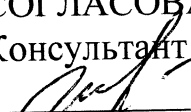
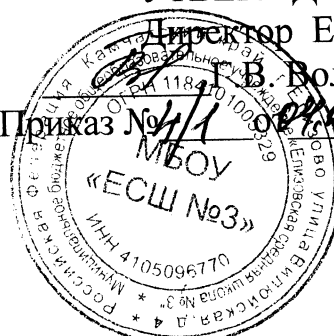


Управление образования администрации ЕМР  
Муниципальное бюджетное общеобразовательное учреждение  
«Елизовская средняя школа №3»

СОГЛАСОВАНО:  
Консультант ОВР и ДО АЕМР  
 Я.С. Иванова  
от « 07 » сентября 2020г.

УТВЕРЖДАЮ:  
Директор ЕСШ №3  
Т.В. Волошина  
Приказ № 4/1  
МБОУ  
«ЕСШ №3»  
ИНН 4105096770  
1003/20  
2020г.



Дополнительная общеразвивающая программа  
естественнонаучной направленности


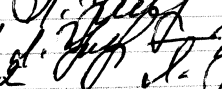
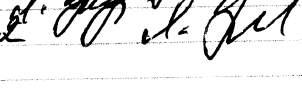
**«Спортивное программирование».**

Возраст обучающихся: 15-16 лет.

Срок реализации: 1 год

Автор-составитель:  
Степанова Татьяна Валерьевна  
Учитель информатики

информация о дополнении и допуске Программы к реализации:

Учебный год	№ и дата приказа по МБОУ «ЕСШ№3»	Подпись замдиректора по ВР
2020-2021	Приказ № 4/1 от 04.09.20	
2021-2022	Приказ № 124/6 от 21.08.21	
2022-2023	Приказ № 2/3 от 02.09.22	

г. Елизово  
2020г.

### Содержание:

I.	Пояснительная записка	3-11
II.	Учебно-тематический план	12
III.	Содержание программы	13-17
IV	Календарно-тематическое планирование	18-19
V.	Методическое обеспечение	20
VI.	Список литературы	21

## 1. Пояснительная записка

**Направленность программы.** Дополнительная общеобразовательная программа «Спортивное программирование.» (далее Программа) по содержанию является *естественнонаучной*; по функциональному назначению - *учебно-познавательной, прикладной*, по форме организации - *индивидуально ориентированной*; по времени реализации - *годовой*.

Нормативно-правовая база Программы:

- Конституция Российской Федерации;
- "Конвенция о правах ребенка" (одобрена Генеральной Ассамблеей ООН 20.11.1989) (вступила в силу для СССР 15.09.1990);
- Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ (действующая редакция, 2016);
- Федеральный закон от 24 июля 1998 г. N 124-ФЗ "Об основных гарантиях прав ребенка в Российской Федерации" (с изменениями и дополнениями);
- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
- СанПин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41)
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства образования и науки РФ от 29.08.2013 № 1008).
- Методические рекомендации по проектированию дополнительных развивающих программ (включая разноуровневые Программы )

программирование» согласно Федеральному государственному образовательному стандарту общего образования второго поколения (далее ФГОС) включают на базовом и профильном уровнях:

- владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- владение умением понимать Программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкции программирования; умением анализировать алгоритмы с использованием таблиц;
- владение стандартными приемами написания на алгоритмическом языке Программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- использование готовых прикладных компьютерных программ по выбранной специализации;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- владение элементарными навыками формализации прикладной задачи и документирования программ.

Однако учебный план общеобразовательного учреждения не содержит необходимого количества часов для изучения школьниками предмета ПРОГРАММИРОВАНИЯ на уровне, который позволил бы будущему выпускнику быть конкурентоспособным при поступлении на факультет информационных технологий в ВУЗы России.

В связи с этим особую актуальность приобретают программы дополнительного образования, предлагающие обучающимся дополнительные образовательные Программы, отличные от школьных программ, способные углубить их знания в теоретической информатике, сформировать практические навыки в области информационных технологий и практического программирования.

