

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа» с. Койгородок
(МБОУ «СОШ» с. Койгородок)

«Шёр школа» Койгорт сиктса муниципальной велёдан съёмкуд учреждение

РАССМОТРЕНО
на заседании ШМО учителей
физики и информатики
(протокол № 1 от 29 августа
2023 г.)

Руководитель ШМО
Кос С.Г.Коскокова

СОГЛАСОВАНО
Заместитель директора (ВР)
И.Г. Кононенко И.Г.Кононенко

УТВЕРЖДЕНО
Директор МБОУ "СОШ"
с.Койгородок Э.А.Кораева
приказ № 359-од от 30
августа 2023 г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 043FDDEE40032AFD5BB4DEF5D92A4B19AA8
Владелец: Кораева Эльвира Александровна
Действителен: с 18.10.2022 до 18.10.2023

Дополнительная общеобразовательная программа – дополнительная
общеразвивающая программа по Точке роста
«Мастерская программирования на языке PYTHON»

НАПРАВЛЕННОСТЬ: ТЕХНИЧЕСКАЯ

Уровень программы: ознакомительный

Возраст обучающихся: 13 – 16 лет

Срок реализации: 2 года

Составитель: Бессонов А.Г.,
педагог дополнительного образования

с. Койгородок, 2023г.

ОГЛАВЛЕНИЕ

1. Пояснительная записка	2 стр
2. Учебный (тематический) план. Первый год обучения	9 стр
3. Содержание учебного (тематического) плана.	
Первый год обучения	14 стр
4.Календарный учебный график первый год обучения	21стр
5. Учебный (тематический) план. Второй год обучения	34 стр
6. Содержание учебного (тематического) плана.	
Второй год обучения	41стр
7.Календарный учебный график второго года обучения	49 стр
8. Формы контроля и оценочные материалы	56 стр
9. Условия реализации Программы	57 стр
10.Комплекс организационно-педагогических условий, включающий формы аттестации	58 стр
11. Список литературы	60 стр
12.Приложение	61 стр

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа «Мастерская программирования на языке PYTHON» (далее – Программа) разработана в соответствии с требованиями:

- Федерального Закона от 29.12.2012г. №273-ФЗ «Об образовании в Российской Федерации»;
- Концепцией развития дополнительного образования, утвержденной распоряжением Правительства Российской Федерации от 04.09.2014г. №1726-р;
- СанПин 2.4.3648-20 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей», утвержденный постановлением Главного государственного санитарного врача РФ;
- Приказа Министерства просвещения России от 09.11.2018г. №196г. Москва «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным программам».
- Устава МБОУ «СОШ» с.Койгородок

Данная программа **технической направленности**.

Python – язык программирования общего назначения, оптимизированный для создания качественного программного обеспечения. Язык Python один из самых используемых языков программирования в мире в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и др.

Программа «Мастерская программирования на языке Python» (далее – Программа) технической направленности ознакомительного уровня, ориентирован на то, чтобы каждый учащийся мог эффективно использовать современные компьютерные технологии в учебной, творческой, самостоятельной и досуговой деятельности. Программа способствует развитию познавательных интересов и творческих способностей учащихся, удовлетворению их потребностей в интеллектуальном, нравственном, физическом совершенствовании, она имеет практическую направленность по развитию IT компетентности.

Актуальность Программы обусловлена быстрым внедрением компьютерной техники в повседневную жизнь, переходом к новым технологиям обработки информации. Изучая программирование, учащиеся лучше понимают возможности и границы применения компьютеров. К ним приходит осознание того, что компьютер является инструментом, управляемым людьми. Не все учащиеся станут профессиональными программистами, но все выигрывают от того, что постигли природу программирования и научились создавать собственные программы.

Новизна Программы заключается в построении индивидуальной образовательной траектории учащегося, в приобретении им знаний, востребованных на рынке труда, в повышении самооценки и осознании перспектив будущей жизни, дальнейшей социализации.

Педагогическая целесообразность Программы заключается в том, что занятие

программированием, даёт необычайно сильный толчок для развития интеллекта учащихся, формирует их логическое мышление, вырабатывает привычку аккуратной и систематической работы.

Отличительная особенность Программы состоит в том, что она является мощным образовательным инструментом, позволяющим дать учащимся навыки по программированию на языке Python. Важным аспектом Программы является использование дистанционных форм обучения.

Реализация Программы, основана на деятельностном подходе, более 60% времени отводится практической деятельности, способствующей развитию активной познавательной деятельности, творчества и достижению высоких результатов в области информационно-коммуникационных технологий.

Цель Программы – ознакомить учащихся с возможностями, синтаксисом, технологией языка Python и обучить методами программирования для решения прикладных математических и информационных задач.

Реализация поставленной цели предусматривает решение ряда задач.

Задачи Программы

Обучающие:

- сформировать представление об основах программирования в среде Python;
- сформировать навыки грамотной работы в системе программирования Python;
- ознакомить с базовыми понятиями теории алгоритмов при решении математических задач;
- обучить методам решения задач, реализуемым на языке Python;
- сформировать практические навыки решения прикладных задач;
- сформировать навыки поиска информации, работы с технической литературой.

Развивающие:

- развить навыки самостоятельного и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- развить алгоритмическое и логическое мышление учащихся;
- развить творческие способности учащихся, их потребность в самореализации;
- развить интеллектуальные и практические умения, самостоятельно приобретать и применять на практике полученные знания;

Воспитательные:

- содействовать воспитанию устойчивого интереса к изучению программирования;
- содействовать воспитанию информационной культуры;
- формировать потребность в творческой деятельности, стремление к самовыражению через техническое творчество;
- содействовать воспитанию интереса профессиям, связанным с программированием.

Категория учащихся

Обучение по Программе ведется в разновозрастных группах, которые комплектуются из учащихся 13-16 лет, имеющих начальные умения и навыки работы в текстовых редакторах, в Skype и Google, с почтовой программой, с браузером, использовании встроенной панели рисования для отображения блок-схем.

