

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ  
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ЯРОСЛАВСКОЙ ОБЛАСТИ  
ЦЕНТР ДЕТСКО-ЮНОШЕСКОГО ТЕХНИЧЕСКОГО ТВОРЧЕСТВА

Утверждаю  
Директор И.А. ДОДЯКО  
*Иванов*  
«27» апреля 2020 г.



Согласовано:  
Методический совет  
от «27» апреля 2020 г.  
Протокол № 24/а-57

Техническая направленность

**Дополнительная общеобразовательная  
общеразвивающая программа**

**«ПРОГРАММИРОВАНИЕ»**

Возраст обучающихся: 14-18 лет

Срок реализации: 4 года

**Автор:**

Носок Сергей Иванович,  
педагог дополнительного  
образования высшей категории

Ярославль  
2020 год

## Содержание

<b>1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....</b>	<b>4</b>
<b>2. ЦЕЛИ И ЗАДАЧИ ОБУЧЕНИЯ .....</b>	<b>7</b>
<b>3. ОРГАНИЗАЦИЯ И ОСОБЕННОСТИ УЧЕБНОГО ПРОЦЕССА.....</b>	<b>8</b>
3.1. ОСОБЕННОСТИ ЗАНЯТИЙ ПРОГРАММИРОВАНИЕМ.....	8
3.2. НАБОР В ОБЪЕДИНЕНИЕ .....	8
3.3. ПРИМЕЧАНИЯ К УЧЕБНО-ТЕМАТИЧЕСКОМУ ПЛАНУ.....	9
3.4. УЧАСТИЕ В ОЛИМПИАДАХ ПО ИНФОРМАТИКЕ И ПРОГРАММИРОВАНИЮ .....	10
<b>4. СПОСОБЫ КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ .....</b>	<b>12</b>
<b>5. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.....</b>	<b>15</b>
5.1. ПРОГРАММА ПЕРВОГО ЭТАПА ОБУЧЕНИЯ .....	15
5.1.1. Системы счисления .....	15
5.1.2. Основы алгебры логики (Булева алгебра).....	16
5.1.3. Представление информации в компьютере (на примере основных встроенных типов языка Паскаль) .....	16
5.1.4. Алгоритмы. Начальный курс.....	17
5.1.5. Основы программирования на языке Паскаль.....	17
5.1.6. Массивы .....	18
5.1.7. Процедуры и функции.....	18
5.1.8. Указатели, работа с файлами .....	18
5.2. ПРОГРАММА ВТОРОГО ЭТАПА ОБУЧЕНИЯ.....	19
5.2.1. Сложность алгоритмов.....	19
5.2.2. Методы решения задач теоретического тура.....	19
5.2.3. Перебор с возвратом .....	20
5.2.4. Арифметические алгоритмы .....	20
5.2.5. Динамическое программирование.....	20
5.2.6. Линейные динамические структуры данных.....	21
5.3. ПРОГРАММА ТРЕТЬЕГО ЭТАПА ОБУЧЕНИЯ .....	21
5.3.1. Деревья .....	21
5.3.2. Графы. Базовые алгоритмы.....	21
5.3.3. Сортировки .....	22
5.3.4. Поиск стратегии в играх с полной информацией .....	22
5.4. Модули по выбору (ПРИМЕРЫ) .....	23
5.4.1. Работа в консоли.....	23
5.4.2. Графы. Усложненный курс .....	23
5.4.3. Вычислительная геометрия.....	23
5.4.4. Алгоритмы и структуры данных. Усложненный курс.....	23
5.4.5. Разработка программного проекта.....	24

5.5. Язык ПРОГРАММИРОВАНИЯ C.....	24
5.5.1. Введение в синтаксис языка C.....	24
5.5.2. Указатели, функции, передача параметров .....	25
5.6. Язык ПРОГРАММИРОВАНИЯ C++. ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ (ООП).....	25
5.6.1. Основные сведения по языку C++.....	25
5.6.2. Классы .....	26
5.6.3. Наследование .....	26
5.6.4. Стандартная библиотека шаблонов STL.....	26
5.7. Язык ПРОГРАММИРОВАНИЯ PYTHON.....	27
5.7.1. Введение .....	27
5.7.2. Структура программы и основные инструкции .....	27
5.8. ОСНОВЫ HTML-ВЁРСТКИ И ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVASCRIPT .....	28
5.8.1. Основы блочной вёрстки HTML документа .....	28
5.8.2. Основы языка JavaScript и динамического HTML .....	29
<b>6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ, ИНФОРМАЦИОННОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ.....</b>	<b>30</b>
<b>7. СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>32</b>
7.1. НОРМАТИВНО-ПРАВОВЫЕ ДОКУМЕНТЫ.....	32
7.2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И НАЧАЛЬНЫЙ КУРС ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПАСКАЛЬ .....	32
7.3. ПОДГОТОВКА К ОЛИМПИАДАМ ПО ИНФОРМАТИКЕ .....	33
7.4. ЯЗЫКИ ПРОГРАММИРОВАНИЯ C И C++ .....	33
7.5. HTML, CSS, JAVASCRIPT.....	34

# 1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа разработана в соответствии с Федеральным законом от 29.12.12 г. №273-ФЗ «Об образовании в Российской Федерации»; приказом Министерства образования и науки Российской Федерации (Минобрнауки России) от 09 ноября 2018 г. № 196 г. Москва «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»; Концепцией развития дополнительного образования детей в Российской Федерации, утвержденной распоряжением Правительства РФ от 4.09.2014 г. № 1726-р; санитарно-эпидемиологическими правилами и нормативами 2.4.4.3172-14 «Требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утв. Главным государственным санитарным врачом РФ от 04.07.2014 г. № 41); Государственной программой РФ «Развитие образования на 2013-2020 годы, утвержденной постановлением Правительства РФ от 15.04.2014 г. № 295; Стратегией инновационного развития Российской Федерации на период до 2020 года, утвержденным распоряжением Правительства РФ от 08.12.2011 года № 2227-р; Федеральной целевой программой развития образования на 2016-2020 годы, утвержденной Постановлением Правительства РФ от 23.05.2015 года № 497; Уставом ГОАУ ДО ЯО Центра детско-юношеского технического творчества.

Президент России В.В. Путин в послании Федеральному собранию в декабре 2016 года заявил: «Предлагаю запустить масштабную системную программу развития экономики нового технологического поколения — *цифровой экономики*. В ее реализации будем опираться на российские компании, научно-исследовательские и инжиниринговые центры страны. Это вопрос национальной безопасности, технологической независимости России, нашего общего будущего...»

Правительством РФ была разработана и в июле 2017 г. утверждена программа развития *цифровой экономики* до 2024 года, определяющая пять базовых направлений, среди которых указаны подготовка кадров и образование.

Среди технологических направлений *цифровой экономики* обычно выделяют интернет вещей, большие данные, искусственный интеллект и машинное обучение, блокчейн, кибербезопасность и сетевое системное администрирование. Все направления *цифровой экономики* требуют участия программистов, так как основаны на использовании компьютеров и соответствующего программного обеспечения.

Можно назвать умения и навыки, которые необходимы любому специалисту в условиях *цифровой экономики*. Это логическое мышление, коммуникационные навыки, навык решения задач, знакомство с областью компьютерных наук и др. Занятия программированием позволяют формировать и развивать эти навыки. В частности, программирование позволяет развивать способности к аналитическому мышлению.

Настоящая программа рассчитана на обучение школьников интересующихся компьютерной техникой и программированием на языках высокого уровня и ориентированных на

творческую работу в области решения программистских (логических, математических) задач и разработку прикладных программ.