Данная Программа призвана решить эту проблему и предназначена для обучения девятиклассников основам программирования.

Содержание обучения, представленное в Программе, позволяет обучающимся изучить основы процедурного программирования, с целью выбора дальнейшей индивидуальной образовательной траектории:

- изучение языка Python в визуальной объектно-ориентированной среде Python
- выбрать для изучения другой язык программирования (C, C#, C++);

- выбрать другой язык программирования;
- объединить свои знания нескольких профилей программирования для проектной деятельности, например, «Программирование» + «Web-программирование»;

Это даст возможность обучающимся более полно выразить свой интеллектуальный и творческий потенциал, реализовать собственные идеи, применить освоенные навыки программирования в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий в области информационных технологий

**Цель Программы** – создать условия для удовлетворения индивидуальных потребностей обучающихся в интеллектуальном развитии, посредством обеспечения прочного и осознанного овладения обучающимися основами знаний процедурного программирования на языке Python и СИ.

создание условий для, выявление и поддержка одаренных учащихся

**Основные задачи:**

- Знакомство с принципами структурирования, формализации информации и выработка умений строить и читать блок-схемы решения задач.
- Развитие алгоритмического и логического стилей мышления.
- Формирование навыков обработки информации различного типа, используя средства языка программирования Python, СИ.

**Новизна программы.** Методологической основой Программы является компетентностно-деятельностный подход, который обеспечивает:

- формирование готовности обучающихся к саморазвитию и непрерывному образованию;
- активную учебно-познавательную деятельность обучающихся;
- построение образовательного процесса с учётом индивидуальных возрастных, психологических и физиологических особенностей обучающихся.

Компетентностно-деятельностный подход в обучении программированию позволит обучающемуся научиться создавать Программы для решения не только стандартных задач, но и задач повышенного уровня сложности, участие в турнирах по программированию.

Компетентностно-ориентированный подход в обучении программированию обучающегося реализуется в содержании Программы через формирование следующих ключевых образовательных компетенций:

**Учебно-познавательные компетенции**

- понимание стремительности развития компьютерной техники и в связи с этим желание следить за прогрессом в области информационных технологий;
- понимание важности изучения базовых, трудных, непопулярных тем информатики;

- формирование стремления обучающихся к самостоятельной познавательной деятельности в области информационных технологий;
- воспитание критической самооценки самостоятельной деятельности;
- формирование любви к программированию, как самой интересной и важной части предмета информатики;
- выработка потребности во внешней экспертизе своей работы.

### ***Информационные компетенции***

- умение различить вид и тип информации и применить для ее обработки наиболее приемлемый алгоритм;
- умение работать с электронными учебниками, информационными сайтами с целью самостоятельного приобретения новых знаний;
- умение сохранять необходимую информацию, архивировать и переносить на другой носитель.

### ***Коммуникативные компетенции***

- владение различными социальными ролями в коллективе;
- умение пользоваться чужой информацией, не нарушая авторских прав; умение слушать, рассуждать, спорить, доказывать свою точку зрения; умение представить результаты своей работы в виде учебного проекта.

При составлении Программы тщательно отбиралось содержание, средства и методы преподавания, руководствуясь следующими принципами:

- научность, в содержании Программы находят отражение новейшие достижения в этой области с адаптацией на познавательные возможности обучающихся;
- доступность, этот принцип реализуется через выделение уровней обучения;
- системность, принцип основан на поиске и построении внутри- и межпредметных связей и ассоциаций.

**Содержание Программы** ориентировано на формирование таких личностных качеств выпускника Центра как креативно и критически мыслящего, осознающего ценность научно-технического образования для себя и общества и подготовленного к осознанному выбору профессии, мотивированного на творчество и инновационную деятельность, на образование и самообразование в течение всей своей жизни.

Содержание Программы «Программирование в среде Python» не предполагает, что обучающиеся ранее были знакомы с базовыми понятиями алгоритмики. Однако, если некоторые из обучающихся имеют опыт

программирования в других средах (Scratch, КуМир, ЛогоМиры и др.), процесс изучения языка Python для них будет проходить более комфортно.

