уроках в 8 классах. Были изучены следующие темы: «Алгоритмы и исполнители», «Способы записи алгоритмов», «Объекты алгоритмов», «Алгоритмическая конструкция следование», «Алгоритмическая конструкция ветвление», «Алгоритмическая конструкция повторение», «Алгоритмы управления».

Апробация комплекта электронных ресурсов прошла успешно. Разработанные электронные образовательные ресурсы повысили интерес детей к теме «Основы алгоритмизации». Особое внимание ученики уделили интерактивным приложениям ресурса learningapps.org и тестам MyTestX.



Таким образом, на основе теоретических положений, изложенных в первой главе, во второй главе представлены методические рекомендации и программно- методическая поддержка раздела «Алгоритмизации». Методические рекомендации составлены на базе УМК Л.Л.Босовой.

В качестве программно-методической поддержке был разработан электронный практикум «Алгоритмизация». Для удобства работы с сайтом электронные ресурсы сгруппированы по типу документов (презентации, практические работы и т.д.).

Таким образом, во второй главе исследования был разработан и апробирован в 8 классах электронный практикум «Алгоритмизация» и подготовлена программно- методическая поддержка к нему.

### **2.3 Разработка электронного лабораторного практикума для проведения внеурочной деятельности «Весёлая алгоритмика»**

Для закрепления теоретических знаний по разделу «Алгоритмизация» и приобретению практических навыков работы с исполнителями школьникам будет полезно выполнить ниже описанные лабораторные работы в среде исполнителя Scratch:

Лабораторная работа №1. Знакомство со средой Scratch.

Внешний вид среды, поля. Алгоритмы. Линейные алгоритмы.

Цель: ввести понятия линейного алгоритма, работы с линейными алгоритмами в среде Scratch.

Теоретические сведения

Scratch в переводе с английского обозначает «царапать» — звуковой эффект, получаемый ручным продёргиванием звуковой дорожки (взад-вперёд, либо назад или вперёд), записанной на виниловую грампластинку или магнитную ленту, во время воспроизведения на проигрывателе или катушечном (бобинном) магнитофоне.

Спрайты это герои программы (объекты) — люди, животные, фантастические существа или предметы. Все действия в программе происходят со спрайтами. Основным героям программы является кот «Царапка», его имя и дало название программе.

На Scratch компьютерные программы создаются практически как конструктор: на экране собираются графические блоки, которые управляют анимированными героями. Таким образом можно создать историю или игру и контролировать поведение разных героев, соединяя графические блоки. Точно так же как из конструктора LEGO собирают различные конструкции, на Scratch можно создавать интерактивные истории и игры [6].

Меню программы. Все команды в Scratch выглядят как цветные блоки. Представь, что перед тобой 10 разноцветных ящичков. Из каждого ящичка ты вынимаешь нужную команду-блок и перетаскиваешь в область скриптов.

Таблица 11 – Команды Scratch

Группа	Цвет	Примечание
Движение	синий	управляют движением спрайта (не применяется в фоне)
Внешний вид	фиолетовый	управляют внешностью спрайта
Звук	розовый	управляют звуком спрайта
События	жёлтый	проверка наличия событий, отправка сигналов ко всем спрайтам

Управление	оранжевый	управляющие конструкции, заголовки обработчиков событий
Сенсоры	голубой	опрос устройств ввода, таймер и имя участника
Операторы	зелёный	арифметико-логические операции
Переменные	красный	раздел для управления переменными и списками
Другие блоки	светло-розовый	преобразуют комбинацию блоков в один блок, что упрощает код, или же упрощает создания своего личного блока

### Алгоритмы

В информатике вместо терминов сценарий и скрипт принято использовать - алгоритм. Термин алгоритм является одним из основных в информатике.

алгоритм=сценарий=скрипт

Перейти дорогу, сходить в магазин, написать диктант, решить математическую задачу - в основе всех этих форм поведения лежит точный порядок действий, выполнение которого гарантирует успех.

Алгоритм — это точный порядок действий объекта для достижения успеха. Действия в алгоритмах называют командами или шагами.

Термин алгоритм происходит от латинского написания имени среднеазиатского математика IX столетия Мухамеда ал-Хорезми. В первой половине XII века его книга «Об индийском счёте» в латинском переводе попала в Европу под названием «Algorithmi de numero Indorum». Правила выполнения арифметических действий в Европе стали называть алгоритмами (от латинского написания имени ал-Хорезми). Постепенно значение слова расширилось, и его стали применять не только к арифметическим, но и к другим действиям.

В октябре 1979 г. состоялся Международный симпозиум, посвященный 1100-летию термина “алгоритм”. В день открытия симпозиума заложили памятник ал-Хорезми.

Линейный алгоритм или следование - это тип алгоритма, в котором последовательность действий не меняется в его процессе выполнения.

Ход работы

1. Установка программы в системе Windows:

1. Скачайте нужную версию программы по ссылке: <http://scratch.mit.edu/download>

Дважды кликните левой кнопкой мышки по установочному файлу. Пройдите 5 шагов до конца установки.

1.1. Шаг первый - нажмите Next:

1.2. Шаг второй (программа сама предложит установить себя на диск C в папку по умолчанию, ничего менять не нужно, если вы не хотите поменять место установки программы) - просто нажмите Next:

1.3. Шаг третий (программа создаст свой раздел в меню установленных на компьютере программ) - ничего менять не нужно просто нажмите Install:

1.4. Шаг четвертый - нажмите Next:

1.5. Шаг пятый (программа предложит запустить Скретч и создать ярлык для быстрого запуска на рабочем столе) - нажмите Finish

Если вы все сделали правильно, запустится установленная программа Scratch.

Задание 1.

Попробуйте нарисовать в форме блок-схемы алгоритм действий скрипта изображённого на рисунке 8:

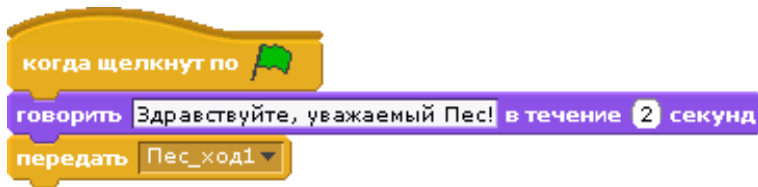



Рисунок 8- Скрипт линейного алгоритма.

Задание 2. Диалог кота и собаки.

Попробуйте собрать эти сценарии. Для начала вам нужно добавить спрайт Пес. Как это можно сделать? Нажмите на каждый спрайт, чтобы увидеть, что Кот и Пес имеют совершенно независимые собственные области скриптов. Как только вы закончите, нажмите зеленый флаг, чтобы запустить свой проект: .

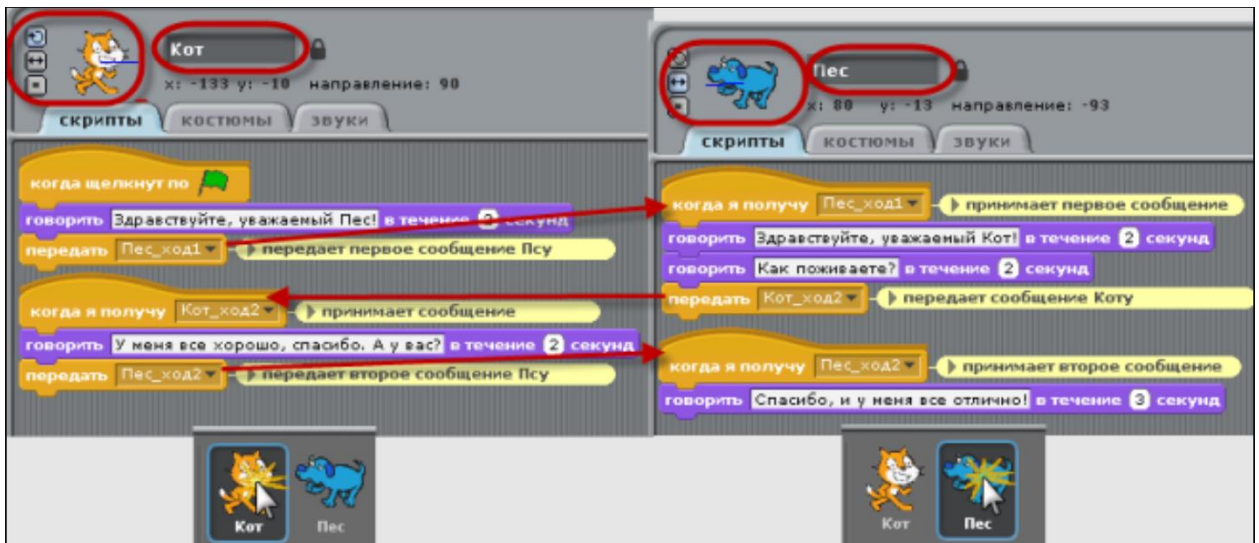


Рисунок 9- Линейный алгоритм диалога кота и собаки.