Рекомендуемое количество учащихся в группе – 10 человек.

Сроки реализации

Программа рассчитана на два года обучения. Общее количество часов в год

составляет 68 часов.

Формы и режим занятий

Программа реализуется через дистанционное обучение – онлайн-занятия и очном режиме 1 раз в неделю по 2 часа. Программа включает в себя теоретические и практические занятия.

Формы организации деятельности – групповая, индивидуально групповая, индивидуальная, по подгруппам.

Планируемые результаты освоения Программы

По итогам **первого года** обучения учащиеся будут **знать**:

- технику безопасности и требования, предъявляемые к организации рабочего места
- термины: «данные», «кодирование», «алгоритм», «программа», понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- методы представления и алгоритмы обработки данных, программную реализацию алгоритмов;
- основы алгоритмической культуры, нормы информационной этики;
- математические и компьютерные модели, их использование;
- виды программного обеспечения, решаемые с его помощью задачи;
- вредоносное программное обеспечение и средства защиты от него;

По итогам **первого года** обучения учащиеся будут **уметь**:

- составлять несложные программы;
- работать с основными видами программных систем и интернет сервисов;
- определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирая основания и критерии для классификации;
- устанавливать причинно-следственные связи, делать логическое умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- составлять алгоритмы управления исполнителями и записывать их на выбранном алгоритмическом языке;
- использовать логические значения, операции и выражения с ними;
- выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов, простых и табличных величин;
- записывать алгоритмические структуры на языке программирования Python;
- решать простые, сложные и нестандартные задачи.

По итогам **второго года** обучения учащиеся будут **знать**:

- этапы разработки программы: алгоритмизация, кодирование, отладка, тестирование и их содержание;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;

- алгоритмы и программы на языке Python для решения простых, сложных и нестандартных задач в математической области;
- стандартизацию в сфере информационно-коммуникационных технологий;
- способы представления данных в зависимости от поставленной задачи.

По итогам второго года обучения, учащиеся будут **уметь:**

- создавать программы для решения несложных алгоритмических задач в выбранной среде программирования;
- использовать готовые прикладные компьютерные программы и сервисы в выбранной специализации;
- использовать Python для решения задач из области математики, физики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- использовать основные алгоритмические приемы при решении математических задач;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

СОДЕРЖАНИЕ ПРОГРАММЫ
Учебный (тематический) план первого года обучения

№	Названия раздела/темы	Количество часов			Формы аттестации и контроля
		Всего	Теория	Практика	
1.	Введение в язык программирования Python	6	2	4	
1.1	Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж	2	1	1	Тест
1.2	Элементы языка. Структура программы. Операции и переменные. Типы данных	2	1	1	
1.3	Ввод и вывод данных на языке Python. Ввод и отладка программ в среде	2	-	2	Текущий контроль. Практикум
2.	Линейные алгоритмы и их реализация на языке Python	10	2	8	
2.1	Линейные алгоритмы целочисленных данных и их реализация на Python	2	1	1	
2.2	Решение задач по теме «Обработка целочисленных данных»	2	-	2	Текущий контроль. Практикум
2.3	Линейные алгоритмы вещественных чисел и их реализация на Python управления	2	1	1	
2.4	Решение задач по теме «Обработка вещественных чисел»	2		2	Текущий контроль. Практикум

2.5	Самостоятельная работа по темам раздела «Линейные алгоритмы и их реализация в среде Python»	2		2	Текущий контроль. Самостоятельная работа
3	Разветвляющиеся алгоритмы и их реализация на языке Python	14	4	10	
3.1	Алгоритм «выбор», графическое изображение, полное и неполное ветвление	2	1	1	
3.2	Реализация алгоритма «выбор» на Python. Примеры решения задач	2	1	1	
3.3	Решение задач по теме «Полное и неполное ветвление»	2		2	Текущий контроль. Практикум
3.4	Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления	2	1	1	
3.5	Множественный выбор и его реализация с помощью вложенных ветвлений	2	1	1	
3.6	Решение задач по теме «Сложные условия. Каскадные ветвления»	2		2	Текущий контроль. Практикум
3.7	Контрольная работа по темам раздела: «Разветвляющиеся алгоритмы и их реализация на языке Python»	2		2	Промежуточный контроль.
4	Циклические алгоритмы и их реализация на Python	22	4	18	

4.1	Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом	2	1	1	
4.2	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием»	4		4	Текущий контроль. Практикум
4.3	Циклические алгоритмы. Цикл с постусловием. Инструкции управления	2	1	1	
4.4	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»	2	-	2	Текущий контроль. Практикум
4.5	Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием»	2	-	2	Текущий контроль. Самостоятельная работа
4.6	Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом	2	1	1	
4.7	Решение задач по теме «Циклические алгоритмы. Цикл с параметром»	2	-	2	Текущий контроль. Практикум
4.8	Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком»	2	-	2	Текущий контроль. Самостоятельная работа
4.9	Сложные циклические алгоритмы. Вложенные циклы	2	1	1	
4.10	Самостоятельная работа по разделу: «Циклические алгоритмы и их реализация на Python». Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
5.	Этапы решения задач на языке Python	4	1	3	

5.1.	Последовательное конструирование алгоритма	2	1	1	
5.2.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма	2	-	2	Текущий контроль. Практикум
6.	Вспомогательные алгоритмы и их реализация на языке Python	8	2	6	
6.1	Вспомогательные алгоритмы и подпрограммы. Функции	2	1	1	
6.2	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций	2	-	2	Текущий контроль. Практикум
6.3	Вспомогательные алгоритмы и подпрограммы. Рекурсия	2	1	1	
6.4	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии	2	-	2	Текущий контроль. Практикум
7.	Итоги обучения	4	1	3	
7.1	Итоговая самостоятельная работа по теме: «Реализация основных типов алгоритмов»	2	-	2	Итоговый контроль. Зачетная работа
7.2	Подведение итогов обучения	2	1	1	
	ИТОГО	68	16	52	

Содержание учебного (тематического) плана первого года обучения

Раздел 1. Введение в язык программирования Python

Тема 1.1. Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж

Теория. Структура образовательной программы, её цель и задачи, содержание обучения. Основные правила и требования техники безопасности при работе за компьютером. Понятие алгоритма и программы.