Обучение программированию предполагает дифференцированный подход. В частности, для одаренных обучающихся могут быть предложены задачи олимпиадного уровня, а также свой творческий потенциал отдельные обучающиеся могут реализовать через проектную деятельность в объектно-ориентированном программировании, основой которых является язык программирования Python. Освоение этих сред предполагает самостоятельную работу обучающихся. Для одаренных обучающихся могут быть предложены задачи олимпиадного уровня, а также свой творческий потенциал отдельные обучающиеся могут реализовать через проектную деятельность (работа с графикой и анимацией).

При освоении Программы у обучающихся формируются два взаимодополняющих стиля мышления: логико-алгоритмическое и системно-комбинаторное.

Логико-алгоритмическое мышление проявляется в умении: строить логические утверждения о свойствах данных, мыслить индуктивно и дедуктивно. Признаки системно-комбинаторного мышления следующие: видение предметов и явлений в целостности, взаимосвязях; умение строить несколько точек зрения на один и тот же объект; умение комбинировать понятийные и орудийные средства из различных тем при построении алгоритмов.

Содержание обучения, представленное в Программе позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, исполняемых файлов — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики.

Активизация познавательного процесса позволяет обучающимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Исключительно велика роль программирования для формирования мышления обучающихся, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач.

Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений обучающихся активизирует процесс индивидуально-личностного становления учащихся.

Программа педагогически целесообразна так как формирует различные приемы умственной деятельности: анализа, синтеза, сравнения, классификации, аналогии и обобщения, способствует воспитанию интереса к информатике и интеллектуальному развитию, а так же расширяет знания школьного курса математики, информатики, физики, литературы, рисования и т.д.

Программа ориентирована на большой объем практических работ с использованием компьютера (до 60% учебного времени) по всем изучаемым темам программирования при условии соблюдения требований СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей" от 4 июля 2014 г. № 41.

**Адресат программы:** Программа рассчитана на обучение и развитие детей 15-18 лет.

Программа может применяться при работе с детьми с особыми возможностями здоровья (за исключением детей с нарушением интеллектуального развития).

**Объем и срок освоения программы. Режим занятий:**

Количество занятий	Общее количество учебных часов в год
1гр. 1 раз в неделю по 1 часу	36
2гр. 1 раз в неделю по 1 часу	36

**Форма обучения** – очная.

**Количество детей в группах**

15 человек.

Состав групп постоянный, набор детей свободный, группы разновозрастные.

**Планируемые результаты освоения обучающимися программы.**

**1 год обучения**

***Изучение предметной области должно обеспечить:***

1. сформированность начальных основ логического, алгоритмического мышления;
2. сформированность начальных умений применять полученные знания при решении различных задач;
3. сформированность начальных представлений о влиянии информационных технологий на жизнь человека в обществе; понимание социального, экономического, политического,



культурного, юридического, природного, эргономического, медицинского и физиологического контекстов информационных технологий;

4. принятие этических аспектов информационных технологий; осознание ответственности людей, вовлечённых в создание и использование информационных систем, распространение информации.

**Предметные результаты изучения включают:**

1. владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;
2. овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
3. владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
4. владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
5. сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, алгоритмах анализа этих объектов, о кодировании и декодировании данных и причинах искажения данных при передаче; систематизацию знаний, относящихся к математическим объектам информатики; умение строить математические объекты информатики, в том числе логические формулы;
6. сформированность умения работать с библиотеками программ; наличие опыта использования компьютерных средств представления и анализа данных.

**Формы отслеживания и фиксации образовательных результатов:**

Отслеживание результатов усвоения Программы это постоянный процесс, выражающийся в постоянном оценивании. В зависимости от этапа обучения используется диагностическое (входное, текущее) и срезовое (тематическое, промежуточное, итоговое) оценивание.

**Стартовое оценивание** (входной мониторинг) – направлено на выявление знаний, умений и навыков, обучающихся. Оно позволяет определить наличный (исходный) уровень знаний и умений, чтобы

использовать его как фундамент, ориентироваться на допустимую сложность учебного материала.