1. Попробуйте изменить проект так, чтобы каждый персонаж сказал дополнительные фразы.

2. У спрайта Кот есть звук Мяу. Если вы сможете, попытайтесь включить игру звука в поведение Кота, вам будут полезны эти блоки из категории звук:

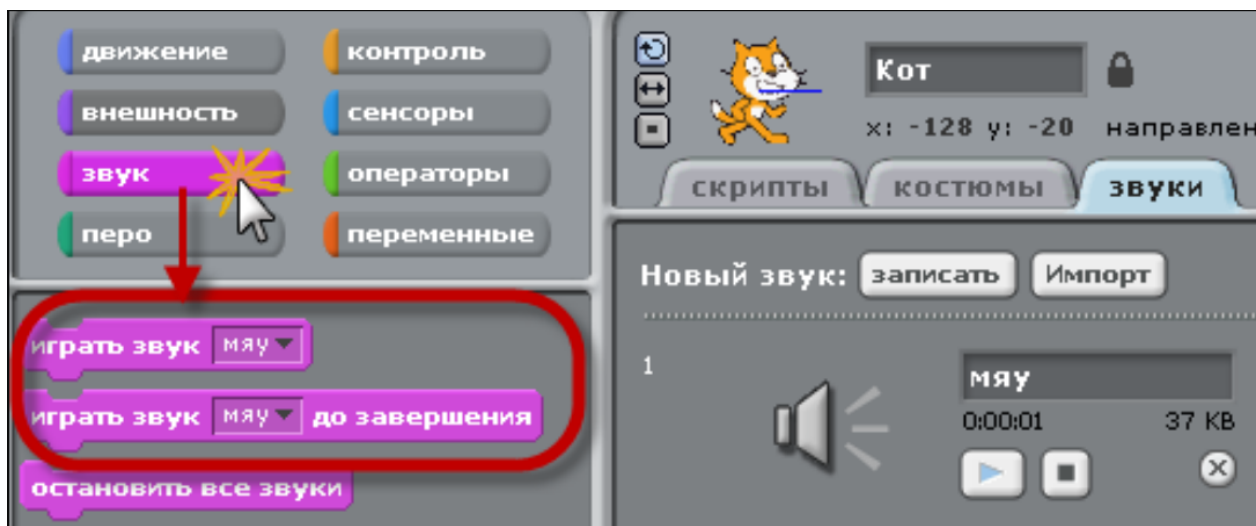


Рисунок 10-Добавление звука в скрипт.

3. Сохраните проект под именем Кот\_и\_Пес удобным для вас способом

### Лабораторная работа №2. Ветвление

Цель: научиться составлять разветвляющиеся алгоритм для исполнителя.

#### Теоретические сведения

Ветвление - одна из трех (наряду с линейным выполнением команд и циклом) базовых конструкций алгоритмов (рисунок 11).

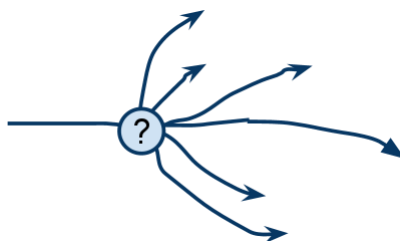


Рисунок 11-Разветвляющаяся структура.

Все языки программирования имеют специальные операторы (команды) для того, чтобы программа могла разветвляться в зависимости от того, выполняется указанное условие или нет. На блок-схеме ветвление показывается с помощью блока ромба, в который «входит» одна стрелка, а

выходят две «стрелки». Принято блок «ромб» называть блоком условия, а оператор ветвления – условным оператором. В категории блоков контроля есть 3 условных оператора: 2 С-блока и 1 стек (Рисунок 12):



Рисунок 12- Условные операторы в Scratch.

Блоки ветвления применяется, когда выполнение некоторого набора команд зависит от выполнения заданного условия.

Неполная форма ветвления изображена на рисунке №6 (условный оператор с одной ветвью).

Мы уже знакомы с логическими данными и выражениями. Именно они используются для организации ветвлений.

Условный оператор разрешает выполнение определенных команд при условии, что некоторое логическое выражение (условие) принимает значение истина (true).

В большинстве языков программирования условный оператор начинается с ключевого слова ЕСЛИ (англ. if).

Условный оператор с одной ветвью - неполная форма (рисунок 13)

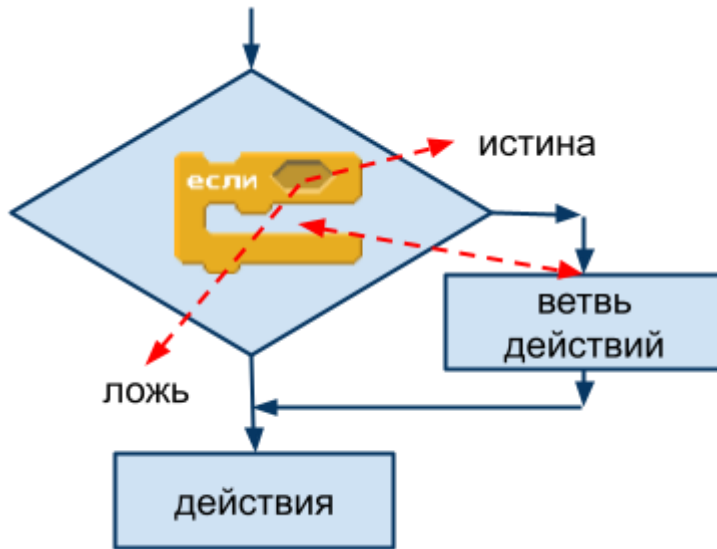


Рисунок 13-Условный оператор с одной ветвью.

При выполнении этого оператора вычисляется условие, и если оно истинно, то выполняются команды, находящиеся внутри с-блока, иначе выполнение программы продолжается со следующего за условным оператором блока (Рис.14).



Рисунок 14-Пример неполного ветвления в Scratch.

Полное ветвление изображена на рисунке 15 (условный оператор с двумя ветвями).



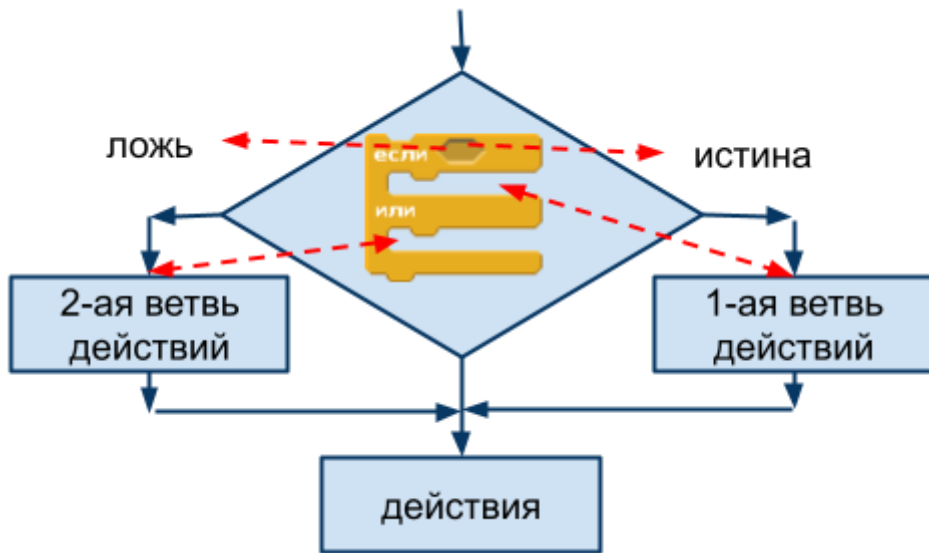


Рисунок 15-Условный оператор с двумя ветвями.