Практика. Первичная диагностика. Тестирование. Установка, работа в среде программирования Python.

Тема 1.2. Элементы языка. Структура программы. Операции и переменные. Типы данных

Теория. Знакомство с основными типами переменных, синтаксисом языка программирования, основными процедурами ввода исходных данных и вывода результатов

Практика. Отработка процедур ввода исходных данных и вывода результатов. Тестирование.

Тема 1.3. Ввод и вывод данных на языке Python. Ввод и отладка программ в среде

Практика. Практикум по созданию элементарных программ ввода вывода данных, работа со средой, отладка программ. Анализ возможных синтаксических ошибок.

Раздел 2. Линейные алгоритмы и их реализация на языке Python

Тема 2.1. Линейные алгоритмы целочисленных данных и их реализация на языке Python

Теория. Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над ними и ресурсы оперативной памяти. Выполнение операции присваивания в оперативной памяти (далее – ОП) компьютера.

Практика. Создание блок-схемы линейного алгоритма. Запись арифметических выражений на языке программирования.

Тема 2.2. Решение задач по теме «Обработка целочисленных данных»

Практика. Практикум по разработке линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел. Анализ готовых линейных программ.

Тема 2.3. Линейные алгоритмы вещественных чисел и их реализация на Python

Теория. Различные типы данных, допустимые операциях над вещественными числами и ресурсы оперативной памяти.

Практика. Разбор типичных задач с линейной структурой алгоритма. **Тема 2.4. Решение задач по теме «Обработка вещественных чисел»**

Практика. Практикум по реализации линейных алгоритмов вещественных чисел. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел.

Тема 2.5. Самостоятельная работа по темам раздела: «Линейные алгоритмы и их реализация в среде Python»

Практика. Самостоятельная работа по составлению линейного алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

Раздел 3. Разветвляющиеся алгоритмы и их реализация на языке Python

Тема 3.1. Алгоритм «выбор», графическое изображение, полное и неполное ветвление

Теория. Понятие алгоритма «выбор», графическое изображение.

Практика. Составление алгоритма «полное ветвление».

Тема 3.2. Реализация алгоритма «выбор» на Python. Примеры решения задач

Теория. Типовые задачи, использующие алгоритм «выбор», изображение алгоритма в виде блок-схемы.

Практика. Ввод и отладка программ в среде Python.

Тема 3.3. Решение задач по теме «Полное и неполное ветвление»

Практика. Практикум по реализации алгоритмов «полное ветвление» и «неполное ветвление». Ввод и отладка программ в среде Python.

Тема 3.4. Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления

Теория. Составление и анализ алгоритмов.

Практика. Ввод и отладка программ в среде Python.

Тема 3.5. Множественный выбор и его реализация с помощью вложенных ветвлений

Теория. Понятие множественного выбора, изображение на блок-схеме.

Практика. Ввод и отладка программ в Python.

Тема 3.6. Решение задач по теме «Сложные условия. Каскадные ветвления»

Практика. Практикум по составлению алгоритмов для решения задач со сложными условиями и каскадными ветвлениями. Ввод и отладка программ в среде Python.

Тема 3.7. Контрольная работа по темам раздела «Разветвляющиеся алгоритмы и их реализация на языке Python»

Практика. Контрольная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ работы

Раздел 4. Циклические алгоритмы и их реализация на языке Python

Тема 4.1. Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом

Теория. Понятие цикла с предусловием, графическое изображение.

Практика. Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.2. Решение задач по теме «Циклические алгоритмы. Цикл с предусловием»

Практика. Практикум по составлению блок-схем, написанию программного кода, вводу и отладке программ, реализующих цикл с предусловием в среде Python.

Тема 4.3. Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом

Теория. Понятие цикла с постусловием, графическое изображение.

Практика. Ввод и отладка простейших программ, реализующих цикл с постусловием в среде Python.

Тема 4.4. Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»

Практика. Практикум по решению типовых задач с помощью цикла с

постусловием. Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с постусловием в Python.

Тема 4.5. Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием»

Практика. Самостоятельная работа по решению типовых задач с помощью цикла с предусловием. Ввод и отладка программ, реализующих цикл с предусловием в среде Python. Анализ составленного алгоритма.

Тема 4.6. Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом

Теория. Понятие цикла с параметром, графическое изображение.

Практика. Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.7. Решение задач по теме: «Циклические алгоритмы. Цикл с параметром»

Практика. Практикум по разработке алгоритмов цикла с параметром, анализу логических ошибок. Ввод и отладка программ, реализующих цикл с параметром в среде Python.

Тема 4.8. Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком».

Практика. Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклы с параметром в среде Python. Анализ самостоятельной работы.

Тема 4.9. Сложные циклические алгоритмы. Вложенные циклы. Решение задач

Теория. Понятие вложенного цикла, графическое изображение.

Практика. Решение задач по разработке и отладке программ, реализующих вложенные циклы в среде Python. Анализ разработанного алгоритма. Оценка эффективности разработанного алгоритма.

Тема 4.10. Самостоятельная работа по темам раздела «Циклические алгоритмы и их реализация на языке Python». Анализ самостоятельной работы

Практика. Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклические алгоритмы в среде Python.

Раздел 5. Этапы решения задач на языке Python

Тема 5.1. Последовательное конструирование алгоритма

Теория. Этапы и особенности решения задачи на компьютере. Этап создания алгоритма. Использование принципа последовательного конструирования алгоритма. Будет также рассмотрен принцип его работы.