Программа предполагает дать учащимся начальные теоретические знания и практические навыки в области программирования, а также выявить и развить их творческие способности, логическое мышление.

### **Планируемый результат**

По итогам первого этапа обучения учащиеся должны:

- иметь общее представление о различных системах счисления, уметь переводить числа из десятичной системы счисления в позиционную систему счисления с основанием отличным от 10, и обратно;
- знать основные логические операции, реализованные в языках программирования, уметь вычислять логические выражения, уметь составлять относительно несложные логические выражения для решения логических задач;
- иметь общее представление о кодировании основных встроенных типов данных языка Паскаль (чисел, символов, строк символов), знать их характеристики;
- знать базовые структурные алгоритмические блоки, уметь составить блок-схему алгоритма для решения относительно несложных задач — упражнений начального уровня;
- знать обобщённую структуру программы на языке программирования Паскаль, операторы языка и уметь записать решение задачи, представленное блок-схемой, в виде программы на Turbo Pascal;
- уметь самостоятельно решать относительно несложные задачи на языке Паскаль

При успешном освоении программы второго этапа (2–3 год) учащиеся должны:

- знать и уметь работать с основными структурными типами данных (массивами, файлами, множествами, типом запись);
- уметь использовать процедуры и функции, правильно выбирать способы передачи параметров в процедуры и функции;
- знать и уметь применить рекурсию, знать базовые методы решения задач, такие как перебор и методы сокращения перебора, вычисление и обработка последовательностей, уметь использовать вспомогательные структуры (типа таблиц с прямой адресацией), динамическое программирование и т.п., знать и уметь работать с базовыми абстрактными типами данных, такими как стек, очередь, куча и т.п.;
- быть готовыми к решению задач уровня городской олимпиады школьников;

В результате обучения на третьем этапе (3-й год и более) учащиеся должны:

- знать сложные методы решения задач: базовые алгоритмы на графах, комбинаторные алгоритмы, алгоритмы на строках и пр.;
- быть готовыми к решению задач олимпиад регионального и всероссийского уровня;
- знать основы программирования на языке C;

— знать основы объектно-ориентированного подхода к разработке программ, основы языка C++ и/или Python.

И еще один результат, который не планируется явно, но, как правило, достигается — это высокий балл при сдаче ЕГЭ по информатике. Несмотря на то, что основная подготовка к ЕГЭ идёт в школе, многие выпускники, занимавшиеся в объединении программирования, отмечают, что сдавали экзамен более уверенно, особенно в части C, и получили достаточно высокие баллы.

## 2. Цели и задачи обучения

**Цель программы** — создание такой среды для учащихся, в которой они могли бы, получив в руки инструмент — знание основ программирования на одном из языков программирования высокого уровня — развивать свои интеллектуальные способности и реализовать свои творческие замыслы, разрабатывая компьютерные программы, участвуя в конкурсах и олимпиадах по информатике и программированию.

### Задачи программы

#### Обучающие:

- Сформировать у учащихся навыки работы с информацией.
- Сформировать систему знаний, умений и навыков учащихся в области информатики, компьютерной техники и программирования.
- Обучить основам алгоритмизации и программирования на языках высокого уровня.
- Обучить практически решать задачи при помощи создания и оптимизации программ на языках высокого уровня.
- Сформировать навыки исследовательской деятельности.

#### Развивающие:

- Создавать условия для развития общих познавательных способностей учащихся: внимания, логического и абстрактного мышления, памяти.
- Развивать творческие способности учащихся.
- Развивать навыки самостоятельной работы и работы в коллективе.
- Развивать способности к самообразованию.

#### Воспитательные:

- Содействовать процессам самопознания и саморазвития личности.
- Создавать условия для самоопределения учащихся в профессиональном выборе.

## 3. Организация и особенности учебного процесса

### 3.1. Особенности занятий программированием

Главная особенность занятия программированием — высокая сложность предмета. Большая часть программы предназначена для школьников старших классов. Обучение на начальном этапе целесообразно начинать не ранее 8-го класса. Программа содержит большой объём теоретического материала, освоение которого требует от ученика значительных интеллектуальных усилий.

Для наиболее эффективной реализации программы необходимо:

- набирать для обучения детей, не просто интересующихся компьютерной техникой, а тех, кто ориентирован на занятия интеллектуальной деятельностью, и, может быть, уже имеет мотивацию к изучению программирования. Целесообразно, например, во время набора в объединение, проводить вступительное испытание, позволяющее выявить школьников, у которых есть способности и стремление к решению нестандартных логических, математических задач и задатки алгоритмического мышления;
- формировать группы учащихся по возрасту, по уровню общеобразовательной подготовки и уровню мотивации. Это позволит корректировать учебный план, повышая или понижая, при необходимости, уровень сложности при отборе учебного материала и составлении заданий;
- поддерживать высокий уровень мотивации для изучения программирования — решение сложных задач, проведение конкурсов и олимпиад программистов, изучение различных языков и технологий программирования, разработка «настоящих» программ, общение и обмен программистским опытом со сверстниками из других учреждений (при проведении массовых мероприятий) и т.п.

В летнее время, наиболее заинтересованным в совершенствовании навыков программирования учащимся, предоставляется возможность углубленного изучения отдельных тем.

### 3.2. Набор в объединение

С целью набора на основную часть программы наиболее мотивированных, готовых усваивать сложный материал, школьников (см. предыдущий раздел), в программу добавлен предварительный этап. Он проводится в очно-заочной форме.

Предварительный этап проводится в сентябре–начале октября, и включает в себя организационный сбор, несколько занятий и домашнее задание. Домашние задания и задания, решаемые в классе, представляют собой задачи на логику, смекалку, требующие обычного школьного уровня знаний математики. В ходе занятий проверяются наличие задатков алгоритмического мышления, базовых навыков работы с компьютером и навыки в

области коммуникационных технологий. Примеры упражнений, образцы заданий и другие материалы, относящиеся к проведению вступительных испытаний, приведены в Приложениях к программе.

Основные цели, которые достигаются в результате прохождения предварительного этапа:

- во-первых, достигается основная цель — выявляются и по итогам тестовых испытаний в первую очередь зачисляются на основную часть программы школьники, проявившие способности к занятиям программированием, имеющие достаточно уверенный уровень школьной подготовки (по математике), хороший уровень логического мышления и смекалки;
- во-вторых, происходит своевременный ранний отсев тех школьников, которые не готовы серьёзно заниматься программированием. Каждый из школьников, записавшийся на предварительный этап, в процессе вступительного испытания может понять (или даже почувствовать), какие качества потребуются от него в процессе обучения, какого рода деятельность ожидает его на занятиях, и самостоятельно может принять решение о том, соответствует ли такой род занятий его ожиданиям и предпочтениям.

### 3.3. Примечания к учебно-тематическому плану

Основная часть программы очень условно делится на три этапа. Наиболее чётко можно выделить первый этап — это период изучения теоретических основ программирования, начального изучения языка программирования Паскаль. Этот этап длится примерно полтора учебных года.

Начало второго этапа характеризуется переходом к изучению тем, связанных с подготовкой к участию в олимпиадах школьников по информатике и программированию школьного и муниципального уровня.

Третий этап представляет собой переход к рассмотрению алгоритмов и методов, необходимых для решения задач уровня региональных и всероссийских олимпиад по информатике, к изучению более мощных языков программирования (C/C++, Python). Третий этап характеризуется повышенной сложностью материала и, в основном, рассчитан на школьников 10–11-го класса, успешно прошедших подготовку на первых двух этапах и проявляющих повышенный интерес к изучаемому предмету.