**Формы аттестации:** - зачет, решение индивидуальной задачи, тестирование, выполнение самостоятельных и творческих работ, создание и защита учебных проектов.

**Результат** – входной мониторинг.

**Текущее оценивание** – осуществляется в повседневной работе с целью проверки усвоения предыдущего материала и выявления пробелов в знаниях обучающихся. Ведущая задача текущего контроля – регулярное управление учебной деятельностью обучающихся и ее корректировка.

Система оценивания выстраивается таким образом, чтобы обучающиеся включались в контрольно-оценочную деятельность, приобретали навыки и привычку к самооценке (самооценка в конце каждого занятия).

**Тематическое оценивание** – осуществляется периодически, по мере прохождения новой темы, раздела, и имеет целью систематизацию знаний обучающихся.

**Формы аттестации:** наблюдение, тестирование, самостоятельное решение задач, написание проекта.

**Результат:** эти виды контроля позволяет получить непрерывную информацию о ходе и качестве усвоения учебного материала и на основе этого оперативно вносить изменения в учебный процесс.

**Промежуточное оценивание** (промежуточный мониторинг) – выявление результатов обучения за 1 полугодие. Оценка уровня подготовки обучаемых в этом случае производится с помощью контрольных заданий в форме: олимпиады, решение индивидуальной задачи, тестирование, выполнение самостоятельных и творческих работ, создание и защита учебных проектов и т.д.

**Формы аттестации:** - контрольное задание в форме: зачета, олимпиады, решение индивидуальной задачи, тестирование, выполнение самостоятельных и творческих работ, создание и защита учебных проектов и т.д.

**Результат** – промежуточный мониторинг.

**Итоговое оценивание** – проводится в конце всего учебного года, а также по окончании обучения по программе. Оно направлено на проверку конкретных результатов обучения, выявления степени усвоения обучающимися системы знаний, умений и навыков, полученных в процессе изучения Программы.

**Формы аттестации:** - контрольное задание в форме: зачета, олимпиады, решение индивидуальной задачи, тестирование, выполнение самостоятельных и творческих работ, создание и защита учебных проектов и т.д.

**Результат** – итоговый мониторинг.

Измерителями перечисленных в Программе **компетенций** являются следующие показатели:

- активное использование сервисов Всероссийской школьной образовательной сети (электронный журнал, виртуальное общение со всеми участниками образовательного процесса, участие в форумах, в дистанционных группах и событиях);
- дополнительное изучение выбранного дистанционного курса по программированию на сайте: <http://www.eduinformatika.ru/moodle>
- участие во Всероссийской олимпиаде школьников по информатике;
- участие в конкурсах и олимпиадах разных форм и уровней.

При оценке результатов освоения программы выделяются **разные уровни усвоения материала:**

- **Высокий** – решает «сверхзадачи», для которых требуются либо самостоятельно добытые знания, либо новые, самостоятельно усвоенные умения и действия, требуемые на следующих ступенях программы, всегда принимает участие в олимпиадах по информатике на различных уровнях, получает призовые места.

- **Выше среднего** – решает «сверхзадачи», для которых требуются либо самостоятельно добытые знания, либо новые, самостоятельно усвоенные умения и действия, требуемые на следующих ступенях программы, всегда принимает участие в олимпиадах по информатике на различных уровнях, иногда получает призовые места.

- **Средний уровень** – решает нестандартные задачи, где требуется, либо действие в новой, непривычной ситуации, либо использование новых, усваиваемых в данный момент знаний, часто принимает участие в олимпиадах.

- **Ниже среднего** - решает типовые задачи, подобные тем, что решали уже много раз, где требуются отработанные действия и усвоенные знания, эпизодически принимает участие в олимпиадах.

- **Низкий уровень** - затрудняется в решении типовых задач, подобные тем, что решали уже много раз, где требуются отработанные действия и усвоенные знания, эпизодически принимает участие в олимпиадах.