Практика. Разработка алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

Тема 5.2. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма

Практика. Практикум по последовательному конструированию алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

Раздел 6. Вспомогательные алгоритмы и их реализация на языке Python

Тема 6.1. Вспомогательные алгоритмы и подпрограммы. Функции

Теория. Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Типовые задачи.

Практика. Ввод и отладка программ с использованием функции в среде Python.

Тема 6.2. Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций

Практика. Практикум по программированию решения отдельных подзадач с помощью отдельных функций, которые потом при необходимости вызываются в различных местах программы. Польза функций при решении задач. Важные принципы в программировании – модульность и повторное использование кода. Их польза при разработке сложных программ.

Тема 6.3. Вспомогательные алгоритмы и подпрограммы. Рекурсия Теория. Понятие рекурсии, в чём её сложность. Рекурсивные алгоритмы

и их особенности: когда нужно использовать рекурсию в программировании, а когда лучше обойтись без этого.

Практика. Ввод и отладка программ с использованием рекурсии в среде Python.

Тема 6.4. Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии

Практика. Практикум по решению типовых задач с вводом и отладкой программ с использованием рекурсии в среде Python.

Раздел 7. Итоги обучения

Тема 7.1. Итоговая самостоятельная работа по теме «Реализация основных типов алгоритмов»

Практика. Зачетная работа по составлению алгоритма, написанию программы, вводу и отладке программного кода.

Тема 7.2. Подведение итогов обучения

Теория. Обзор пройденного материала.

Практика. Анализ результатов работы.

Учебный (тематический) план второго года обучения

№	Названия раздела/темы	Количество часов			Формы аттестации и контроля
		Всего	Теория	Практика	
1.	Повторение. Основные типы алгоритмов	8	4	4	
1.1.	Основные типы алгоритмов. Реализация линейного алгоритма и выбора на языке Python. Решение задач	2	1	1	Текущий контроль. Тест
1.2.	Основные типы алгоритмов. Реализация циклического алгоритма на языке Python. Решение задач	2	1	1	Текущий контроль. Практикум
1.4.	Реализация вложенных циклов на языке Python. Решение задач	2	1	1	
1.3.	Основные типы алгоритмов. Реализация вспомогательных алгоритмов на языке Python. Решение задач	2	1	1	Текущий контроль. Практикум
2.	Структурные типы данных. Списки	8	3	5	
2.1.	Структурные типы данных. Списки. Решение задач	2	1	1	
2.2.	Структурные типы данных. Обработка списков. Линейный поиск. Добавление и удаление элементов. Копирование списков. Решение задач	2	1	1	

2.3.	Структурные типы данных. Обработка списков. Двоичный поиск и сортировка. Решение задач	2	1	1	
2.4.	Решение задач по теме «Структурные типы данных. Обработка списков»	2	-	2	Текущий контроль. Практикум
3.	Структурные типы данных. Символьные строки	8	2	6	
3.1.	Структурные типы данных. Символьные строки. Операции обработки строк	2	1	1	
3.2.	Структурные типы данных. Символьные строки. Типовые задачи обработки строк	2	1	1	
3.3.	Решение задач по теме «Символьные строки. Операции обработки строк»	2	-	2	Текущий контроль. Практикум
3.4.	Самостоятельная работа по темам раздела «Структурные типы данных. Символьные строки». Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
4.	Структурные типы данных. Матрицы	8	2	6	
4.1.	Структурные типы данных. Матрицы. Основные понятия	2	1	1	
4.2.	Структурные типы данных. Матрицы. Типовые задачи обработки элементов	2	1	1	

4.3.	Решение задач по теме: «Структурные типы данных. Матрицы. Обработка элементов. Обработка исключений»	2	-	2	Текущий контроль. Практикум
4.4.	Зачетная работа по темам раздела «Структурные типы данных. Матрицы». Анализ самостоятельной работы	2	-	2	Промежуточный контроль.
5	Структурные типы данных. Работа с файлами	6	1	5	
5.1.	Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла	2	1	1	
5.2.	Решение задач по теме «Изменение файлов»	2	-	2	Текущий контроль. Практикум
5.3.	Самостоятельная работа по темам раздела «Структурные типы данных. Работа с файлами». Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
6.	Объектно-ориентированное программирование (ООП)	12	4	8	
6.1.	Объектно-ориентированное программирование (ООП). Концепция ООП. Объекты и классы	2	1	1	
6.2.	Объектно-ориентированное программирование (ООП). Программирование объектной модели. Решение задач	2	1	1	Текущий контроль. Практикум
6.3.	Принципы объектно ориентированного программирования (ООП): инкапсуляция	2	1	1	

6.4.	Принципы объектно ориентированного программирования (ООП): наследование и полиморфизм	2	1	1	
6.5.	Решение задач по теме «Принципы объектно ориентированного программирования (ООП)»	2	-	2	Текущий контроль. Практикум
6.6.	Самостоятельная работа по теме «Объектно ориентированное программирование (ООП)», Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
7.	Приложения с графическим пользовательским интерфейсом	14	6	8	
7.1.	Основы графического интерфейса. Решение задач	2	1	1	
7.2.	Создание приложения с использованием виджетов графического интерфейса. Решение задач	2	1	1	
7.3.	Создание графического интерфейса без использования программы визуализатора. Решение задач	2	1	1	
7.4.	Управление макетом графического интерфейса: блочный макет. Решение задач	2	1	1	
7.7.	Модель обработки данных в приложении с графическим интерфейсом. Решение задач	2	1	1	

7.8.	Представление в приложении с графическим интерфейсом. Создание собственного виджета	2	1	1	Текущий контроль. Практикум
7.9.	Самостоятельная работа по темам раздела «Приложения с графическим пользовательским интерфейсом». Анализ проектов	2	-	2	Текущий контроль. Самостоятельная работа
8.	Итоги обучения	4	1	3	
8.1.	Итоговая самостоятельная работа по темам Программы	2	-	2	Итоговый контроль. Зачетная работа
8.2.	Обзор пройденного материала. Подведение итогов обучения	2	1	1	
	ИТОГО	68	23	45	

Содержание учебного (тематического) плана второго года обучения

Раздел 1. Повторение. Основные типы алгоритмов

Тема 1.1. Основные типы алгоритмов. Реализация линейного алгоритма и выбора на языке Python. Решение задач

Теория. Типы алгоритмов и их графическое изображение.