Планирование учебного процесса — непрерывный процесс. Приведенный ниже учебно-тематический план может служить основой, но обязательно должен корректироваться в зависимости от уровня общеобразовательной подготовки учащихся, возраста.

Занятия по данной программе могут проводиться как в очной форме, так и с применением дистанционных технологий и (или) электронного обучения.

По данной программе в летний период может быть организована работа с обучающимися, которые проходят подготовку для участия в массовых мероприятиях, работают над

индивидуальными или командными проектами, а также проявляют особый интерес к выбранному виду деятельности.

Образовательный процесс по данной программе ведется в соответствии с годовым календарным учебным графиком на текущий учебный год, утвержденным приказом директора ГОАУ ДО ЯО ЦДЮТТ.

### **3.4. Участие в олимпиадах по информатике и программированию**

Большое место в учебном процессе занимает подготовка к олимпиадам по информатике и программированию. Начиная со второго года обучения, учащиеся объединения принимают участие в следующих ежегодных соревнованиях на уровне города и области:

Всероссийская олимпиада школьников по информатике.

- Школьный этап, участвуют все учащиеся 2-го и более годов обучения, время проведения: октябрь;
- Муниципальный этап, участвуют все учащиеся 2-го и более годов обучения, время проведения: конец ноября – начало декабря;
- Региональный этап, участвуют победители и призёры муниципального этапа, время проведения: январь;

Всероссийская командная олимпиада школьников по программированию. Региональный отборочный этап по Центральному федеральному округу. Формируются 1-2 команды по три человека. Время проведения: октябрь.

Дистанционная обучающая олимпиада по информатике (организаторы: ЦТИСО).

Практически все учащиеся второго и более годов обучения распределяются по командам-участницам (3-4 команды по 3-5 человек), возможно участие и в личном зачёте.

Время проведения: декабрь.

Открытый областной командный турнир по программированию (организаторы: ГОАУ ЯО ЦДЮТТ и ЯГПУ им. К.Д. Ушинского). Формируются 3-4 команды по 3 человека.

Время проведения: апрель.

В связи с участием в вышеперечисленных олимпиадах, кроме изучения теоретических тем, указанных ниже в учебно-тематическом плане, в учебном процессе большое место уделяется практической подготовке, которая включает в себя:

разбор теоретических заданий олимпиад прошлых лет;

разбор и решение практических заданий олимпиад прошлых лет;

самостоятельное решение задач, предлагаемых на сайтах с установленными автоматизированными системами тестирования программ в режиме реального времени [12. , 1. ];

выполнение домашних заданий олимпиад. Так, дистанционная олимпиада (ЦТИСО) проводится как заочная, т.е. учащиеся получают задание и выполняют его на занятиях в объединении и дома в течение примерно двух недель. Один из этапов областного командного турнира предполагает выполнение «домашнего задания». Участники команд

в течение полутора–двух месяцев, предшествующих дате проведения турнира, должны подготовить программу, играющую в заданную организаторами логическую игру.

Время, затрачиваемое на самостоятельную практическую работу, точно учесть очень трудно. В общей сложности оно составляет, приблизительно, около 30–40%% учебного времени (для учащихся второго года обучения) и более (для старших групп).

## 4. Способы контроля и оценки результатов

Способы контроля и оценки различны на разных этапах обучения.

### Первый этап

На первом этапе (см. Учебно-тематический план) школьники получают теоретические знания и практические навыки, которые потом используются как простейшие базовые знания и навыки в течение всего последующего процесса обучения. На этом этапе требуется постоянный контроль степени усвоения знаний, особенно в теоретических темах.

Фронтальный текущий контроль при изучении теоретических тем первого этапа обучения включает в себя устный опрос и письменные самостоятельные работы, обычно короткие по времени проведения (15–20 минут). По итогам таких опросов и самостоятельных работ отмечаются и озвучиваются правильные ответы. Неправильные ответы анализируются, разбираются, указываются ошибки, допущенные при ответе. Оценка как таковая не фиксируется, но правильный ответ может быть отмечен в некоторой сводной таблице, по которой в дальнейшем можно будет оценить работу ученика в течение всего учебного года.

Следствием текущего тематического контроля и оценки его результатов является корректировка учебного плана, например, повторение уже пройденных тем или отдельных вопросов, возможно, увеличение времени отведённого на изучение темы (в разумных пределах).

Тему завершает контрольная работа (итоговый контроль), которая проводится либо в виде письменной работы, либо с применением компьютера при помощи специального программного обеспечения.

Шкала оценок не используется, работа оценивается либо как «зачтённая», либо нет. При этом заранее озвучиваются критерий, позволяющий сдать контрольную работу — количество правильно решённых заданий (или процент от общего числа заданий в контрольной работе). Те ученики, которые не смогли сдать контрольную работу с первого раза, имеют возможность для её пересдачи.

Темы «Алгоритмы» и «Основы программирования на языке Паскаль» предполагают выполнение учащимися большого числа практических заданий — им необходимо выполнить порядка четырёх десятков блок-схем и программ на Паскале, соответственно.

При изучении этих тем, в основном, отслеживается своевременность и обязательность выполнения заданий, особенно домашних, соблюдение правил оформления работ. Как показывает практика, именно дисциплинированное отношение к выполнению практических заданий позволяет справиться с проблемами в освоении содержательной части данных тем, т.е. в вопросах, связанных с применением базовых структурных алгоритмических блоков, не смотря на их сложность. В сводной таблице отмечается каждое правильно выполненное задание (сначала блок-схема, затем программа). В отдельной таблице отмеча-

ется факт выполнения домашней работы, выполнение упражнений в классе (выполненное задание обозначается знаком «+», не выполненное знаком «-»).

Система контроля и оценки на первом этапе служит не только целям проверки усвоения знаний и выработки умений и навыков по конкретной теме. Она ставит более важную социальную задачу: развить у школьников умение проверять и контролировать себя, критически оценивать свою деятельность, находить ошибки и пути их устранения.

В конце первого года обучения подводятся итоги работы ученика в течение всего учебного года. По результатам работы за год каждому ученику рекомендуется либо продолжить обучение в группе второго года, либо, при недостаточно убедительных итогах, повторить курс первого года обучения.

Основными критериями успешности являются:

- выполнение практических заданий по теме «Алгоритмы» и «Основы программирования на языке Паскаль» в требуемом объёме (блок-схемы и программы);
- своевременное, регулярное и достаточно успешное выполнение домашних заданий и самостоятельных работ в классе в течение учебного года;
- успешное прохождение теоретических тем и успешная сдача контрольных работ по этим темам.

И ещё один важный критерий, который учитывается при переводе на второй год обучения и чаще всего являющийся определяющим — посещаемость.

### **Второй этап**

Большинство тем, рассматриваемых на втором этапе, содержат комплект практических заданий. Успешность изучения данных тем можно проконтролировать и оценить по объёму и срокам выполнения таких заданий.

Второй этап обучения характеризуется увеличением объема практической работы, т.е., непосредственно, программирования. И в этот период формируется почерк программиста, а, поэтому, необходимо сделать так, чтобы этот формирующийся почерк в дальнейшем способствовал более успешной, эффективной работе.