При выполнении этого оператора вычисляется условие, и если оно истинно, то выполняются команды, которые находятся внутри ветви ЕСЛИ, в противном случае выполняются блоки внутри ветки ИЛИ (рисунок 16).

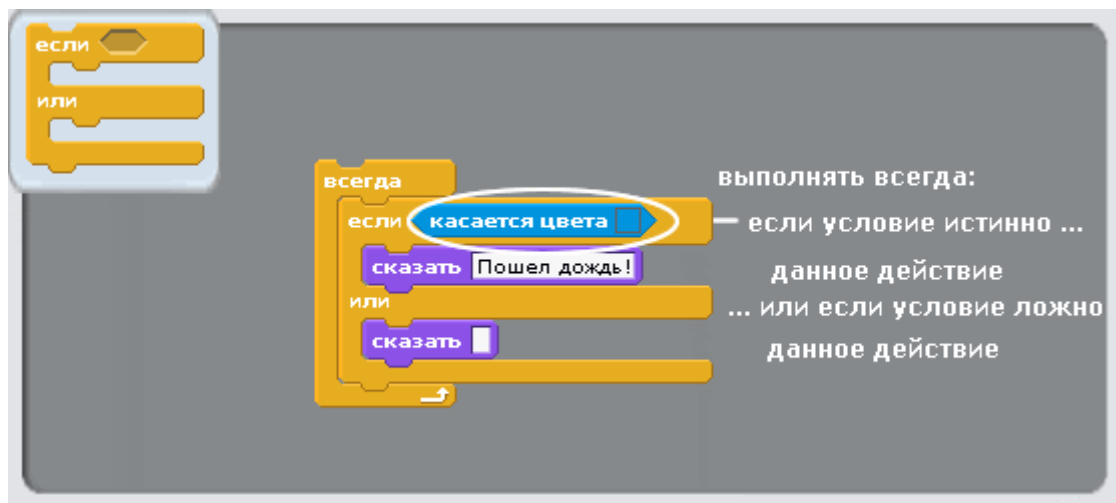


Рисунок 16-Реализация полного ветвления

В Scratch нет ограничений на вложение условных операторов друг в друга. Можно собрать ветвление любой сложности (рисунок 17).



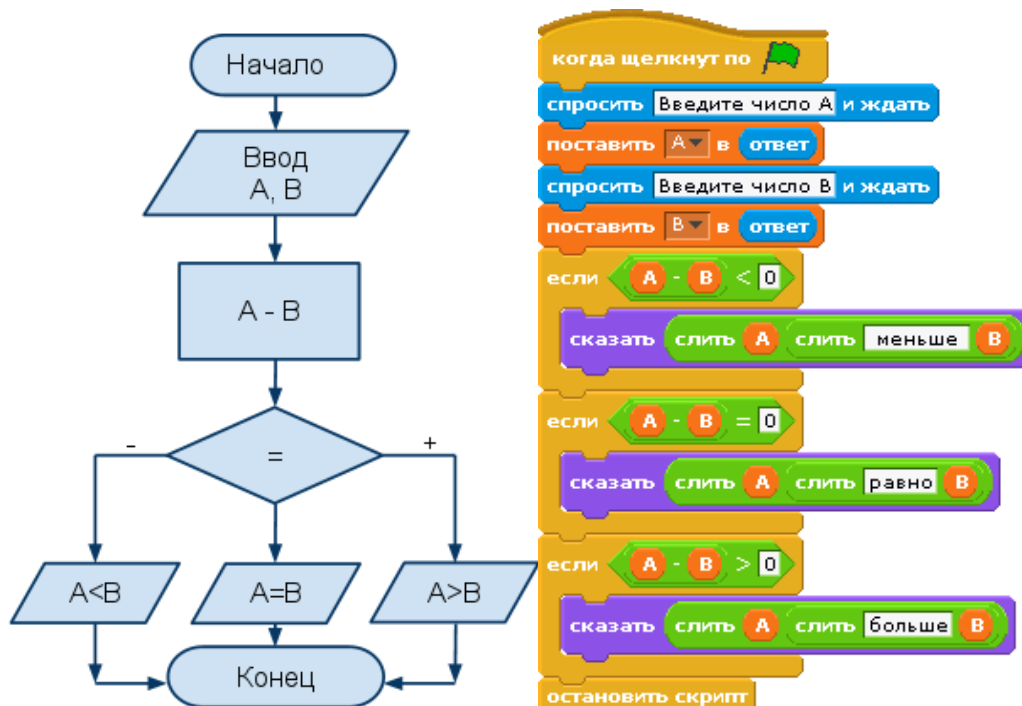
Рисунок 17- Блоки условных операторов можно вкладывать друг в друга

### Задание 1.

Алгоритм нахождения большего из двух заданных чисел

1. Создайте две переменные А и В.

Соберите скрипт по образцу (рисунок 18).



## Рисунок 18-Реализация алгоритма

3. Проверьте его работу.
4. Сохраните проект под Вашем именем.

Задание 2. Использование полного и неполного ветвления

Составьте алгоритм в виде блок-схемы и в виде скрипта.

Обязательно вычитать из числа  $A$  число  $B$ ? Нет. Использование арифметического оператора оправдано, если нужно не только сравнить величины, но и выяснить, насколько они отличаются друг от друга. Для сравнения величин можно воспользоваться логическими блоками непосредственно.

1. Упростите логические условия в блоках ЕСЛИ.
2. Упростите программу, добавив в нее полный условный оператор.
3. Сохраните проект с внесенными изменениями.

Проверьте себя по образцу представленном на рисунке 19.

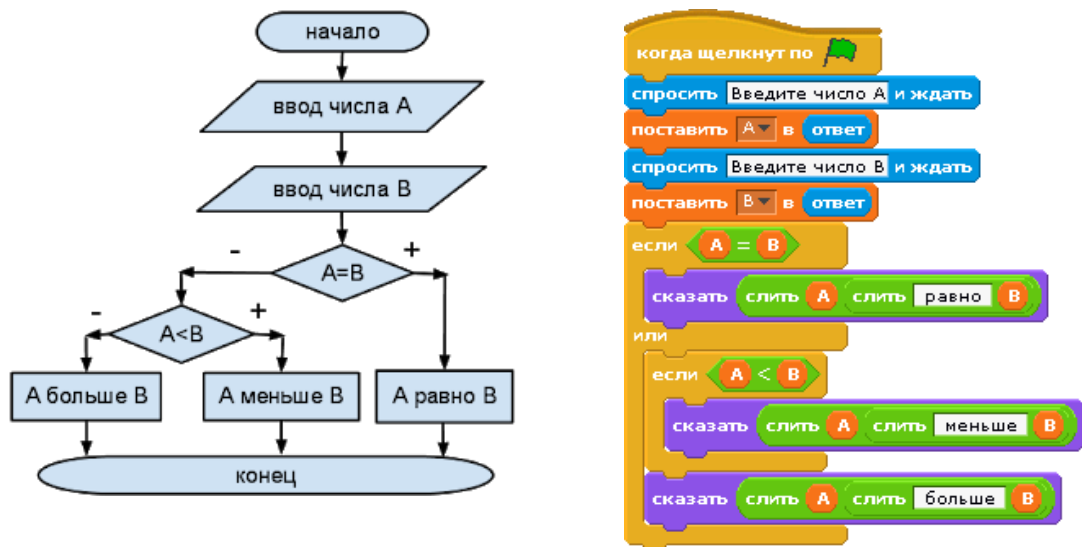


Рисунок 19- Реализация алгоритма нахождения большего из двух заданных чисел с использованием полного и неполного ветвления

Дополнительные вопросы:

Сколько условных операторов в языке Скретч?

Сколько условных операторов можно вставить во внутрь одного условного оператора: а) один; б) не более двух; в) любое количество; г) вообще нельзя.

Можно ли использовать условный оператор без логических репортеров? Почему?

Почему во всех языках программирования есть условные операторы?

Приходилось ли вам когда-нибудь делать выбор? Приведите примеры.

Задание 3. Создайте скрипт игры «Угадай число»

Игрок должен угадать случайное число, "задуманное" компьютером, в диапазоне от 0 до 100.