Практика. Решение задач по реализации линейного алгоритма на языке Python

Тема 1.2. Основные типы алгоритмов. Реализация циклического алгоритма на языке Python. Решение задач

Теория. Реализация циклического алгоритма на Python.

Практика. Решение задач по реализации циклического алгоритма на языке Python

Тема 1.3. Основные типы алгоритмов. Реализация вложенных циклов на языке Python. Решение задач

Теория. Вложенные циклы, графическое изображение.

Практика. Решение задач по реализации вложенных циклов на языке Python.

Тема 1.4. Основные типы алгоритмов. Реализация вспомогательных алгоритмов на языке Python

Теория. Реализация вспомогательных алгоритмов на Python.

Практика. Решение задач по реализации вспомогательных циклов на языке Python.

Раздел 2. Структурные типы данных. Списки

Тема 2.1. Структурные типы данных. Списки. Решение задач

Теория. Понятие списка, его польза в программировании, особенности списка. Заполнение списка значениями. Вывод списка на экран. Способы создания «ввода-вывода» элементов списков в программах на языке Python.

Практика. Решение задач на ввод и отладку программ со списками.

Тема 2.2. Структурные типы данных. Обработка списков. Линейный поиск. Добавление и удаление элементов. Копирование списков. Решение задач

Теория. Обработка списков. Типовые задачи. Возможности. Добавление и удаление элементов списков. Линейный поиск элемента в списке.

Практика. Решение задач на ввод и отладку программ по линейному поиску, добавлению и удалению элементов, копированию списков.

Тема 2.3. Структурные типы данных. Обработка списков. Двоичный поиск и сортировка. Решение задач

Теория. Методы поиска и сортировки в списках. Стандартные алгоритмы обработки списков. Сортировка и её назначение. Алгоритмы сортировки пузырьком. Рекурсивный алгоритм быстрой сортировки элементов списка. Метод поиск элемента в отсортированном списке.

Практика. Решение задач на ввод и отладку программ по поиску и сортировке в списках.

Тема 2.4. Решение задач по теме «Структурные типы данных. Обработка списков»

Практика. Практикум по разработке стандартных алгоритмов обработки списков.

Тема 2.5. Самостоятельная работа по темам раздела «Структурные типы данных. Списки». Анализ самостоятельной работы

Практика. Самостоятельная работа по составлению алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

Раздел 3. Структурные типы данных. Символьные строки

Тема 3.1. Структурные типы данных. Символьные строки. Операции обработки строк

Теория. Понятие символьных строк, как немутуирующих объектов. Строковые литералы. Строковые функции, методы и операторы.

Практика. Основные операции со строками, которые позволяют быстро преобразовывать любые последовательности символов.

Тема 3.2. Структурные типы данных. Символьные строки. Типовые задачи обработки строк

Теория. Основные операции обработки строк: конкатенация (склеивание) строк, повторение строки, форматирование. Характеристика и классификация типовых задач по обработке строк.

Практика. Ввод и отладка основных операций обработки строк в среде Python.

Тема 3.3. Решение задач по теме: «Символьные строки. Операции обработки строк»

Практика. Практикум по проведению операций и решению типовых задач обработки строк в среде Python.

Тема 3.4. Самостоятельная работа по темам раздела «Структурные типы данных. Символьные строки». Анализ самостоятельной работы

Практика. Самостоятельная работа по составлению алгоритмов, вводу и отладке программ в среде Python, анализу созданных проектов.

Раздел 4. Структурные типы данных. Матрицы

Тема 4.1. Структурные типы данных. Матрицы. Основные понятия Теория. Понятие матрицы. Методы ввода элементов.

Практика. Реализация матриц на языке Python и особенности их хранения.

Тема 4.2. Структурные типы данных. Матрицы. Типовые задачи обработки элементов

Теория. Анализ типовых задач обработки элементов матрицы.

Практика. Обработка элементов матриц в среде Python.

Тема 4.3. Решение задач по теме «Структурные типы данных. Матрицы. Обработка элементов. Обработка исключений»

Практика. Практикум по решению задач с применением свойств матриц в среде Python.

Тема 4.4. Зачетная работа по темам раздела «Структурные типы данных. Матрицы». Анализ самостоятельной работы

Практика. Контрольная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ синтаксических и логических ошибок.

Раздел 5. Структурные типы данных. Работа с файлами

Тема 5.1. Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла

Теория. Типы файлов, этапы обработки, режимы открытия. Программирование файлового ввода/вывода. Текстовые и двоичные файлы. Этапы обработки файла. Основные инструменты обработки файлов. Методы считывания данных из файла.

Практика. Программирование файлового ввода/вывода в среде Python.

Тема 5.2. Решение задач по программированию обработки файлов

Практика. Практикум по программированию обработки файлов в среде Python.

Тема 5.3. Самостоятельная работа по темам раздела «Структурные типы данных. Работа с файлами». Анализ самостоятельной работы

Практика. Самостоятельная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ составленного алгоритма.

Раздел 6. Объектно-ориентированное программирование (ООП)

Тема 6.1. Объектно-ориентированное программирование (ООП). Концепция ООП. Объекты и классы

Теория. Последовательное конструирование алгоритма. Объектная модель решения задачи. Объектно-ориентированный анализ (ООА). Этапы ООА. Концепция ООП.