## 2. Учебно-тематический план.

I год обучения. Программирование на Python (216ч)

№	Название раздела	Всего часов	Теория	Практика
1	Введение в программирование. Базовые понятия. Основные понятия	2	1	1
2	Простые (линейные) программы	2	1	1
3	Разветвляющийся алгоритм	4	1	3
4	Циклический алгоритм	4	1	3
5	Списки (массивы) и кортежи	4	1	3
6	Функции	4	1	3
7	Строки	4		3
8	Файлы	4	1	3
9	Графический интерфейс Python - Tkinter и GUI-инструментарий Tk.	4	1	3
10	Подготовка творческих работ с использованием графического интерфейса	4	1	3
	Итого	36		

### III. Содержание программы I год обучения.

#### Программирование на Python

##### 1. Введение в программирование. Базовые понятия.

**Основные понятия.** Понятие алгоритма и его свойства. Исполнитель алгоритмов: назначение, среда исполнителя, система команд исполнителя, режимы работы. Языки для записи алгоритмов (язык блок-схем, учебный алгоритмический язык). Линейные, ветвящиеся и циклические алгоритмы. Вспомогательные алгоритмы. Метод пошаговой детализации.

**Рассматриваемые вопросы.** Знакомство педагога с детьми и детей друг с другом. Цели и задачи работы объединения на текущий учебный год. Вводный инструктаж: знакомство учащихся с правилами работы в кабинете информатики, акцентирование их внимания на опасные моменты, с которыми можно столкнуться в процессе работы, и сообщение о соответствующих мерах предосторожности. Работа с учебным исполнителем алгоритмов; составление линейных, ветвящихся и циклических алгоритмов управления исполнителем; составление алгоритмов со сложной структурой; использование вспомогательных алгоритмов (процедур, подпрограмм).

**Требования к уровню подготовки обучающихся и планируемые результаты.**

Обучающийся должен знать:

Технику безопасности и правилам поведения в компьютерном классе.

**Среда программирования Python. Основные понятия языка Python**  
(16 часов)

**Основные понятия.** Строка и опция меню, компиляция, алгоритм, программа, алфавит.

**Рассматриваемые вопросы.** Инструктаж по основным этапам решения задач на компьютере. Алгоритм. Типы алгоритмов. Свойства алгоритма. Способы записи алгоритмов. Язык программирования Python. Знакомство с интегрированной средой Python. Структура экрана. Команды меню.

**Требования к уровню подготовки обучающихся и планируемые результаты.**

Обучающийся должен знать:

- историю появления языка Python и причины его популярности,
- основные файлы среды Python,
- назначение основных горячих клавиш,
- структуру

программы.

Обучающийся должен

уметь:

- запускать программу,
- сохранять файлы,

- пользоваться опциями меню среды Python.

## 2. Простые (линейные) алгоритмы

**Основные понятия.** Тип переменной (integer, float, char, string, boolean), операторы присваивания, чтения и записи (=, input, print), позиционный вывод целых и вещественных чисел, операции целочисленного деления (//, %), стандартные встроенные модули (import math, import random.)

**Рассматриваемые вопросы.** Основные определения (программа, алфавит, идентификатор, служебные слова, переменные, константы, комментарий). Классификация типов данных. Структура программы. Первая программа. Целочисленные типы данных. Динамическая типизация. Стандартные функции и процедуры для данных целого типа. Оператор присваивания. Оператор вывода print. Ввод данных. Оператор input. Вещественные типы данных в Паскале. Стандартные функции и процедуры для данных вещественных типов. Форматированный вывод. Арифметические выражения. Состав и правила записи на Python арифметических выражений. Блок-схема линейного алгоритма.

**Требования к уровню подготовки обучающихся и планируемые результаты.**

Обучающийся должен знать:

- описание простых типов данных,
- действия над переменными числового типа данных,
- правила линейной записи арифметических выражений,
- порядок выполнения операций,
- требования к форматированию текста программы,
- команды ввода и вывода информации,
- правила применения стандартных

подпрограмм.