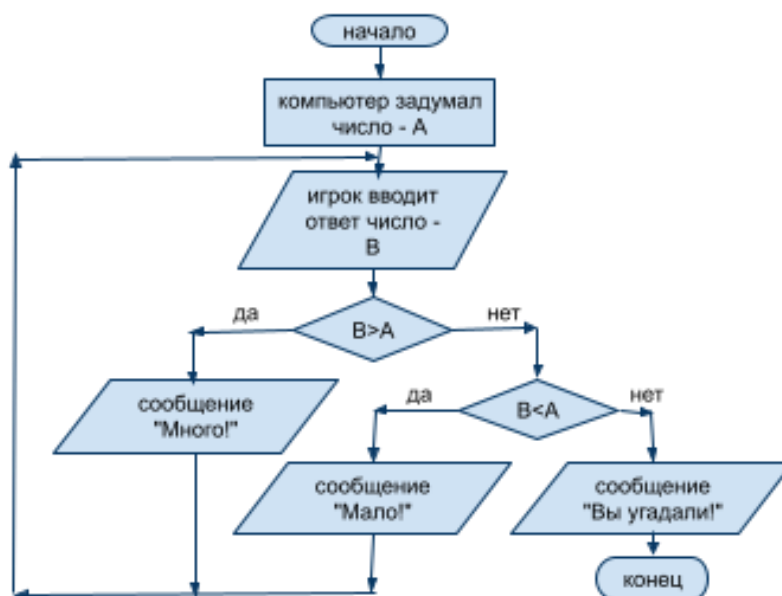


Рисунок 20-Алгоритм игры «Угадай число»

Начните новый проект.

Создайте глобальную переменную «число».

Соберите скрипт игры (рисунок 21)



Рисунок 21-Скрипт игры «Угадай число».

4. Поиграйте.

5. Сохраните проект с именем «Угадай число».

Дополнительные вопросы:

Как можно сделать игру интереснее и зрелищнее?

Попробуйте внести изменения и сохранить свой вариант проекта.

Лабораторная работа №3. Циклические алгоритмы. Цикл с параметром.

Цель: знакомство с алгоритмической структурой «цикл со счетчиком», ее реализацией в среде Scratch.

Теоретические сведения

При составлении алгоритмов довольно часто встречаются случаи, когда некоторую последовательность команд нужно выполнить несколько

раз подряд. Для упрощения записи алгоритма в таких случаях можно использовать специальную конструкцию повторения.

Цикл может быть с известным числом повторений и с неизвестным числом повторений (с предусловием и с постусловием). В среде программирования Scratch цикл с известным числом повторений реализуется с помощью команды «повторить». Описание действий, повторяющихся в цикле, называется телом цикла.

### Задание 1.

Требуется составить программу для графического исполнителя рисования «звездочки»:

Пусть для определенности исполнитель находится в точке (30; 0) и смотрит влево.

Алгоритм рисования «звездочки» такой: нарисовать прямую, повернуться на 144 градуса, опять нарисовать прямую, опять повернуться и так 5 раз (рисунок 22).

Блок – схема алгоритма рисования «звездочки»:

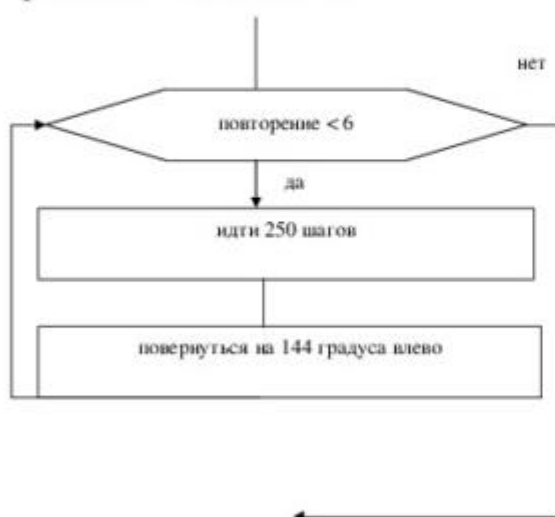


Рисунок 22- Алгоритм рисования звезды

## Задание 2.

Составить программу для исполнителя рисования правильного треугольника (рисунок 23):



Рисунок 23-Скрипт рисования треугольника

## Задание 3.

Составить программу для исполнителя рисования четырех разноцветных звездочек (рисунок 24):

В алгоритме рисования четырех звездочек используется алгоритм рисования звездочки. Алгоритм рисования звездочки необходимо повторить 4 раза. Получается, что в теле цикла нужно использовать цикл, который называется вложенным циклом.

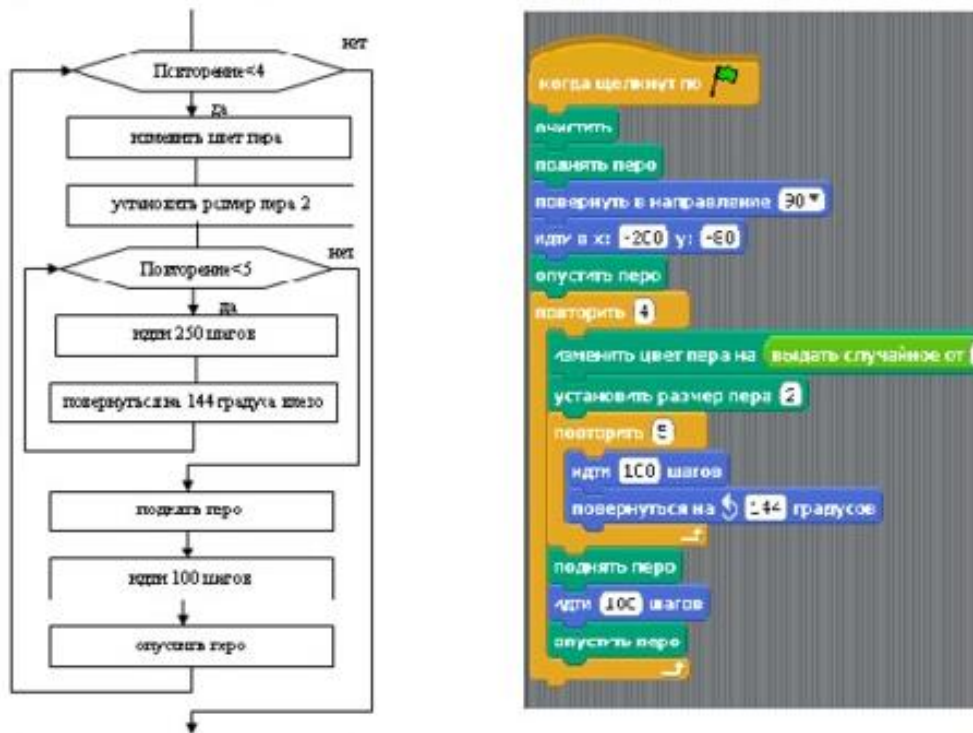


Рисунок 24- Алгоритм рисования четырёх звездочек

#### Лабораторная работа №4. Циклический алгоритм с предусловием

Цель работы: овладеть практическими навыками разработки и анализа алгоритмов циклической структуры с предусловием и с постусловием

#### Теоретические сведения

Как быть, если число повторений цикла неизвестно как, например, в игре «Угадай число»? В этом случае применяют разновидность цикла с предусловием (рисунок 25) или постусловием (рисунок 26).



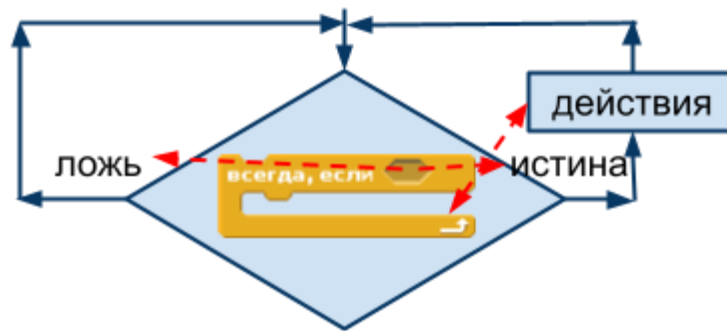


Рисунок 25- Блок-схема реализации цикла с предусловием

В этой разновидности цикла условие проверяется перед выполнением тела цикла, поэтому цикл не будет выполнен ни разу, если указанное условие не наступит. Вы не получите конфет, не выполнив домашнее задание, если такое условие поставлено взрослыми.