Практика. Конструирование алгоритма методом объектно ориентированного анализа.

Тема 6.2. Объектно-ориентированное программирование (ООП).

Программирование объектной модели. Решение задач

Теория. Абстракция. Поля и методы класса. Программирование объектов и классов.

Практика. Программирование объектной модели.

Тема 6.3. Принципы объектно-ориентированного программирования (ООП): инкапсуляция

Теория. Определение инкапсуляции. Применение инкапсуляции в программировании.

Практика. Программирование объектной модели, реализация класса в конкретной задаче.

Тема 6.4. Принципы объектно-ориентированного программирования (ООП): наследование и полиморфизм

Теория. Принципы объектно-ориентированного программирования (ООП): наследование и полиморфизм, применение их в программировании.

Практика. Программирование объектной модели, реализация класса в конкретной задаче с использованием наследования и полиморфизма.

Тема 6.5. Решение задач по теме: «Принципы объектно ориентированного программирования (ООП)»

Практика. Практикум по разработке проектов с использованием принципов объектно-ориентированного программирования (ООП). Анализ разработанных проектов.

Тема 6.6. Самостоятельная работа по разделу: «Объектно ориентированное программирование (ООП)». Анализ самостоятельной работы

Практика. Самостоятельная работа по разработке проектов с использованием метода объектно-ориентированного программирования(ООП).Анализ разработанных проектов.

Раздел 7. Приложения с графическим пользовательским интерфейсом

Тема 7.1. Основы графического интерфейса. Решение задач

Теория. Современные прикладные программы. RAD-среды. Простейшая программа с графическим интерфейсом.

Практика. Работа с графическими библиотеками для языка Python.

Тема 7.2. Создание приложения с использованием виджетов графического интерфейса. Решение задач

Теория. Элементы управления: кнопки, метки, поля ввода. Создание и настройка окон сообщений.

Практика. Создание приложения с использованием виджетов графического интерфейса.

Тема 7.3. Создание графического интерфейса без использования программы-визуализатора. Решение задач

Теория. Создание окна приложения и запуск цикла обработки событий. Размещение на форме и настройка виджетов.

Практика. Создание графического интерфейса без использования программы-визуализатора.

Тема 7.4. Управление макетом графического интерфейса: блочный макет. Решение задач

Теория. Макет графического интерфейса. Типы макетов. Блочный макет.

Практика. Управление блочным макетом графического интерфейса.

Тема 7.5. Управление макетом графического интерфейса: сеточный макет. Решение задач

Теория. Принцип работы сеточного макета. Программирование сеточного макета. Работа программы при изменении размеров ее окна.

Практика. Управление сеточным макетом графического интерфейса.

Тема 7.6. Рисование на форме. Решение задач

Теория. Рисование при создании графического интерфейса. Инструменты для рисования.

Программирование рисования. Классы для рисования в графической библиотеке PyQt5

Практика. Программирование рисования на форме в среде Python.

Тема 7.7. Модель обработки данных в приложении с графическим интерфейсом.

Решение задач

Теория. Декомпозиция приложения с графическим интерфейсом. Модель обработки данных приложения «Калькулятор».

Практика. Обработка данных в приложении с графическим интерфейсом.

Тема 7.8. Представление в приложении с графическим интерфейсом

Теория. Представление приложения «Калькулятор». Контроллер в приложении с графическим интерфейсом.

Практика. Создание приложения «Калькулятор».

Тема 7.9. Создание собственного виджета

Практика. Программирование собственного виджета. Создание виджета для побитового отображения целых чисел.

Тема 7.10. Самостоятельная работа по темам раздела «Приложения с графическим пользовательским интерфейсом». Анализ проектов

Практика. Самостоятельная работа по созданию приложений с графическим пользовательским интерфейсом. Анализ выполненных проектов.

Раздел 8. Итоги обучения

Тема 8.1. Итоговая самостоятельная работа по темам Программы

Практика. Зачетная работа по составлению алгоритмов, написанию программ, вводу и отладке программных кодов.

Тема 8.2. Обзор пройденного материала. Подведение итогов обучения

Теория. Обзор пройденного материала.

Практика. Анализ результатов работы.

Форма контроля и оценочные материалы

Результативность обучения обеспечивается применением различных форм, методов и приемов, которые тесно связаны между собой и дополняют друг друга. Большая часть занятий отводится практической работе, в ходе практической деятельности педагог тактично контролирует, советует, направляет учащихся. Содержание и объем материала, подлежащего проверке, определяется Программой. При проверке усвоения материала необходимо выявлять полноту, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Текущий контроль уровня усвоения материала осуществляется по окончании изучения каждой темы – выполнением практических заданий. Промежуточный контроль проходит в середине учебного года в форме контрольной работы. Итоговый контроль проходит в конце учебного года – в форме зачета, на котором обучающиеся представляют свои работы и обсуждают их.

Формы проведения аттестации:

- тестирование;
- практикум;
- контрольная работа;
- зачетная работа.

Условия реализации программы

Реализация Программы строится на дистанционных формах обучения и очных занятиях.

В каждом модуле, рассчитанном на одну учебную неделю, предлагаются:

- учебные материалы;
- видеоуроки;
- практические задания;
- задачник;
- индивидуальные задания к занятию;
- методические материалы для преподавателя.

В нулевом блоке располагаются форумы, дополнительные сетевые материалы для учащихся, выставка работ.

Управление ходом деятельности ученика осуществляется посредством перекрестных гиперссылок, взаимодействие – через программу Skype (iChat, TeamViewer, Join.Me). Контроль деятельности ученика частично в онлайн режиме осуществляется с помощью общего доступа через Skype или TeamViewer. Показ и контроль осуществляется через веб-камеры.