В связи с этим, при проверке практических заданий по любой теме курса, проверяется, кроме содержательной части, ещё и правильность оформления программ, чистота программного кода. Основными моментами, на которые обращается внимание, являются:

- правильное использование базовых структурных блоков;
- структурные отступы, горизонтальное и вертикальное форматирование в целом;
- правильное использование переменных, их типов, именованье переменных;
- аккуратное и правильное оформление заголовков, объявлений и комментариев;
- в более сложных программах — структурирование программы, грамотное разбиение на процедуры и функции, передача параметров и т.п.

Обычно, учащийся получает зачёт за выполненную и правильно работающую программу, только после устранения всех недочётов и замечаний по оформлению и по стилю программирования.

В этот же период появляется ещё один важный критерий успешности обучения — это результат участия в олимпиадах.

### **Третий этап**

Содержание третьего этапа, по большей части, далеко выходит за рамки школьного курса информатики. Некоторые вопросы, рассматриваемые на третьем этапе, изучаются на первых курсах профильных специальностей в вузах. Бессмысленно требовать от каждого обучающегося освоения всех этих вопросов в полном объёме. Оценка степени освоения программы становится более индивидуальной, чем на предыдущем этапе. Контролируется объём самостоятельно выполненных практических заданий, выявляются не только трудности в освоении текущих вопросов, но и пробелы в освоении изученных ранее базовых тем (а эти пробелы очень хорошо видны при решении сложных задач), прикладываются усилия к устранению этих пробелов.

Основным объективным критерием в этот период можно считать результаты участия обучающегося в олимпиадах и конкурсах по информатике и программированию. Стабильность результатов участия обучающегося в различных олимпиадах можно считать показателем достигнутого в процессе обучения уровня знаний и навыков.

Фактически, результаты третьего этапа — это итог работы обучающегося и педагога на протяжении всего курса обучения.

## 5. Учебно-тематический план

### 5.1. Предварительный этап

Включает в себя занятия, упражнения и тестовые задания по таким вопросам, как:

Проверка уровня логического мышления;

Проверка уровня базовых математических навыков, математического кругозора;

Проверка наличия задатков алгоритмического мышления;

Проверка базовых пользовательских навыков (работа с файлами, папками в операционной системе Windows, операции создания, копирования, перемещения, удаления и поиск папок и файлов), набор и редактирование простого текста (в русской и английской раскладке);

Проверка навыков в области телекоммуникационных технологий (работа с электронной почтой, поиск информации в сети Интернет с использованием поисковых систем).

(24 ч.)

### 5.2. Программа первого этапа обучения

#### 5.2.1. Системы счисления

В рамках данного модуля школьники должны изучить следующие темы:

Позиционные системы счисления. Основные определения. (3 ч.)

Перевод чисел из  $P$ -ой системы счисления в десятичную. Перевод чисел из десятичной системы счисления в  $P$ -ую. Перевод из двоичной системы в систему с основанием  $2^n$  (на примере восьмеричной и шестнадцатеричной). (6 ч.)

Арифметические операции в  $P$ -ых системах счисления (на примере двоичной системы счисления). (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

переводить числа из одной позиционной системы счисления в другую.

выполнять арифметические операции (сложение, вычитание, умножение, деление) в двоичной системе счисления.

**Методы аттестации по данному модулю:**

Контрольная работа. Задания на перевод чисел из одной системы в другую и выполнение арифметических операций в двоичной системе счисления.

### 5.2.2. Основы алгебры логики (Булева алгебра)

**В рамках данного модуля школьники должны изучить следующие темы:**

Алгебра логики: понятие высказывания, основные логические операции, их таблицы истинности, обозначения в языках программирования. (3 ч.)

Основные законы алгебры логики. Вычисление сложных логических выражений. Примеры сокращения сложных логических выражений при помощи основных теорем и тождеств алгебры логики. (9 ч.)

Примеры построений логических выражений. (6 ч.)

**По окончании обучения по данному модулю школьники должны:**

знать логические функции И, ИЛИ, НЕ, искл. ИЛИ, их таблицы истинности и обозначения.

уметь вычислять логические выражения, при необходимости уметь сокращать сложные логические выражения применяя тождества и теоремы булевой алгебры.

уметь построить логическое выражение по диаграмме Эйлера, по высказыванию на естественном языке.

**Методы аттестации по данному модулю:**

Контрольная работа (см. Приложение б).

### 5.2.3. Представление информации в компьютере (на примере основных встроенных типов языка Паскаль)

**В рамках данного модуля школьники должны изучить следующие темы:**

Представление целых чисел в компьютере. Двоичная запись. Бит, байт, машинное слово. (3 ч.)

Хранение отрицательных чисел. Дополнительный код. Выполнение арифметических операций сложения и вычитания в дополнительном коде. Арифметическое переполнение (6 ч.)

Битовые операции. (3 ч.)

Представление вещественных чисел. Представление десятичной дробной части в двоичной системе счисления (3 ч.)

Кодирование символов, ASCII-таблица, строки (тип string языка Паскаль). (3 ч.)

**По окончании изучения данного модуля школьники должны:**

Иметь начальное представление об основных типах данных языка Паскаль, таких как целочисленные типы (размер занимаемой памяти и диапазоны значений), вещественные числа (точность и диапазон вещественных типов), символы и строки.

**Методы аттестации по данному модулю:**

Итоговый опрос или контрольная работа.

#### 5.2.4. Алгоритмы. Начальный курс

**В рамках данного модуля школьники должны изучить следующие темы:**

Понятие алгоритма. Основные свойства алгоритма. Примеры словесного описания алгоритмов (3 ч.)

Начальные сведения о структурном подходе к составлению алгоритмов. Понятие структурного блока. Блок-схемы, основные обозначения. Последовательный алгоритм. (3 ч.)

Развилки: полная и усеченная. Выполнение упражнений по составлению алгоритмов с использованием развилки. (9 ч.)

Циклы с пред- и постусловием. Выполнение упражнений. (9 ч.)

Цикл с параметром. (3 ч.)

Выполнение упражнений по построению блок-схем алгоритмов. (9 ч.)

Вспомогательный алгоритм. (3 ч.)

**По окончании обучения по данному модулю школьники должны:**

Иметь начальные навыки составления простых алгоритмов с использованием базовых структурных блоков. Уметь построить блок-схему алгоритма с использованием одного-двух базовых блоков: последовательный алгоритм развилка в развилке, развилка в цикле и т.п.

Понимать, что такое переменная, оператор (операция) присваивания.

Выполнить необходимый объем практических упражнений на составление блок-схем (см. Приложение).

**Методы аттестации по данному модулю:**

Контрольные упражнения на составление блок-схем алгоритмов (см. Приложение 7).

#### 5.2.5. Основы программирования на языке Паскаль

**В рамках данного модуля школьники должны изучить темы, позволяющие перейти на практическое программирование на языке Паскаль:**

Операторы языка Паскаль. Консольный ввод-вывод. (6 ч.)

Структура программы. Среда программирования (краткий обзор, необходимый для начала работы). (3 ч.)

Типы данных. (3 ч.)

Выражения, операции, приоритет операций. (3 ч.)

Стандартные процедуры и функции для обработки данных простых типов. (3 ч.)

Стандартные процедуры и функции для работы со строками. (6 ч.)

Работа с файлами. Текстовый файл. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

работать со средой программирования: писать простые программы, запускать на выполнение, проводить отладку;

**Методы аттестации по данному модулю:**

Выполнение требуемого объёма практических упражнений (не менее 30–40 часов самостоятельных практических занятий со сдачей и защитой своих решений).

**5.2.6. Массивы**

**В рамках данного модуля школьники должны изучить следующие темы:**

Массивы. (3 ч.)

Линейный поиск в массиве. (3 ч.)