Блок «всегда если» находится в состоянии ожидания и запускает тело цикла в тот момент, когда соблюдается указанное условие. Цикл будет повторяться до тех пор, пока проверка условия будет давать результат истина. Если условие ложно, действия внутри цикла не выполняются.

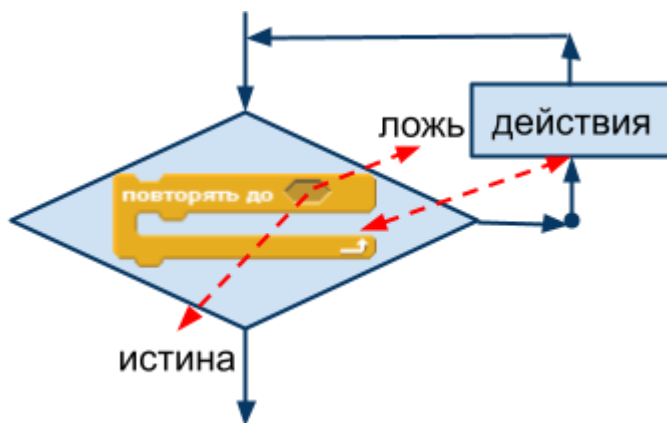


Рисунок 26-Блок-схема реализации цикла с постусловием

В трактовке условия этого цикла в разных языках есть отличия. Например, в Паскале условие такого цикла работает как условие

выхода - цикл завершается, когда условие становится истинным: вы перестаете делать домашнее задание, как только получаете конфеты.

В языке Си иначе - как условие продолжения - цикл выполняется, пока условие истинно и завершается, когда оно становится ошибочным: вы делаете домашнее задание до тех пор, пока вам дают конфеты.

Пример.

Вычислить произведение чисел от 1 до 5 используя различные варианты цикла

Математическая модель:

$$P = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

Составим алгоритм в виде блок-схемы.

Для проверки правильности алгоритма заполним трассировочную таблицу (рисунок 27).

Проверка условия происходит в несколько шагов: проверка условия и выполнение команд на одной из ветвей. Поэтому в трассировочной таблице записываются не команды алгоритма, а отдельные операции, выполняемые компьютером на каждом шаге.

Шаг первый:  $P$  присваивается значение один.

Шаг второй:  $i$  присваивается значение один.

Шаг третий: при  $i$  равном единице проверяем условие один меньше или равен пяти, да, условие истинно, значит  $P$  присваивается значение один умноженное на один, будет два. Для  $i$ : один плюс один, будет два.

Шаг четвертый: при  $i$  равном двум проверяем условие два меньше или равен пяти, да, условие истинно, значит  $P$  присваивается значение 2 умноженное на один, будет 2. Для  $i$ : два плюс один, будет три.

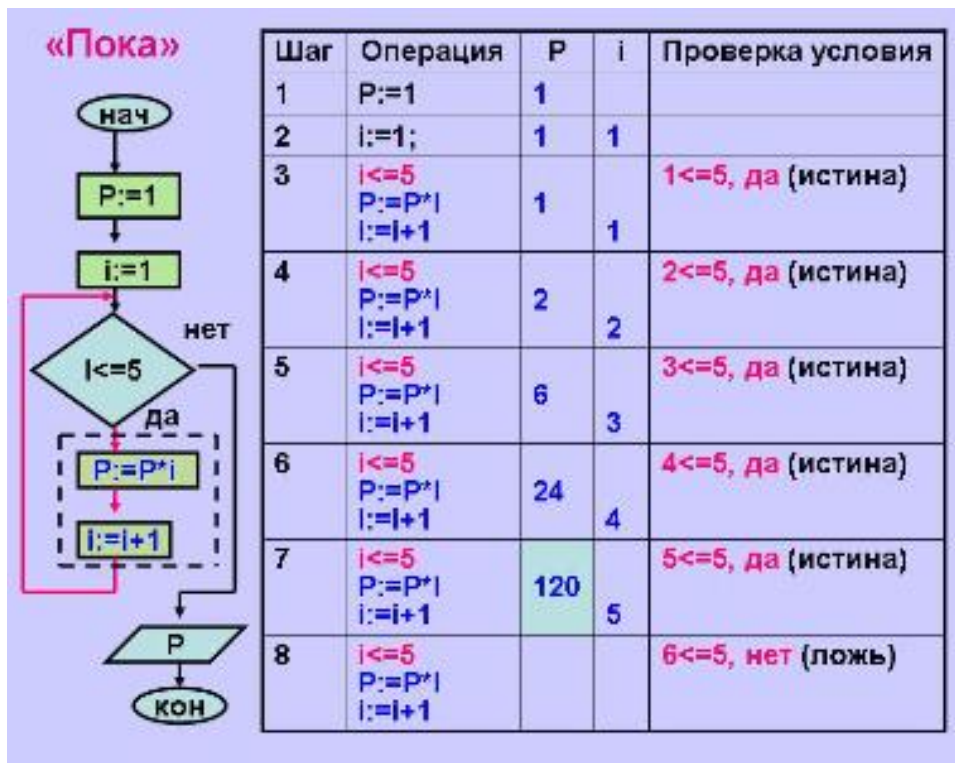


Рисунок 27 – Алгоритм решения задачи с помощью цикла с предусловием и трассировочная таблица

Проверка условия происходит в несколько шагов: проверка условия и выполнение команд на одной из ветвей. Поэтому в трассировочной таблице записываются не команды алгоритма, а отдельные операции, выполняемые компьютером на каждом шаге.

Шаг первый: P присваивается значение один.

Шаг второй: i присваивается значение один.

Шаг третий: при i равном единице проверяем условие один меньше или равен пяти, да, условие истинно, значит P присваивается значение один умноженное на один, будет два. Для i: один плюс один, будет два.

Шаг четвертый: при i равном двум проверяем условие два меньше или равен пяти, да, условие истинно, значит P присваивается значение 2 умноженное на один, будет 2. Для i: два плюс один, будет три.

Шаг пятый: при  $i$  равном трем проверяем условие три меньше или равен пяти, да, условие истинно, значит  $P$  присваивается значение два умноженное на три, будет шесть. Для  $i$ : три плюс один, будет четыре.

Шаг шестой: при  $i$  равном четырем проверяем условие четыре меньше или равен пяти, да, условие истинно, значит  $P$  присваивается значение шесть умноженное на четыре, будет двадцать четыре. Для  $i$ : четыре плюс один, будет пять.

Шаг седьмой: при  $i$  равном пяти проверяем условие пять меньше или равен пяти, да, условие истинно, значит  $P$  присваивается значение двадцать четыре умноженное на пять, будет сто двадцать. Для  $i$ : пять плюс один, будет шесть.

Шаг восьмой: при  $i$  равном шести проверяем условие шесть меньше или равен пяти, нет, условие ложно, тогда мы выходим из цикла, а в результате получаем последнее значение равное ста двадцати.

Для цикла с постусловием построим блок-схему и трассировочную таблицу.

В результате получаем последнее значение равное ста двадцати на седьмом шаге (рисунок 28).