Занятие включает в себя иллюстрированное изложение теоретического материала с демонстрацией примеров, после каждого теоретического занятия учащемуся предлагается выполнить практическое задание, которое затем проверяется и рецензируется преподавателем. Выполненные задания ученики выкладывают в ресурс «задание», законченные творческие работы в тематический форум-выставку. Материалы теоретической и практической частей разработаны в текстовом и частично в видео форматах.

Теоретические занятия предполагают:

- лекционные формы (материал выкладывается в сети),
- уроки-беседы,
- демонстрационные формы и др.

Практические занятия предполагают:

- самостоятельную работу учащихся (основа практических занятий);
- совместную работу групп (группы объединяет общая цель работы, ее тема, содержание и сроки сдачи материала преподавателю);
- работу с материалами лекций, дополнительными источниками информации;
- анализ собственных работ;
- групповое и индивидуальное консультирование;
- подготовка и защита индивидуальных и групповых проектов.

Подобная организация обучения способствует развитию познавательной активности и творческих способностей учащихся.

Комплекс организационно-педагогических условий, включающий формы аттестации

Материально-технические условия реализации Программы

Для успешной реализации Программы необходимо, что бы рабочее место учащегося и преподавателя включали в себя:

- компьютеры, обеспечивающие возможность работы с системой программирования Python - 10;
- периферийное оборудование:
 - ✓ принтер (черно/белой печати, формата А4) - 1;
 - ✓ устройства для ввода визуальной информации (сканер) – 1;
 - ✓ акустические колонки - 1;
 - ✓ оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).
- программное обеспечение компьютера:
 - ✓ операционная система семейства MacOS или Windows;
 - ✓ программа Python;
 - ✓ файловый менеджер (в составе операционной системы или др.);
 - ✓ почтовый клиент (в составе операционных систем или др.);
 - ✓ браузер (в составе операционных систем или др.);
 - ✓ мультимедиа проигрыватель (в составе операционной системы или др.);
 - ✓ антивирусная программа;
 - ✓ программа-архиватор;
 - ✓ программа интерактивного общения;
 - ✓ текстовый редактор,
 - ✓ растровый графический редактор;
 - ✓ звуковой редактор;
 - ✓ редактор Web-страниц.

Список литературы

1. Бэрри П. Изучаем программирование на Python. – Москва: Эксмо, 2017. – 624 с.
2. Бриггс, Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. — М.: Манн, Иванов и Фербер, 2017. — 320 с.
3. Буйначев, С. К. Основы программирования на языке Python: учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 91, [1] с.
4. Лутц, Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2019.—832с.
5. Лутц, Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2020. — 720с.
6. Программирование для детей на языке Python. – М.: Издательство АСТ, 2017.
7. Слаткив Бретт. Секреты Python: 59 рекомендаций по написанию эффективного кода.: Пер. с англ. -М.: ООО ии.д. Вильяме», 2016. - 272 с.
8. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М.: Издательство Юрайт, 2019. — 161 с.
9. Шуман Х.-Г. Python для детей / пер. с нем. М. А. Райтман. – М.: ДМК Пресс, 2019. – 344 с.

ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

№	Наименование модуля	Предмет оценивания	Формы и методы оценивания	Показатель и оценивания	Критерии оценивания	Вид контроля/ аттестации
1	1 год обучения	Базовый уровень проверки общих знаний и понимания правил в области основ программирования	Собеседование	<p>- Владение терминологией.</p> <p>Коммуникативные умения.</p> <p>Познавательная активность.</p>	<p>Высокий- 5 баллов Средний- 4-3 балла Низкий-2- 0 балла</p> <p>Высокий – инициативен со сверстниками и педагогом, указывает другим как надо делать что-то. Средний – не инициативен в общении, однако проявляет общительность в ответ на чужую инициативу. Низкий - не вступает в контакт со сверстниками.</p> <p>Высокий – ребенок предлагает выраженный интерес к предлагаемым заданиям, сам задает вопросы, прилагает усилия к преодолению трудностей. Средний – ребенок активно включается в работу, но при первых же трудностях</p>	Входящий
		Уровень освоения теоретических и практических знаний по модулям программы	Практическая работа, протокол	<p>Основы видов деятельности</p> <p>Соответств</p>	<p>Высокий– легко ориентируется в эффективно использовать современные компьютерные технологии в учебной, творческой, самостоятельной и досуговой деятельности ИТ компетентности Средний – совместно с педагогом определяет вид, в котором будет работать. Низкий – не ориентируется в изученных видах</p> <p>Высокий – умеет самостоятельно следовать</p>	Итоговый

				<p>ие структуре видах жизнедеятельности</p> <p>Владение терминологией</p> <p>Следование алгоритму действий</p>	<p>структуре вида; Средний – самостоятельно, но требуя дополнительных подсказок со стороны педагога Низкий – не может без помощи педагога следовать структуре выбранного вида.</p> <p>Высокий – использует изученные алгоритмы в предлагаемых заданиях Средний – опираясь на записи в тетради, применяет данные правила; Низкий – не применяет</p> <p>Высокий – самостоятельно выполняет алгоритм действий ИТ компетентности . Средний - в сотрудничестве с педагогом следует алгоритму. Низкий - алгоритму действий не следует.</p>	
		Сформированность регулятивных УУД (авт. Г.В.Репкина, Е.В.Заика).	Наблюдение, диагностическая карта	<p>Действия целеполагания</p> <p>Действия планирования</p>	<p>Высокий - самостоятельно ориентируется в практических заданиях, учебная задача удерживается и регулирует весь процесс выполнения задания. Средний - ориентируется в практических заданиях с помощью педагога, осознает, что надо делать и что сделал в процессе решения практической задачи. Низкий - способен принимать только простейшие задания.</p> <p>Высокий – может совместно с педагогом планировать последовательность выполнения задания и успешно самостоятельно работать по плану. Средний– в сотрудничестве с педагогом</p>	В начале года (октябрь) и в конце года (май).