Нахождение минимума, максимума, суммы элементов массива. (3 ч.)

Сортировки: пузырьком, выбором, вставками. Сортировка подсчетом. (6 ч.)

Двумерные массивы. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

решать задачи с использованием массивов, линейного поиска, находить минимум, максимум, сумму элементов массива.

**Методы аттестации по данному модулю:**

Задачи и упражнения для самостоятельного решения (см. Приложение 8).

**5.2.7. Процедуры и функции**

**В рамках данного модуля школьники должны изучить следующие темы:**

Процедуры и функции. Синтаксис, примеры использования. (3 ч.)

Глобальные и локальные переменные. (3 ч.)

Передача параметров по значению и по ссылке. (3 ч.)

Рекурсия. Простые примеры использования рекурсии (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовать программу с использованием процедур и функций, в том числе рекурсивных.

**Методы аттестации по данному модулю:**

Задачи и упражнения для самостоятельного решения.

**5.2.8. Указатели, работа с файлами**

**Темы, указанные в данном разделе относятся к языку Паскаль. Но они могут изучаться как в конце первого этапа, так и в течение второго этапа, совместно с темами по изучению алгоритмов и структур данных:**

Указатели, работа с указателями, операция разыменования. Динамическое выделение памяти, динамические переменные. (6 ч.)

Типизированные и нетипизированные файлы. (6 ч.)

Множества. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовать программу с использованием динамических переменных и с динамическим выделением памяти;  
реализовывать программу, работающую с типизированным и нетипизированными файлами.

**Методы аттестации по данному модулю:**

Разработка программы копирования файлов, программы просмотра графических \*.bmp файлов.

## 5.3. Программа второго этапа обучения

### 5.3.1. Сложность алгоритмов

**В рамках данного модуля школьники должны изучить следующие темы:**

Понятие сложности алгоритма. Сложность по времени и по памяти. Асимптотические оценки. Смысл символа  $O()$ . (3 ч.)

Примеры оценок времени работы алгоритмов поиска, простых сортировок, рекурсивных алгоритмов. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

анализировать сложность алгоритма.

**Методы аттестации по данному модулю:**

Несколько вопросов в теоретической части итогового испытания.

### 5.3.2. Методы решения задач теоретического тура

В рамках данного модуля школьники изучают темы связанные с доказательством некоторых свойств алгоритма (конечность алгоритма, эффективность), доказательство правильности алгоритма на примере решения задач теоретических туров олимпиад по информатике.

Инвариант

Полуинвариант

Индукция

Доказательство от противного, чётность, делимость и остатки и др.[7. ]

**По окончании обучения по данному модулю школьники должны уметь:**

анализировать задачу и применять, стараясь чётко формулировать, один из изученных способов решения.

**Методы аттестации по данному модулю:**

Самостоятельное решение задач теоретического тура городской олимпиады по информатике прошлых лет[2. 4. ].

### 5.3.3. Перебор с возвратом

**В рамках данного модуля школьники должны изучить следующие темы:**

- Общая схема рекурсивного перебора. Перебор с возвратом (6 ч.)
- Перечисление комбинаторных объектов. Перестановки. Сочетания. (6 ч.)
- Оптимизация перебора. Метод ветвей и границ. (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать алгоритм рекурсивного перебора.

**Методы аттестации по данному модулю:**

Самостоятельное решение задач на перебор с возвратом.

### 5.3.4. Арифметические алгоритмы

**В рамках данного модуля школьники должны изучить следующие темы:**

- Алгоритм Евклида. (3 ч.)
- Алгоритм быстрого возведения в степень. (3 ч.)
- Проверка числа на простоту. Решето Эратосфена (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

- реализовать алгоритм Евклида, доказывать его корректность;
- реализовать алгоритм быстрого возведения в степень;
- реализовать алгоритмы проверки числа на простоту.

**Методы аттестации по данному модулю:**

Решение задача с использованием изученных алгоритмов.

### 5.3.5. Динамическое программирование

**В рамках данного модуля школьники должны изучить следующие темы:**

- Рекуррентные соотношения. (3 ч.)
- Треугольник Паскаля (3 ч.)
- Числа Фибоначчи (3 ч.)
- Динамическое программирование. Простой пример. (3 ч.)
- Поиск суммы. Разбиение подзадач. (3 ч.)
- Поиск максимума. Перекрытие подзадач. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

- анализировать рекуррентные соотношения;
- выводить простые рекуррентные соотношения;
- реализовать программу, вычисляющую значение по рекуррентному соотношению с использованием динамического программирования.

**Методы аттестации по данному модулю:**

Задача в практической части итогового испытания.

### 5.3.6. Линейные динамические структуры данных

**В рамках данного модуля школьники должны изучить следующие темы:**

- Указатели, динамически распределяемая память. Линейные динамические структуры данных. (3 ч.)
- Связанные списки. Линейные и кольцевые списки. Односвязные и двусвязные списки. (3 ч.)
- Очереди. Реализации на базе массива и списка. (3 ч.)
- Стеки. Реализации на базе массива и списка. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать линейные динамические структуры данных и использовать их при решении задач.

**Методы аттестации по данному модулю:**

Задача в практической части итогового испытания.

## 5.4. Программа третьего этапа обучения

### 5.4.1. Деревья

**В рамках данного модуля школьники должны изучить следующие темы:**

- Деревья. Основные понятия. Корень. Листья. Высота дерева. Примеры. Реализация деревьев. (3 ч.)
- Обходы дерева. (3 ч.)
- Динамическое программирование на дереве. (3 ч.)
- Бинарные деревья. (3 ч.)
- Деревья поиска (без алгоритмов балансировки). (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать деревья и использовать их при решении задач.

**Методы аттестации по данному модулю:**

Задача в практической части итогового испытания..

### 5.4.2. Графы. Базовые алгоритмы

**В рамках данного модуля школьники должны изучить следующие темы:**

- Понятие графа. Основная терминология. Ориентированные и неориентированные графы. (3 ч.)
- Кратчайшие пути. Волновой алгоритм и поиск в ширину. (3 ч.)
- Поиск в глубину. Приложения. (6 ч.)
- Игры на графах. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать базовые алгоритмы на графах.

**Методы аттестации по данному модулю:**

Задача в практической части итогового испытания.

**5.4.3. Сортировки****В рамках данного модуля школьники должны изучить следующие темы:**

Сортировки. Трудоемкость, устойчивость. (3 ч.)

Простые алгоритмы сортировок: пузырьком, выбором, вставками (повторение). Анализ сложности данных алгоритмов. (3 ч.)

Слияние отсортированных массивов. Сортировка слиянием. (3 ч.)

Быстрая сортировка. Поиск К-ой порядковой статистики. (3 ч.)

Двоичный поиск в отсортированном массиве. (3 ч.)

Сортировка последовательностей, хранящихся в файлах, с оценкой числа сравнений и присваиваний. (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать алгоритм сортировки слиянием;

реализовывать алгоритм быстрой сортировки;

реализовывать алгоритм поиска К-ой порядковой статистики;

реализовывать алгоритм двоичного поиска в отсортированном массиве.

**Методы аттестации по данному модулю:**

Задача в практической части итогового испытания.

**5.4.4. Поиск стратегии в играх с полной информацией****В рамках данного модуля школьники должны изучить следующие темы:**

Игры с полной информацией. Предопределенность исхода игры, выигрышные и проигрышные позиции. Примеры простейших детерминированных игр (3 ч.)

Дерево игры, перебор дерева игры на примере игры «Крестики-нолики». (6 ч.)