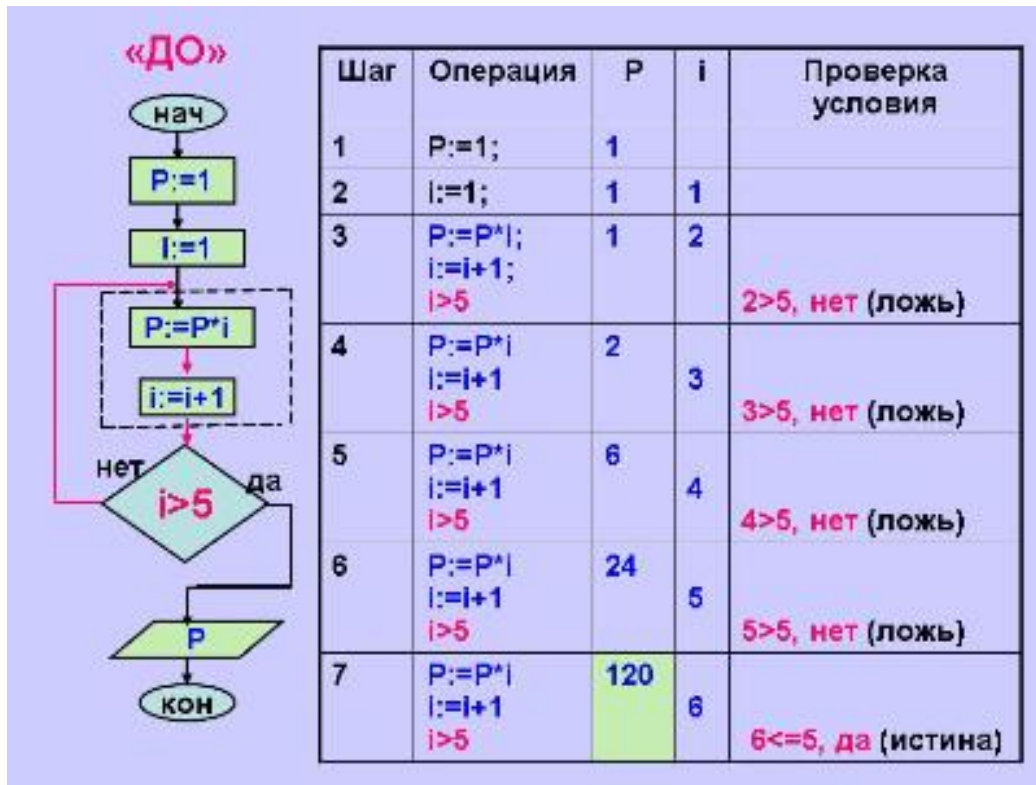


Рисунок 28 – Алгоритм решения задачи с помощью цикла с постусловием и трассировочная таблица

И для Цикла с параметром построим блок-схему и трассировочную таблицу.

В результате получаем последнее значение равное ста двадцати на шестом шаге (рисунок 29).

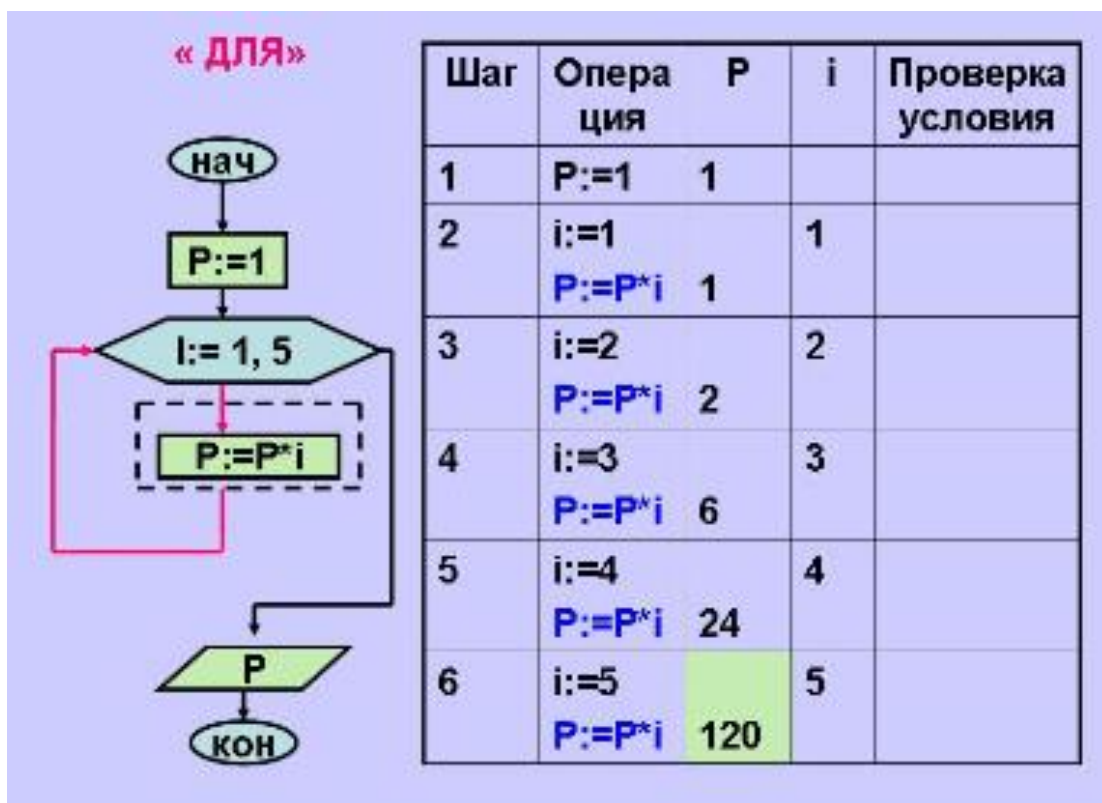


Рисунок 29 – Алгоритм решения задачи с помощью цикла с параметром и трассировочная таблица

Вопросы:

1. Что называется циклом?
2. Какие существуют виды циклов?
3. В чем отличие цикла с предусловием от цикла с постусловием?
4. В каком из циклов условие выполнится хотя бы один раз?
5. В каком из циклов «тело цикла» может не выполниться ни разу?
6. В каком из циклов заранее известен шаг?

### Лабораторная работа №5. Музыка чисел.

Мы живем в океане звуковых волн, источником которых являются явления природы, различные механизмы и сами люди. Человек способен говорить высоким голосом, похожим на комариный писк, и низким, подобным раскатам грома, говорить очень громко и шепотом, в быстром и

медленном темпе. А некоторые люди умеют изменять тембр, подражая чужим голосам. В музыке важны не только эти физические характеристики звука, но и порядок звуков во времени, ритм.

Основой музыкальной системы является высота звука. Она определяется количеством колебаний источника звука в секунду. Чем больше колебаний в секунду, например, гитарной струны, тем выше частота и высота создаваемого ею звука, и наоборот.

### Влияние темпа на длительность нот

Пронаблюдайте за изменением длительности нот при изменении темпа. Установите максимальную и минимальную величину темпа в Скретч.

1. Соберите скрипт по образцу (рисунок 30).

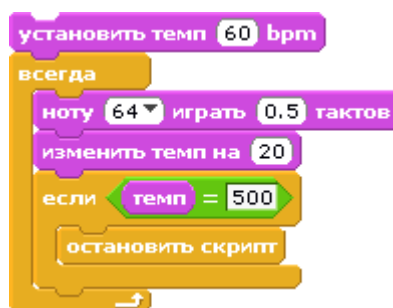


Рисунок 30- Программирование темпа

2. Запустите скрипт на выполнение, наблюдая за изменениями длительности звучания нот в каждом шаге цикла при ускорении темпа.

3. Выясните опытным путем, может ли темп быть больше 500 ударов за минуту?

4. Определите минимально возможное значение темпа.

5. Сохраните проект.

### Линейный алгоритм гаммы

В музыке гамма - это постепенно восходящее или нисходящее мелодичное движение степеней звукоряда (нот) в пределах какой-либо октавы. Сделаем программу, которая будет проигрывать гамму в пределах первой октавы на основе простого линейного алгоритма:

- 1) установить темп → 2) играть по очереди 12 нот от 60 до 71.
1. Создайте новый проект.
  2. Соберите скрипт по образцу.



Рисунок 30- Восходящая гамма в темпе 60 ударов на минуту

3. Сохраните проект.

Алгоритм с предусловием «Как по нотам», представлен на рисунке 31.



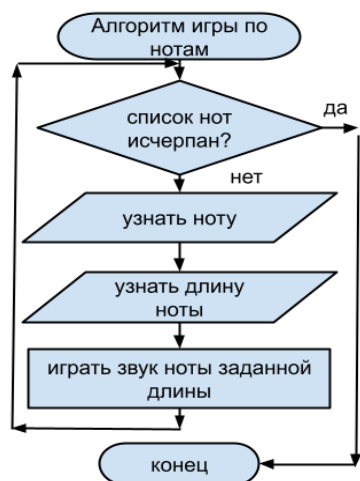


Рисунок 31-Алгоритм с предусловием «Как по нотам»

Чтобы реализовать такой алгоритм, нам понадобится блок цикла с числом повторений, которое равно количеству нот, список нот и список их длины, переменная цикл, которая будет считать уже сыгранные ноты и выбирать новые ноты и их длину (рисунок 32).