Обучающийся должен уметь:

- правильно применять операторы input и print,
- составлять диалог компьютера и пользователя,
- выполнять форматированный вывод чисел,
- преобразовывать сложные арифметические выражения в линейную форму,
- правильно использовать стандартные подпрограммы.

## 3. Разветвляющийся алгоритм

**Основные понятия** Ветвление, условный оператор if...else, оператор выбора elif..else..end. Полный, неполный условный оператор, вложенность условных операторов. Условие, составное условие, логические операции (or, and, not), логические выражения.

**Рассматриваемые вопросы.** Ветвление. Условный оператор if. Блок-схема разветвляющегося алгоритма. Логический тип данных. Логические операции. Нахождение значений логических выражений. Вложенность условий. Составные условия. Оператор выбора elif.

### ***Требования к уровню подготовки обучающихся и планируемые результаты.***

Обучающийся должен знать:

- синтаксис операторов условия и выбора,
- формы условного оператора,
- основные логические операции и таблицы истинности,
- правила построения и вычисления логических

выражений.

Обучающийся должен уметь:

- вычислять значения логических выражений,
- решать задачи на применение условного оператора и оператора выбора,
- строить блок-схемы алгоритма решения задач.

#### **4. Циклический алгоритм**

**Основные понятия** Цикл, циклический алгоритм, цикл с предусловием (while..do), цикл с постусловием (repeat..until), цикл со счетчиком (for..in ..range), переменная цикла, тело цикла, заикливание, условие выполнения (окончания) цикла, вложенные циклы, простое число, совершенное число, автоморфное число, числа-близнецы.

**Рассматриваемые вопросы.** Циклы. Цикл с предусловием while. Целочисленная арифметика. Работа с окнами среды программирования. Поиск ошибок с помощью отладчика. Работа с клавиатурой. Стандартные функции read key и key pressed, их применение в циклических алгоритмах. Алгоритм Евклида. Блок-схема циклического алгоритма. Цикл с параметром.

### ***Требования к уровню подготовки обучающихся и планируемые результаты.***

Обучающийся должен знать:

- синтаксис операторов цикла,
- отличия между циклами for,

while, repeat.

Обучающийся должен уметь:

- заменять один вид цикла на другой,
- решать задачи с использованием вложенных циклов,
- выполнять работу программы по шагам,
- строить блок-схемы решения задач.

#### **5. Списки (массивы) и кортежи**

**Основные понятия** Тип списка, элемент списка(массива), индекс элемента.

**Рассматриваемые вопросы.** Понятие массива. Одномерные массивы. Способы задания. Доступ к элементам массива. Поиск элементов массива. Удаление элементов из массива. Вставка элементов в массив. Перестановка элементов.

### ***Требования к уровню подготовки обучающихся и планируемые результаты.***

Обучающийся должен знать:

- определения массива, индекса элемента,
- особенности использования перечисляемого и ограниченного типа данных,
- описание и способы задания массива,
- алгоритмы вставки и удаления элемента массива.

Обучающийся должен уметь:

- использовать стандартные алгоритмы анализа элементов массива,
  - вставлять и удалять несколько элементов массива по условию,
  - менять местами элементы массива,
  - использовать в решении задачи несколько массивов.
- Проверка знаний, умений и навыков в форме по выбору обучающегося.

## 6. Функции

**Основные понятия.** Метод пошаговой детализации, процедура, функция, стек, локальные и глобальные переменные, формальные и фактические параметры, процедура без параметров, процедура с параметрами, механизм передачи параметров, передача значений по ссылке и по значению, рекурсия, косвенная рекурсия, рекурсивный алгоритм.

**Рассматриваемые вопросы.** Понятие подпрограммы. Процедуры и функции. Формальные и фактические параметры. Вызов по ссылке и по значению. Локальные и глобальные переменные и подпрограммы. Понятие рекурсии. Примеры задач рекурсивного решения. Косвенная рекурсия.

**Требования к уровню подготовки обучающихся и планируемые результаты.**

Обучающийся должен знать:

- метод пошаговой детализации,
- нисходящий способ программирования,
- особенности использования процедур и функций,
- правила использования рекурсии.