Игра Ним, поиск стратегии (3 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

уметь определять победителя в простейших играх, находить выигрышную стратегию.

уметь реализовать перебор дерева игры

**Методы аттестации по данному модулю:**

Решение задач на поиск стратегии в детерминированных играх.

Подготовка программы-игрока для участия в турнире программ областного турнира по программированию.

## 5.5. Модули по выбору (примеры)

### 5.5.1. Работа в консоли

**В рамках данного модуля школьники должны изучить следующие темы:**

Консоль. Основные команды консоли (в зависимости от изучаемой ОС). (3 ч.)

Запуск программ из консоли. Передача параметров. (3 ч.)

Файлы сценариев (bat, cmd или sh, в зависимости от изучаемой ОС). (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

совершать простые действия из консоли;

писать и запускать простые файлы сценариев.

### 5.5.2. Графы. Усложненный курс

**В рамках данного модуля школьники должны изучить следующие темы:**

Взвешенные графы. Кратчайшие пути. Алгоритмы Дейкстры, Форда-Беллмана, Флойда. (9 ч.)

Остовные деревья. Алгоритмы Краскала и Прима. (6 ч.)

Поиск мостов, точек сочленения, максимального паросочетания. Поиск максимального потока в сети. (12 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

реализовывать стандартные алгоритмы на графах.

### 5.5.3. Вычислительная геометрия

**В рамках данного модуля школьники должны изучить следующие темы:**

Координаты и векторы на плоскости. (3 ч.)

Способы описания линий на плоскости. (3 ч.)

Взаимное расположение точек и фигур. (9 ч.)

Многоугольники. Проверка выпуклости. Проверка принадлежности точки. Вычисление площади. (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

строить программы, решающие задачи вычислительной геометрии.

### 5.5.4. Алгоритмы и структуры данных. Усложненный курс

**В рамках данного модуля школьники должны изучить следующие темы:**

Куча. Сортировка кучей. (3 ч.)

Деревья поиска. (6 ч.)

Алгоритмы на строках. (6 ч.)

Хеш-таблицы. (6 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

использовать сложные алгоритмы и структуры данных.

### **5.5.5. Разработка программного проекта**

**В рамках данного модуля школьники должны изучить следующие темы:**

Построение простых пользовательских интерфейсов. (3 ч.)

Построение небольшой программы. Например:

- Игра «Жизнь»
- Игра «Крестики-нолики»
- Игра «Пятнашки»
- (12 ч.)

**По окончании обучения по данному модулю школьники должны уметь:**

строить небольшие программы (игры, тесты, редакторы);

строить простые пользовательские интерфейсы.

**Методы аттестации по данному модулю:**

Реализация программы и документации к ней с последующим анализом преподавателем

## **5.6. Язык программирования С**

**Комментарий к разделу**

Данная тема предназначена для учеников прошедших базовый курс по программированию на языке Паскаль. Переход к программированию на языке С возможен не ранее конца второго — начала третьего года обучения при наличии достаточного практического опыта в создании программ на Паскале.

Предполагается, что основы языка С (основной синтаксис) можно изложить за 6–8 занятий-лекций, и затем перейти к практическим занятиям (решению задач), в ходе которых более подробно на конкретных примерах рассматриваются некоторые тонкости и особенности в использовании языка С, стандартные функции библиотеки времени исполнения и другие вопросы.

### **5.6.1. Введение в синтаксис языка С**

Структура программы.

Объявления переменных и простые типы данных.

Основные операции.

Синтаксис операторов, основных алгоритмических структур.

Массивы.

Стандартные функции ввода-вывода.

### 5.6.2. Указатели, функции, передача параметров

Работа с указателями и динамическое выделение памяти.

Функции, передача параметров в функцию. Прототипы функций, заголовочные файлы.

Строки символов, стандартные функции для обработки строк, функции для работы с блоками памяти. Обзор математической библиотеки. Служебные функции.

Таблица приоритетов операций. Некоторые особые для языка С операции и операторы: тернарный, составные операторы присваивания, переключатель (switch).

Структурный тип, объединение, перечисление.

Инициализация переменных структурных типов. Приведение типов.

Область видимости имен переменных, функций.

Получение исполняемого кода. Препроцессинг, компиляция, сборка программы. Многофайловый проект.

## 5.7. Язык программирования С++. Основы объектно-ориентированного программирования (ООП)

### Комментарий к разделу

Данная тема предназначена для учеников прошедших базовый курс по программированию на языке Паскаль и С. Переход к программированию на языке С++ возможен не ранее конца третьего — начала четвёртого года обучения при наличии достаточного практического опыта в создании программ на языке С. Изучение вопросов указанных в первом разделе позволяют программировать на языке С++ в стиле языка С, т.е. используя структурный подход к программированию. Далее, в последующих разделах, основы языка рассматриваются параллельно с базовыми понятиями ООП, такими как инкапсуляция, наследование, полиморфизм и др. Последний раздел содержит обзор и некоторые основные понятия стандартной библиотеки шаблонов — мощной библиотеки позволяющей эффективно использовать многие стандартные структуры данных и алгоритмы для решения различных задач.

### 5.7.1. Основные сведения по языку С++

Ядро языка С++ — это расширенное ядро языка С. Расширение ядра: объявление переменных в любом месте блока, передача параметров функции по ссылке, параметры по умолчанию. Перегрузка функций. Новые заголовочные файлы. Пространства имён. Новая библиотека ввода-вывода.

Новые операторы для работы с динамическими переменными (new, delete).

Обработка исключений. Защищённые блоки.

Шаблоны функций.

(9 ч.)

### 5.7.2. Классы

Инкапсуляция — сокрытие данных и объединение в одной структуре данных и методов (функций) обработки этих данных.

Структура класса, общедоступный и закрытый разделы, члены-данные и функции-члены. Объявление и определение функций-членов класса.

Конструкторы и деструкторы. Конструктор по умолчанию, инициализирующий и копирующий конструкторы.

Дружественные функции.

Перегрузка операций, перегрузка операций ввода-вывода.

Статические члены класса. Инициализация статических членов-данных. Обращение к статическим членам.

Практический пример: разработка класса Fraction, реализующего работу с обыкновенными дробями.

Шаблоны классов.

Практический пример: разработка класса-шаблона Stack<>.

Применение ключевого слова const. Константные величины. Указатели на постоянные данные и константные указатели. Константные методы класса.

(15 ч.)

### 5.7.3. Наследование

Понятие наследования. Способы наследования. Защищённые члены класса.

Множественное наследование. Виртуальное наследование.

Порядок вызова конструкторов и деструкторов в базовом и наследуемом классах.

Инициализация базового класса.

Виртуальные функции, чисто виртуальные функции. Абстрактные классы.

Обращение к членам класса через указатель на базовый класс. Динамический полиморфизм.

Операторы приведения типа.

(6 ч.)

### 5.7.4. Стандартная библиотека шаблонов STL

Обзор библиотеки. Основная терминология: класс-контейнер, коллекция, итератор, алгоритм, функтор, адаптер и т.п.

Контейнерные адаптеры — стек, очередь и очередь с приоритетами.

Базовые последовательные контейнеры: вектор, дек, список. Основной интерфейс.

Общая часть и различия, обусловленные внутренней структурой контейнеров.

Базовые ассоциативные контейнеры: множества, мультимножества, отображения и мультиотображения. Основной интерфейс. Общая часть для всех контейнеров и отличия ассоциативных контейнеров, обусловленные внутренней структурой. Отображение как ассоциативный массив.

Объекты-функции, применение их в качестве критерия сортировки во множествах, приоритетных очередях. Унарные и бинарные предикаты.