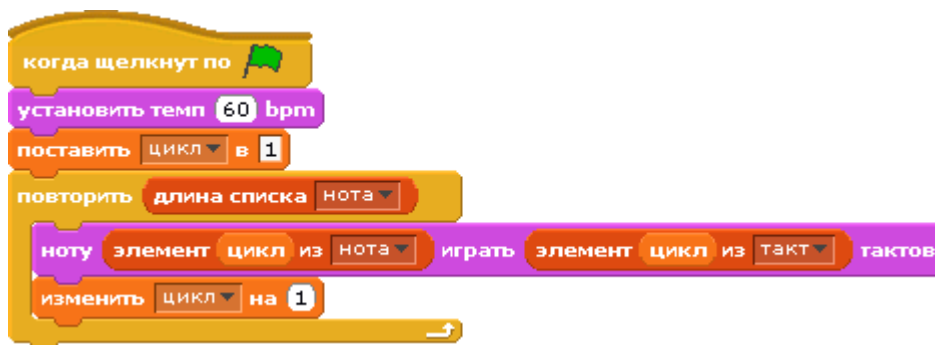


Рисунок 32-Скрипт игры по нотам

Поручить выполнение этого алгоритма можно любому объекту, например, сцене (рисунок 33).

1. Создайте новый проект.
2. Создайте переменную цикл.
3. Создайте списки нота и такт (Рис. 15.7), заполните их.

нота	такт
1 60	1 0.5
2 61	2 0.5
3 62	3 0.5
4 63	4 0.5
5 64	5 0.5
6 65	6 0.5
7 66	7 0.5
8 67	8 0.5
9 68	9 0.5
10 69	10 0.5
11 70	11 0.5
12 71	12 0.5

+ длина: 12    + длина: 12

Рисунок 33-Список нот первой октавы и список их длительности

4. Соберите скрипт игры по нотам (Рисунок 34)
5. Проверьте работу скрипта.
6. Подумайте, как добавить возможность изменять темп?
7. Сохраните проект.

Что случится, если прибавить следующие блоки?



Рисунок 34- Скрипт игры по нотам

### Музыкальный инструмент

Пришло время сделать из клавиатуры музыкальный инструмент. Если каждой из 12 нот назначить свою клавишу клавиатуры, мы получим музыкальную клавиатуру, на которой можно играть. Алгоритм очень простой:

- 1) когда нажата клавиша → 2) звучит ее нота.

Как добавить возможность извлечения звуков разных октав? Можно закрепить звуки других октав за другими клавишами или прибавить переключатели октав, например, при нажатии на числовую клавишу будут звучать ноты закрепленной октавы: 1 - субконтроктави, 2 - контроктавы, 3 - большой октавы, 4 - малой октавы, 5 - первой, 6 - второй, 7 - третьей, 8 - четвертой и 9 - пятой. Для этого понадобится переменная номера октавы, которую мы назовем #, и список значений октава. Первая октава станет точкой отсчета, потому ее значение равняется 0. Октавам слева присвоим отрицательные значения, а справа - положительные (Рисунок 35).

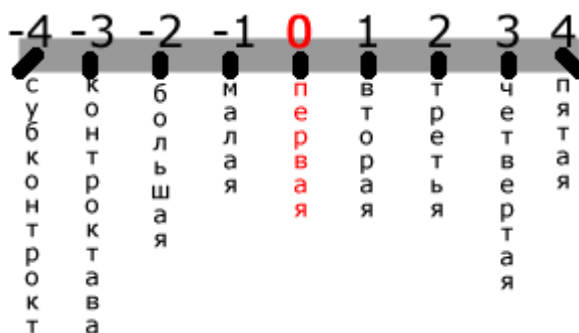


Рисунок 35- Числовая ось октав

Так как между одинаковыми нотами соседних октав всегда 12 единиц, используя одну простую формулу, мы сможем вычислить ноту любой октавы: нота X октавы N = нота X 1-й октавы + (12 \* значение октавы N) Например, если мы хотим сыграть ноту ДО большой октавы, нужно к 60 (ее значение в первой октаве) прибавить произведение 12 \* 2 = 24. Так как прибавление к положительному числу (60) отрицательного числа (- 24) приводит к вычитанию, мы получим число 36, это и есть нота ДО большой октавы. В Скретче эта формула выглядит так:

так: `60 + 12 * элемент # из октава`

1. Откройте проект.

2. Создайте список девяти значений октав с именем октава (рисунок 36).



Рисунок 36-Список значений октав

3. Создайте переменную с именем # для выбора элементов списка октава.

4. Переключение октав поручим сцене. Соберите для сцены девять скриптов для выбора октав после нажатия на числовую клавишу (рисунок 37)



Рисунок 37- Скрипты выбора октавы после нажатия на числовую клавишу клавиатуры

4. Сделайте 12 спрайтов для 12 нот, например, так (рисунок 38):



Рисунок 38- Образы спрайтов нот на сцене

Изображения нот можно для наглядности расположить на сцене в порядке роста высоты звуков (рисунок 39).

5. Используя схему клавиатуры (рисунок 40), соберите скрипты (рисунок 41) музыкальных клавиш для каждой ноты.

6.

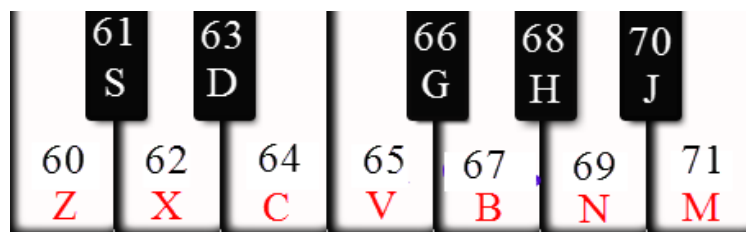


Рисунок 39- Модель виртуальной первой октавы

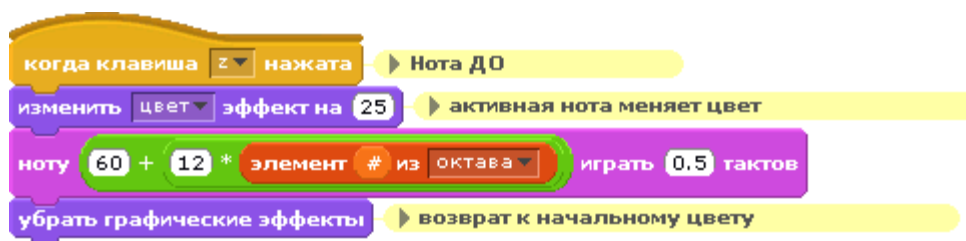


Рисунок 40- Скрипт музыкальной клавиши для ноты ДО (60)

7. Проверьте работу скрипта - при нажатии на запрограммированные клавиши должны звучать ноты. Поиграйте на музыкальной клавиатуре в разных октавах.
8. Сохраните проект.

Вопросы:

1. Какое физическое устройство внутри компьютера отвечает за создание звуков?
2. Какая нота установлена в Скретче по умолчанию?
3. Какой темп установлен в Скретче по умолчанию?
4. Чему по умолчанию равна величина такта?

5. Чему по умолчанию равна величина паузы?

## **ЗАКЛЮЧЕНИЕ**

В ходе исследования была достигнута цель, которая заключалась в разработке электронных образовательных ресурсов по изучению раздела «Алгоритмизация» в школьном профильном курсе информатики. С помощью разработанных методических рекомендаций по коммуникационным технологиям можно раскрыть содержание данной темы, более глубоко, организовать активную деятельность учащихся на компьютере, переключить учащихся с одного вида учебной деятельности на другой, способствуя тем самым развитию внимания и интереса к изучаемой проблеме. Средства коммуникационных технологий обучения способствуют восприятию, усвоению и систематизации учебного материала.

Разработанные материалы могут быть использованы на различных этапах: погружения в тему, закрепления и отработки практических навыков, на этапе контроля и мониторинга знаний и умений. А также в различных формах: фронтальной, индивидуальной, офф-лайн и онлайн, очно-урочной и дистанционной.