Обучающийся должен уметь:

- разбивать задачу на подзадачи,
- описывать переменные и параметры подпрограммы,
- устанавливать связи между основной программой и подпрограммами.

## 7. Строки

**Основные понятия** Строковый тип данных (String), стандартные подпрограммы для работы со строкой (pos, insert, val, str, copy, delete, len), таблица кодов ASCII.

**Рассматриваемые вопросы.** Тип данных char. Операции над символами. Тип данных string. Строковые переменные, их описание. Длина строки. Операции над строками. Стандартные функции для работы со



строками (concat, copy, length, pos, upcase). Стандартные процедуры для работы со строками (delete, insert, str, val). Бегущая строка. Осыпающиеся буквы.

### ***Требования к уровню подготовки обучающихся и планируемые результаты.***

Обучающийся должен знать:

- представление символов в таблице кодов ASCII,
- операции над символами, строками,
- определение и описание строки,
- правила использования стандартных подпрограмм для работы со строкой.

Обучающийся должен уметь:

- применять операции и подпрограммы работы со строкой при решении задач,
- использовать возможности текстового режима для нестандартного вывода символов при решении задач творческого характера.

## **8. Файлы**

***Основные понятия*** Типизированные, текстовые и нетипизированные файлы.

***Рассматриваемые вопросы.*** Файлы, виды, примеры решения задач. Типизированные файлы, особенности описания и применения при решении задач. Текстовые файлы, особенности описания и применения при решении задач. Нетипизированные файлы, особенности описания и применения при решении задач.

### ***Требования к уровню подготовки обучающихся и планируемые результаты***

Обучающийся должен знать:

- виды файлов,
- особенности описания файлов,
- основные алгоритмы работы с файлами разных видов.

Обучающийся должен уметь выбрать необходимый вид файла и применить при решении задач различной сложности.

## **9. Графический интерфейс Python - Tkinter и GUI-инструментарий Tk.**

***Рассматриваемые вопросы.*** Импорт библиотеки. Создание главного окна. Создание виджет. Установка их свойств. Определение событий. Определение обработчиков событий. Расположение виджет на главном окне. Отображение главного окна

## **10. Подготовка творческих работ с использованием графического интерфейса.**

***Рассматриваемые вопросы.*** Систематизация полученных знаний, умений и навыков. Консультация

**Итоговое занятие. Зачет.** Проверка знаний, умений и навыков в форме по выбору обучающегося.

#### IV.Календарно тематическое планирование

№	ТЕМА	ДАТА
<b>Блок I. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ. БАЗОВЫЕ ПОНЯТИЯ</b>		
<b>Тема 1. Основы алгоритмизации. Основы ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PASCAL</b>		
<b>Язык программирования. Программа.</b>		
1.	Алгоритмизация и требования к алгоритму. Способы записи алгоритмов. Линейные алгоритмы.	
2.	Язык программирования. Состав языка программирования . Компиляция и интерпритация. Типы данных и способы внутреннего представления данных	
<b>Тема 3. Простые (линейные) программы.</b>		
3.	Пример учебной программы. Оператор присваивания.	
4.	Арифметические выражения. Арифметические операции.	
5.	Ввод и вывод данных.	
6.	Примеры программ Вычисления по формулам.	
<b>Тема 4. Программирование разветвлений.</b>		
7.	Логические выражения.	
8.	Составной оператор.	
9.	Условный оператор.	
10.	Оператор выбора.	
<b>Тема 5. Циклические программы.</b>		
11.	Цикл с постусловием.	
12.	Цикл с предусловием.	
13.	Цикл с параметром.	
14.	Вложенные циклы.	
<b>Тема 6. Массивы. Списки.</b>		
15.	<b>СПИСКИ (МАССИВЫ).</b> Описание списков. Инициализация массивов начальными значениями.	
16.	Заполнение массива данными. Вывод элементов массива. Методы списков. Генераторы списков.	
17.	Турнир по программированию «Осенний LIST»	
18.	Списки. Функции <i>map()</i> , <i>zip()</i> , <i>filter()</i> и <i>reduce()</i> .	
19.	Добавление и удаление элементов списка. Поиск элемента в списке.	
20.	Выбор элементов случайным образом. Сортировка списка.	