Итераторы. Категории итераторов: произвольного доступа, двунаправленные, прямой итератор, итераторы ввода, вывода. Итераторные адаптеры.

Алгоритмы. Обзор. Немодифицирующие, модифицирующие алгоритмы. Алгоритмы удаления, перестановки, алгоритмы упорядоченных интервалов, вычислительные. Общие принципы применения алгоритмов. Примеры использования алгоритмов.

Строковый класс.

(24 ч.)

## 5.8. Язык программирования Python

### Комментарий к разделу

Python — язык высокого уровня, с достаточно простым и интуитивно понятным синтаксисом. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения. Python обладает большой стандартной библиотекой модулей. Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач.

К изучению языка Python учащиеся объединения переходят после изучения основ программирования на языке Pascal и, возможно, C/C++, то есть не ранее второго–третьего года обучения.

### 5.8.1. Введение

Знакомство с простейшей средой программирования.

Написание простейших программ с использованием простого ввода-вывода.

Оператор присваивания, типы данных, числовой, строковый типы, динамическая типизация. Простые развилки, циклы while и for.

### 5.8.2. Структура программы и основные инструкции

Структура программы. Условный оператор. Простые встроенные функции. Знакомство с циклами while и for. True и False, break и continue.

Строки, методы строк.

Отладчик.

### 5.8.3. Коллекции

Строки. Индексация, срезы.

Списки, основные методы списков.

Кортежи. Преобразование коллекций

Множества

Методы Split и join. Списочные выражения

Методы списков и строк подробно.

Вложенные списки  
Знакомство со словарями  
(48 ч.)

## 5.9. Основы HTML-вёрстки и основы программирования на языке JavaScript

### Комментарий к разделу

Данная тема является факультативной и предназначена для учеников, желающих заниматься программированием в каникулы, в основном, в июне. Рассматриваются принципы html-вёрстки потоком, применение таблиц стилей, начальные сведения по внедрению в html-документы сценариев на языке JavaScript. Перечислены теоретические темы, каждая сопровождается разбором практических примеров и выполнением самостоятельных заданий.

Необходимое время для рассмотрения теоретических вопросов — порядка 30 часов. Плюс дополнительная самостоятельная практическая работа с html и JavaScript.

### 5.9.1. Основы блочной вёрстки HTML документа

Простой html-файл, понятие элемента и тега, атрибуты и их значения, примеры.

Структура html-файла, декларация типа документа, заголовок документа, тело документа.

Блочные и строчные элементы. Границы элементов, поля и отступы. Размеры элементов. Принципы размещения блочных и строчных элементов на странице по умолчанию. Понятие о стилевых свойствах элементов. Примеры применения таблиц стилей. Подключение css-файла с описанием стилей к html-файлу.

Свойства шрифта и текста. Способы обозначения цвета, единицы длины. Свойства, задающие фон элементов.

Теги и свойства стилей для создания и форматирования таблиц и списков.

Псевдоклассы и псевдоэлементы, тень для текста и блока, фон с градиентом, Вставка графических элементов.

Свойства управляющие отображением элементов на странице. Плавающие элементы, позиционирование. Свойства float, position, display и visibility.

Создание форм и добавление элементов управления на страницу.

Инструменты разработчика html. Текстовые редакторы, браузеры и встроенные средства просмотра и отладки, справочники по тегам и свойствам стилей, онлайн-редакторы для проверки и отладки небольших фрагментов кода, валидаторы кода, программы для подбора цветовой гаммы, генераторы «рыбного» текста и др.

Разработка макета страницы с фиксированной шириной.

### **5.9.2. Основы языка JavaScript и динамического HTML**

Общие сведения о JavaScript и динамическом html. Понятие о программировании на стороне сервера и на стороне клиента. Подключение скриптов на JavaScript к html-файлу. Понятие события и обработчика события.

Основы синтаксиса JavaScript. Основные синтаксические структуры, объявления переменных. Создание простой функции, передача параметров.

Динамическое изменение свойств стиля элементов на странице при помощи JavaScript. Обработка событий элементов форм, создание простой интерактивной html-страницы.

## **6. Материально-техническое, информационное и учебно-методическое обеспечение программы**

### **Технические средства обучения**

Рабочее место учителя — персональный компьютер (с установленным ПО), стол, стул;  
Стол мобильный, для проектора;  
Мультимедиа проектор;  
Экран;  
Принтер лазерный, ч/б;  
Источник бесперебойного питания;  
Рабочее место ученика (10–11 шт.) — персональный компьютер (с установленным ПО), стол компьютерный, стул, отвечающие санитарным нормам;  
Комплект сетевого оборудования;  
Комплект оборудования для подключения к сети Интернет;  
Копировальный аппарат;  
Мобильное устройство для хранения информации (флэш-память);

### **Программное обеспечение**

ОС и графический оконный интерфейс пользователя;  
Калькулятор;  
Клавиатурный тренажер;  
Кодирование и обработка звука;  
Воспроизведение видеозаписей;  
Воспроизведение аудиозаписей;  
Файловый менеджер для работы с файлами;  
Информация о системе;  
Запись на CD/DVD;  
Управление задачами, контроль загрузки памяти и процессора;  
Текстовый процессор;  
Словарь иностранных языков;  
Сканирование;  
Табличный редактор;  
Мощный редактор точечной графики;  
Редактор векторной графики;  
Программа подготовки презентаций;  
Редактор 3D графики;  
Среда разработки Pascal;  
Визуальная среда разработки C/C++;  
Браузер интернета;  
Электронная почта;  
Обзор сети;

Шифрование закрытым и открытым ключом;  
Антивирусный сканер;  
Прокси-сервер с фильтрацией контента;  
Оболочки для архивирования и защиты файлов паролем

### **Информационно-методические средства**

Справочная и учебная литература, а так же доступ к источникам в Интернет:  
по языку Паскаль;  
по языкам C/C++;  
по языку Python;  
по алгоритмам, методам решения олимпиадных задач;  
по HTML, CSS, JavaScript;  
по операционным системам, командным оболочкам, системным утилитам и т.п;  
по различным темам, связанным с компьютерами и информатикой: по криптографии, работе с графикой, искусственному интеллекту и т.п.  
Сборники упражнений, задач по теоретическим темам и по программированию.  
Системы тестирования программ-решений в режиме реального времени.

## 7. Список информационных источников

### 7.1. Нормативно-правовые документы

1. Федеральный закон «Об образовании в Российской Федерации» № 273-ФЗ от 29.12.12 года. – [Электронный ресурс]. – Режим доступа: [base.garant.ru/70291362/](http://base.garant.ru/70291362/) (информационно-правовой портал «Гарант»).
2. Приказ Министерства образования и науки Российской Федерации от 09 ноября 2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» [Электронный ресурс]. - Режим доступа: <https://www.garant.ru/products/ipo/prime/doc/72016730/> (информационно-правовой портал «Гарант»).
3. Концепция развития дополнительного образования детей, утв. распоряжением Правительства РФ от 4.09.2014 года № 1726-р. [Электронный ресурс] — Режим доступа: <http://минобрнауки.рф/документы/ajax/4429> (официальный сайт Министерства образования и науки РФ).
4. СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей", утв. Главным государственным санитарным врачом РФ от 04.07.2014 N 41. – [Электронный ресурс]. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_168723/](https://www.consultant.ru/document/cons_doc_LAW_168723/) (официальный сайт справочной правовой системы «КонсультантПлюс»).
5. Государственная программа РФ «Развитие образования на 2013-2020 годы, утвержденной постановлением Правительства РФ № 295 от 15.04.2014 г. – [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/70643472/#friends> (информационно-правовой портал «Гарант»).
6. Стратегия инновационного развития Российской Федерации на период до 2020 года, утвержденным распоряжением Правительства РФ № 2227-р от 08.12.2011 года. – [Электронный ресурс]. – Режим доступа: <https://www.garant.ru/products/ipo/prime/doc/70006124/> (информационно-правовой портал «Гарант»).
7. Федеральная целевая программа развития образования на 2016-2020 годы, утвержденной Постановлением Правительства РФ № 497 от 23.05.2015 года. – [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/71044750/> (информационно-правовой портал «Гарант»).