				<p>учащийся способен выделить учебные действия, необходимые для решения учебной задачи; способен работать по предложенному плану при незначительном контроле педагога.</p> <p>Низкий – Копирует действия педагога, плохо осознавая их направленность и взаимосвязь, самостоятельно работать по предложенному педагогом плану не может.</p> <p>Высокий – находит, исправляет и объясняет ошибки после решения задачи; в многократно повторенных действиях ошибок не допускает.</p> <p>Средний – заметив ошибку, обучающийся не может обосновать своих действий; сделанные ошибки исправляет неуверенно.</p> <p>Низкий – не может обнаружить и исправить ошибку даже по просьбе педагога, некритично относится к исправленным ошибкам в своей работе.</p>	
			<p>Действия контроля и коррекции</p>	<p>Высокий – находит, исправляет и объясняет ошибки после решения задачи; в многократно повторенных действиях ошибок не допускает.</p> <p>Средний – заметив ошибку, обучающийся не может обосновать своих действий; сделанные ошибки исправляет неуверенно.</p> <p>Низкий – не может обнаружить и исправить ошибку даже по просьбе педагога, некритично относится к исправленным ошибкам в своей работе.</p>	
			<p>Действие оценки.</p>	<p>Высокий – умеет самостоятельно оценить свои действия, показать правильность или ошибочность результата, соотносит со схемой действия.</p> <p>Средний – не умеет самостоятельно оценить свои действия, но испытывает потребность в получении оценки со стороны педагога; может оценить действия других учащихся.</p> <p>Низкий – не умеет, не пытается и не испытывает потребности оценивать свои действия – ни самостоятельно, ни по просьбе педагога.</p>	
			<p>Саморегул</p>	<p>Высокий – выполняет и заканчивает действие</p>	

				<p>яция.</p>	<p>в требуемый временной момент, способен тормозить свои импульсивные поведенческие реакции</p> <p>Средний – помнит, но не всегда выполняет и заканчивает действия в требуемый временной момент не всегда может сдерживать свои импульсивные поведенческие реакции на занятие</p> <p>Низкий – не выполняет и не стремится выполнить задание до конца, не способен сдерживать свои импульсивные поведенческие реакции на занятии.</p>	
		<p>Сформированность познавательных УУД (авт. Г.В.Репкина, Е.В.Заика).</p>	<p>Наблюдение, диагностическая карта</p>	<p>Умение добывать новые знания, находить ответы на вопросы, используя учебник и информацию полученную на занятии.</p> <p>Умение делать выводы.</p>	<p>Высокий – способен самостоятельно и быстро находить необходимую информацию для выполнения учебных заданий.</p> <p>Средний – самостоятельно, но требуя дополнительных указаний со стороны педагога, находит необходимую информацию для выполнения учебных занятий.</p> <p>Низкий – не может без помощи педагога найти необходимую информацию для выполнения учебных занятий</p> <p>Высокий – способен при незначительной поддержке педагога сделать выводы по результатам работы.</p> <p>Средний – совместно с педагогом или одноклассниками может сделать выводы по результатам работы.</p> <p>Низкий – даже при незначительной помощи со стороны педагога не может сделать выводы по результатам работы.</p>	<p>В начале года (октябрь) и в конце года (май).</p>

		Сформированность коммуникативных УУД (авт. Г.В.Репкина, Е.В.Заика).	Наблюдение, диагностическая карта	<p>Умение работать в паре и группе.</p> <p>Умение оформлять свою мысль в устной речи.</p> <p>Сформированность норм в общении с детьми и взрослыми.</p>	<p>Высокий – согласует свой способ действия с другими; сравнивает способы действия и координирует их, строя совместное действие.</p> <p>Средний – приходит к согласию относительно способа действия при участии педагога; испытывает затруднения в координации совместного действия.</p> <p>Низкий – не пытается договориться или не может прийти к согласию, настаивая на своем.</p> <p>Высокий – умеет оформлять свою мысль в устной речи на уровне небольшого текста.</p> <p>Средний – умеет оформлять свою мысль в устной речи на уровне одного предложения.</p> <p>Низкий – не умеет самостоятельно оформлять свою мысль в устной речи.</p> <p>Высокий – знает и соблюдает нормы общения с детьми и взрослыми.</p> <p>Средний – знает но иногда не соблюдает нормы общения с детьми и взрослыми.</p> <p>Низкий – не знает и не соблюдает нормы общения с детьми и взрослыми.</p>	В начале года (октябрь) и в конце года (май)
		Сформированность личностных УУД (авт. Г.В.Репкина, Е.В.Заика).	Наблюдение, диагностическая карта	Сформированность учебно-познавательного интереса	<p>Высокий – проявляет устойчивый интерес к информационным технологиям, старательно и с желанием выполняет любые задания педагога.</p> <p>Средний – проявляет интерес преимущественно к программированию, учащийся проявляет познавательную активность преимущественно лишь в сотрудничестве с педагогом.</p> <p>Низкий – обнаруживает безразличное или негативное отношение к информатике,</p>	

				<p>неохотно включается в выполнение заданий, не принимает помощь со стороны педагога, охотно выполняет лишь привычные действия, чем осваивает новые.</p> <p>Высокий – во всем реально оценивает себя, свои достижения и возможности (допустима чуть сниженная самооценка).</p> <p>Средний – в основном реально оценивает себя, свои возможности (допустима чуть завышенная самооценка).</p> <p>Низкий – чрезмерно завышенная или сниженная самооценка, некритичность к своему поведению.</p>	
			<p>Самооценка</p>	<p>Высокий – всегда сопереживает и стремится сразу оказать помощь другим.</p> <p>Средний – способен к сопереживанию, но сразу оказать помощь другим не стремится.</p> <p>Низкий – переживает только собственные неудачи и безразлично относится к проблемам других.</p>	
			<p>Эмоциональная отзывчивость</p>		