Таким образом, практическая значимость представленного в работе методического сопровождения темы «Алгоритмизация» очень высока. Цель исследования достигнута.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р 53620-2009 Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения. Москва : Стандратинформ, 2018г.
2. ГОСТ Р 55751-2013 Информационно-коммуникационные технологии в образовании. Электронные учебно-методические комплексы. Требования и характеристики.: от 01.01.2015.
3. ГОСТ Р 55750-2013 Информационно-коммуникационные технологии в образовании. Метаданные электронных образовательных ресурсов. Общие положения от 01.01.2015.
4. ГОСТ Р 52657-2006 Информационно-коммуникационные технологии в образовании. Образовательные интернет-порталы федерального уровня. Рубрикация информационных ресурсов от 30.06.2008
5. Федеральный Государственный образовательный стандарт основного общего образования, утвержден приказом Министерства образования и науки Российской Федерации от 17.05.2012 №413 / Министерства образования и науки Российской Федерации. – Москва 2012, URL: <https://base.garant.ru>
6. Примерная образовательная программа среднего общего образования [Электронный ресурс]: одобрена решением федерального учебно-методического объединения по общему образованию, протокол от 28.06.2016г. , №2 /16-3 / Министерство образования и науки Российской Федерации. – Москва: 2016г. - URL:[http://www.consultant.ru/document/cons\\_doc\\_LAW\\_161101/](http://www.consultant.ru/document/cons_doc_LAW_161101/)
7. БИНОМ. Лаборатория знаний [Электронный ресурс]. - Режим доступа: <http://www.lbz.ru/metodist/authors/informatika/>. - Загл. с экрана.



8. Босова Л.Л. Информатика 7-9 классы, методическое пособие / Л.Л.Босова., А.Ю. Босова. — М.: БИНОМ. Лаборатория знаний , 2016, — 464с.
9. Босова, Л.Л. Информатика. 7-9 классы [Текст]: методическое пособие / Л.Л. Босова, А.Ю. Босова. - М.: БИНОМ. Лаборатория знаний, 2015. - 472 с.
- 10.Босова, Л.Л. Информатика. 7-9 классы [Текст]: примерная рабочая программа / Л.Л. Босова, А.Ю. Босова. - М.: БИНОМ. Лаборатория знаний, 2016. - 30 с.
- 11.Босова, Л.Л. Информатика. 7-9 классы [Текст]: сборник задач и упражнений / Л.Л. Босова, А.Ю. Босова, Н.А. Аквилянов. - М.: БИНОМ. Лаборатория знаний, 2018. - 224 с.
- 12.Босова, Л.Л. Информатика. 8 класс [Текст]: рабочая тетрадь в 2 ч. Ч 1. / Л.Л. Босова, А.Ю. Босова. - 2-е изд., исправл.- М.: БИНОМ. Лаборатория знаний, 2017. - 88с.
- 13.Босова, Л.Л. Информатика. 8 класс [Текст]: рабочая тетрадь в 2 ч. Ч 2. / Л.Л. Босова, А.Ю. Босова. - 2-е изд., исправл.- М.: БИНОМ. Лаборатория знаний, 2017. - 88с.
- 14.Босова, Л.Л. Информатика. 8 класс [Текст]: самостоятельные и контрольные работы / Л.Л. Босова, А.Ю. Босова и др. - М.: БИНОМ. Лаборатория знаний, 2018. - 112 с.
- 15.Босова, Л.Л. Информатика. 8 класс [Текст]: учебник / Л.Л. Босова, А.Ю. Босова. - 2-е изд., исправл.- М.: БИНОМ. Лаборатория знаний, 2014. - 160с.
- 16.Векслер В. А. Эргономические требования к электронным образовательным ресурсам. [Электронный ресурс] Электронный научно-практический журнал «Психология, социология и педагогика». [сайт] URL:<http://psychology.snauka.ru/2015/05/4949>

17. Данилов Н.С. Планирование [Электронный ресурс]: Поурочное планирование по информатике. 2 – 11 классы. – Электронные данные. – 2009. – URL <http://dannik.ru/teaching/plan.php> (дата обращения: 2.02.2020).
18. Единая Коллекция цифровых образовательных ресурсов [Электронный ресурс] : сайт. – URL : <http://school-collection.edu.ru>
19. Единое окно доступа к образовательным ресурсам. – <http://window.edu.ru>.
20. Единые требования к электронным образовательным ресурсам. – М., 2011.
21. Информатика в школе [Электронный ресурс] : сайт. – URL : <http://www.infoschool.narod.ru>
22. Инфоурок [Электронный ресурс] : сайт. – URL : <https://infourok.ru/>
23. Исполнитель «Рисователь» [Электронный ресурс]. - Режим доступа: <https://www.lyceum62.ru/files/b5203dd4ebd1dcb93b996829944d38b8/>. - Загл. с экрана.
24. Князев С. Т. Методические инструкции по разработке электронных образовательных ресурсов. [Электронный ресурс] ФГАОУ ВПО «УрФУ имени первого Президента России Б. Н. Ельцина». 2011г. [Электронный ресурс] : сайт. - URL:<http://pandia.ru/text/77/339/92253.php>
25. Лапенко М. В. Научно-педагогические основания создания и использования электронных образовательных ресурсов информационной среды дистанционного обучения. [Электронный ресурс]. Автореферат диссертации на соискание ученой степени доктора педагогических наук [сайт] URL: <https://search.rsl.ru/ru/record/01005548320>

26. Лобачев, С. Основы разработки электронных образовательных ресурсов [Электронный ресурс]: учебный курс / С. Лобачев. - 2-е изд., исправ. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 189 с. - URL:  
<http://biblioclub.ru/index.php?page=book&id=429160>
27. Малев, В.В. Практикум по методике преподавания информатики [Электронный ресурс]: практикум / В.В. Малев, А.А. Малева. - Воронеж: ВГПУ, 2006. - 146 с. - URL:  
<http://biblioclub.ru/index.php?page=book&id=103304>
28. Морозова И. В. Классификация информационных электронных образовательных ресурсов. [Электронный ресурс]. Материалы IX Всероссийской научно-практической конференции «Применение информационно-коммуникационных технологий в образовании». «ИТО-МарийЭл», 2012. URL:  
<http://mari.ito.edu.ru/2012/section/191/97345/>
29. Образовательный портал «РЕШУ ОГЭ» [Электронный ресурс]. - Режим доступа: <https://inf-oge.sdangia.ru/test?theme=20>. - Загл. с экрана.
30. Осин А. В. ЭОР нового поколения: открытые образовательные модульные мультимедиа системы. [Электронный ресурс] : сайт. - URL:<https://studfiles.net/preview/5877379/>
31. Поляков К. Ю. Информатика. 7–9 классы : методическое пособие / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2016. — 80с.
32. Поляков К. Ю. Информатика. Углубленный уровень [Электронный ресурс] : учебник для 11 класса : в 2 ч. Ч. 2 / К. Ю. Поляков, Е. А. Еремин. – М. : БИНОМ. Лаборатория знаний, 2013. – 304 с. – URL:

[https://vk.com/doc67715714\\_489824519?hash=df8c12d29c94622867&dl=c45e6cf01ffa842e03](https://vk.com/doc67715714_489824519?hash=df8c12d29c94622867&dl=c45e6cf01ffa842e03)

33. Поляков, К.Ю. [Электронный ресурс]: Преподавание, наука и жизнь.  
- Режим доступа: <https://www.kpolyakov.spb.ru/school/test8a/a23.htm>. -  
Загл. с экрана.
34. Сайт «Программирование в среде Scratch» с учебно-методически  
материалом для проведения уроков в дистанционном формате:  
<http://w96308mt.beget.tech/>
35. Семакин И. Г. Информатика. Углубленный уровень [Электронный  
ресурс] : учебник для 11 класса : в 2 ч. Ч. 1 / И. Г. Семакин, Е. К.  
Хеннер, Л. В. Шестакова. – М. : БИНОМ. Лаборатория знаний, 2014.  
– 176 с. – URL:  
[https://vk.com/doc67715714\\_489017864?hash=8a70655fb95c2cc46a&dl=fa5ee96d06b4fe53](https://vk.com/doc67715714_489017864?hash=8a70655fb95c2cc46a&dl=fa5ee96d06b4fe53)
36. Семакин, И.Г. Информатика. 9 класс [Текст]: учебник / И.Г.  
Семакин, Л.А. Залогова. С.В. Русаков, Л.В. Шестакова. - 3-е изд. -  
М.: БИНОМ. Лаборатория знаний, 2015. - 200с.
37. Угринович, Н.Д. Информатика. 9 класа [Текст]: учебник / Н.Д.  
Угринович. - 4-е изд. - М.: БИНОМ. Лаборатория знаний. 2016. - 152  
с.