### 7.2. Теоретические основы информатики и начальный курс программирования на языке Паскаль

1. Андреева, Е.В. Математические основы информатики. Элективный курс: Учебное пособие [Текст] / Е. В. Андреева, Л. Л. Босова, И. И. Фалина. – М.: БИНОМ. Лаборатория знаний, 2005 – 328 с.: ил.

2. Информатика. Методическая газета для учителей информатики [Текст]. – Издательский дом «Первое сентября».
3. Мануйлов, В.Г. Разработка программного обеспечения на Паскале. [Текст] / В.Г. Мануйлов. – М.: Издательство ПРИОР, 1998.
4. Марченко, А.И. Программирование в среде Turbo Pascal 7.0 [Текст] / Марченко А.И., Марченко Л.А.; под ред. В.П. Тарасенко. – М.: Бином Универсал, К.: Юниор, 1997. – 496 с.

### **7.3. Подготовка к олимпиадам по информатике**

1. Архив задач с проверяющей системой на Timus Online Judge [Электронный ресурс]. – Режим доступа: <http://acm.timus.ru/>.
2. Бадин, Н.М. Ярославские олимпиады по информатике [Текст] / Н.М. Бадин, С.Г. Волченков, Н.Л. Дашниц, П.А. Корнилов. – Ярославль: Изд-во ЯрГУ. – 1995.
3. Брудно, А.Л. Олимпиады по программированию для школьников [Текст] /А.Л. Брудно, Л.И. Каплан. – М.: Наука, 1985. – 95 с.
4. Волченков, С.Г. Ярославские олимпиады по информатике. Сборник задач с решениями [Текст] / С. Г. Волченков, П. А. Корнилов, Ю. А. Белов, и др. – М.: БИНОМ. Лаборатория знаний, 2010. – 405 с. : ил.
5. Дистанционная подготовка по информатике [Электронный ресурс]. – Режим доступа: <http://informatics.mcsme.ru/>.
6. Задачи, загадки, логические игры, ребусы, математика «Игры разума» [Электронный ресурс]. – Режим доступа: <http://www.braingames.ru/>.
7. Канель-Белов, А. Я. Как решают нестандартные задачи [Текст] / А.Я. Канель-Белов, А.К. Ковальджи; под ред. В. О. Бугаенко. – М.: МЦНМО, 2008. – 96 с.
8. Кирюхин, В.М. Методика решения задач по информатике. Международные олимпиады [Текст] / Кирюхин В. М., С. М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2007. – 600 с.: ил.
9. Окулов, С.М. Программирование в алгоритмах [Текст] / С. М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2002. – 341 с.: ил.
10. Олимпиады по информатике [Электронный ресурс]. – Режим доступа: <http://neerc.ifmo.ru/school>.
11. Олимпиады по программированию [Электронный ресурс]. – Режим доступа: <http://olympiads.ru/>.
12. Школа программиста [Электронный ресурс]. – Режим доступа: <http://acmp.ru/>.

### **7.4. Языки программирования С и С++**

1. Джосьютис, Николай. С++ Стандартная библиотека. Для профессионалов [Текст] / Н. Джосьютис. – СПб: Питер, 2004. – 730 с.: ил.

2. Справочники по языкам C и C++ [Электронный ресурс]. – Режим доступа: <http://ru.cppreference.com/>.
1. Шилдт, Герберт. Полный справочник по C, 4-е издание.: Пер. с англ. [Текст] / Г. Шилдт. – М.: Издательский дом «Вильямс», 2005. – 704 с.: ил.
2. Шилдт, Герберт. Полный справочник по C++, 4-е издание.: Пер. с англ. [Текст] / Г. Шилдт. – М.: Издательский дом «Вильямс», 2003. – 800 с.: ил.
3. Шилдт, Герберт. Самоучитель C++: Пер. с англ. — 3-е изд. [Текст] / Г. Шилдт. – СПб.: БХВ-Петербург, 2003. – 688 с.

## 7.5. HTML, CSS, JavaScript

1. Дуванов, А.А. CSS-вёрстка для продолжающих [Текст] / А.А. Дуванов. – Информатика. – 2009. – №11–12.
2. Фленаган, Дэвид JavaScript. Подробное руководство. – пер. с англ. [Текст] / Д. Фленаган. – СПб: Символ-Плюс, 2008. – 992 с.,ил.
3. Освоить вёрстку за 9 недель. Сайт со справочной информацией, самоучителями и пр. по языку html, таблицам стилей, вопросам, связанным с вёрсткой html-страниц [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/>.
4. Сайт для загрузки свободно-распространяемого многофункционального текстового редактора Notepad++ [Электронный ресурс]. – Режим доступа: <http://notepad-plus-plus.org/download/>.
5. Сайт разработчиков многофункционального текстового редактора Sublime Text 3 (условно-бесплатная программа) [Электронный ресурс]. – Режим доступа: <http://www.sublimetext.com/3>.
6. Сервис проверки HTML от W3C [Электронный ресурс]. – Режим доступа: <http://validator.w3.org/>.
7. Сервис проверки CSS от W3C [Электронный ресурс]. – Режим доступа: <http://jigsaw.w3.org/css-validator/>.
8. Инструмент для отладки и просмотра небольших фрагментов кода (HTML+CSS+JavaScript) [Электронный ресурс]. – Режим доступа: <http://jsfiddle.net/>.
9. Инструмент для отладки и просмотра небольших фрагментов кода (HTML+CSS+JavaScript) [Электронный ресурс]. – Режим доступа: <http://cssdeck.com/>.
10. Подбор цветовой гаммы для сайта [Электронный ресурс]. – Режим доступа: <http://colorscheme.ru/>.
11. Генератор «рыбного» текста [Электронный ресурс]. – Режим доступа: <http://www.blindtextgenerator.com/ru>.

12. Справочник по языку, самоучители, справочные и учебные статьи и другие материалы по языку [Электронный ресурс]. – Режим доступа: JavaScript.<http://javascript.ru/>.

## 8. Список информационных источников

Приложение 1. Порядок проведения вступительных испытаний.

Приложение 2. Вступительные испытания. Задачи для будущих программистов. Задачи для домашнего задания.

Приложение 3. Вступительные испытания. Тесты на проверку уровня логического мышления. 6 вариантов. Правильные ответы для проверки. Бланки для ответов.

Приложение 4. Вступительные испытания. Примеры задач на логику с ответами.

Приложение 5. Итоговое задание вступительного испытания. Бланк для заполнения ответов. Правильные ответы.

Приложение 6. Задания контрольной работы по Булевой алгебре. 6 вариантов.

Приложение 7. Задачи для учащихся первого года обучения по теме «Алгоритмы».

Приложение 8. Пример практических заданий для самостоятельного решения. Упражнения по начальным темам языка Паскаль.

Приложение 9. Тематический план для экспериментальной группы семиклассников.