21.	Преобразование списка в строку.	
<b>Тема 7. Обработка строк текста.</b>		
22.	<i>Строки. Создание строки. Специальные символы. Операции над строками.</i>	
23.	Функции и методы для работы со строками. Изменение регистра символов.	
24.	Функции для работы с символами. Поиск и замена в строке.	
25.	Функции для работы с символами. Поиск и замена в строке.	
<b>Функции.</b>		
26.	Пользовательские функции. Создание функции и ее вызов.	
27.	Расположение определений функций.	
28.	Рекурсия. Вычисление факториала.	
29.	Глобальные и локальные переменные.	
30.	Вложенные функции.	
31.	Решение олимпиадных задач	
32.	Решение олимпиадных задач	
<b>Тема 8. Файлы</b>		
33.	Некоторые сведения о файловой системе.	
34.	Имя и расширение файла Каталоги Устройства.	
35.	Описание файлового типа. Виды файлов. Файловая переменная.	
36.	Общая схема работы с файлами.	

## **V. Материально-техническое обеспечение программы.**

Материально-техническое обеспечение:

- компьютерный класс с 10 - 12 персональными компьютерами для обучающихся;
- локальная сеть с доступом в Интернет;
- проектор и демонстрационный экран (или демонстрационный монитор);
- доска
- МФУ (принтер, сканер, копир)
- Интерактивная доска

### **ЭОР**

- Мультимедийные презентации «Изучаем Pascal»
- Мультимедийные презентация «Изучаем Python»
- Электронные учебники:
  - Piton\_Unkn.pdf
  - Program\_Python.pdf
  - Python\_3\_Proh.pdf
  - Python\_Bizli.pdf
  - Python\_Nahaev.pdf
- Марк Лутц - Изучаем Python - 4-е изд -2011.djvu
- Саммерфилд - Программирование на Python 3.2009.djvu
- Д.В. Душистов - Решение 50 типовых задач по программированию на языке Pascal – 2012

### **Программное обеспечение:**

- Windows-7
- Ms. Office
- 8python-3.4.2.msi
- 9wingide-101-5.0.9-1.exePascal ABC.Net

### **Учебники и методические пособия**

1. С. В. Феськов «Программирование на языке C++»
2. А.А. Галахов «Самоучитель для детей. Программирование от нуля до....»
3. Э.С. Ларина «Олимпиадные задачи с решениями. Информатика »
4. Н. А. Владимирова «Увлекательная информатика. Логические задачи, кроссворды, ребусы, игры»
5. В. Д. Паронджанов «Занимательная информатика»
6. А.Ф. Чернов «Для преподавателей. Задачи для подготовки к олимпиадам. Информатика »

## VI. Список литературы.

### Рекомендуемая литература для педагога

1. Кирюхин В. М. Методика проведения и подготовки к участию в олимпиадах по информатике: всероссийская олимпиада школьников, Москва, БИНОМ, 2011 г.
2. Кетков А., Кетков Ю. Свободное программное обеспечение. PYTHON для студентов и школьников. - БХВ-Петербург, 2012.
3. Митрофанов С. П., Проскурякова С. Г. Учебно-методический комплект «Программирование».
4. Окулов С. М. «Основы программирования», Москва, БИНОМ, 2010 г.

### Рекомендуемая литература для обучающихся

1. Алексеев Е.Р. Python и Lazarus. Учебник по программированию. SBN 978-5-94074-611-9; 2010 г.

### Интернет-ресурсы

- Дистанционный курс «Программирование на Паскале (базовый курс)». <http://www.eduinformatika.ru/moodle>.
- Национальный открытый университет [www.intuit.ru](http://www.intuit.ru).
- Олимпиады по информатике (ХМАО-Югра) <http://acmu.ru/>