

**Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа № 1
р. п. Самойловка Самойловского района Саратовской области»**

Принято на педагогическом совете
МБОУ «СОШ № 1 р. П. Самойловка»
Протокол № 1 от 25 августа 2023 г.



**Дополнительная общеобразовательная общеразвивающая программа
технической направленности «Программирование на языке Python»
для детей 15-17 лет**

Срок реализации дополнительной общеобразовательной
общеразвивающей программы 1 год

Автор-составитель:
педагог дополнительного образования
Центра цифрового и гуманитарного
профилей «Точка роста»:
Дудникова И.Е.

Самойловка
2023 год

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1 Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность и разработана в соответствии с Положением «О порядке разработки и утверждения дополнительных общеобразовательных общеразвивающих программ».

Актуальность программы обучения. Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других.

Сегодня Python один из самых популярных языков программирования, области его применения только расширяются. Последние несколько лет он входит в ТОП-3 самых востребованных языков на рынке IT.

Также по информации порталов по трудоустройству в России в течение последних 5 лет Python-разработчики востребованы на рынке труда, специалистов в этой сфере до сих пор не хватает.

Освоив данную программу обучения, в будущем школьники приобретут престижную профессию, востребованную на сегодняшнем рынке труда. В этом и заключается актуальность данной программы.

Педагогическая целесообразность данной Программы заключается в том, что занятие программированием, даёт необычайно сильный толчок для развития интеллекта обучающихся, формирует их логическое мышление, вырабатывает привычку аккуратной и систематической работы.

Отличительные особенности программы в том, что она является практико-ориентированной. Освоение подростками IT-навыков происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать IT-технологиями, что поможет им самоопределиваться и выстроить траекторию личностного роста в современном информационном обществе.

Научившись программировать на языке Python, обучающиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Адресат программы: обучающиеся в возрасте 15-17 лет.

Возрастные особенности обучающихся:

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» предназначена для школьников в возрасте 15–17 лет, мотивированных к обучению и обладающих системным мышлением.

Подростки этого возраста отличаются открытием своего внутреннего мира, внутреннего «Я». Главным измерением времени в самосознании является будущее, к которому он (она) себя готовит. Ведущая деятельность в этом возрасте — учебно-профессиональная, в процессе которой формируются такие новообразования, как мировоззрение, профессиональные интересы, самосознание, мечта и идеалы, поэтому основной формой проведения занятий выбраны практические занятия.

Также следует отметить, что подростки данной возрастной группы характеризуются такими психическими процессами, как стремление углублённо понять себя, разобраться в своих чувствах, настроениях, мнениях, отношениях. Это порождает у подростка стремление к самоутверждению, самовыражению (проявления себя в тех качествах, которые он считает наиболее ценными) и самовоспитанию. Эти процессы позволяют положить начало созданию начального профессионального самоопределения обучающихся.

Форма обучения – очная. Программа предусматривает 2 вида занятий: теоретические занятия и практические занятия. В обучении применяется групповая форма с индивидуальным подходом, включающая обучение в малых группах.

Объем и срок освоения Программы:

Программа рассчитана на 72 часа. Срок освоения программы 1 год.

Режим занятий: занятия проводятся 1 раз в неделю по 2 академических часа с перерывом в 10-15 минут. Количество обучающихся в группе 10-12 человек.

1.2 Цель и задачи программы:

Цель программы – получение обучающимися базовых знаний, умений и навыков в области программирования на языке Python.

Задачи:

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 обучающихся, 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.

1.3

Планируемые результаты

Предметные результаты:

В рамках курса «Программирование на языке Python» учащиеся овладевают следующими **знаниями, умениями** и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.
- формирование понятий «алгоритм», «программа»;

- формирование понятий об основных конструкциях языка программирования Python;
- формирование понятий о структурах данных языка программирования Python;
- формирование основных приёмов составления программ в программировании на языке программирования Python;

Метапредметные результаты:

- формирование самостоятельности и творческого подхода к решению задач;
- формирование и развитие алгоритмического и логического мышления у школьников;
- формирование и развитие творческих способностей у обучающихся;
- формирование интеллектуальных и практических умений.

Личностные результаты:

- формирование интереса к изучению языка программирования Python;
- формирование информационной культуры;
- формирование стремления к самовыражению через техническое творчество;
- формирование интереса к программированию и профессиям IT-сферы.

1.4 Содержание программы Учебный план

№ п/п	Наименование темы	Количество часов			Формы контроля
		Всего	Теорети- ческие занятия	Практи- ческие занятия	
	Вводное занятие. Инструктаж по ТБ при работе за ПК.	2	1	1	Опрос
1.	Раздел 1. Знакомство с языком Python	3	1	2	Тестирование, решение практических задач
1.1	Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	1	1		
1.2	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	1		1	

1.3	Тест № 1. Знакомство с языком Python	1		1	
2.	Раздел 2. Переменные и выражения	7	1	6	Тестирование, решение практических задач
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	1		1	
2.2	Практическая работа: Переменные	1		1	
2.3	Занятие 4. Выражения Практическая работа: Выражения	1		1	
2.4	Занятие 5. Ввод и вывод		1		
2.5	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	1		1	
2.6	Тест № 2. Выражения и операции.	1		1	
3.	Раздел 3. Условные предложения	14	5	9	Тестирование, решение практических задач
3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	1	1		
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	2	1	1	
3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление	2	1	1	

3.4	Занятие 10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	1		1		
3.5	Самостоятельная работа № 1 по теме «Условные операторы».	5	2	3		
3.6	Занятие 11. Зачетная работа № 1. "Составление программ с ветвлением".	2		2		
3.7	Тест № 3. "Условные операторы".	1		1		
4.	Раздел 4. Циклы	17	3	14		Тестирование, решение практических задач, творческая работа
4.1	Занятие 12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	1	1		
4.2	Занятие 13. Оператор цикла for Практическая работа Решение задачи с циклом for.	2	1	1		
4.3	Занятие 14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	2	1	1		
4.4	Занятие 15. Случайные числа Практическая работа: Случайные числа	1		1		
4.5	Занятие 16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	3		3		
4.6	Самостоятельная работа № 2 "Составление программ с циклом"	4		4		

4.7	Тест № 4. Циклы	1		1	Тестирование, решение практических задач
4.8	Занятие 17. Творческая работа № 1. "Циклы"	2		2	
5.	Раздел 5. Функции	8	3	5	
5.1	Занятие 18. Создание функций Практическая работа Создание функций	1	1		
5.2	Занятие 19. Локальные переменные Практическая работа Локальные переменные	1		1	
5.3	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	1		1	
5.4	Самостоятельная работа № 3 по теме "Функции"	2		2	
5.5	Занятие 21. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	1	1	1	
5.6	Тест № 5. Функции	1	1		
6.	Раздел 6. Строки - последовательности символов	5	1	4	
6.1	Занятие 22. Строки Практическая работа: Строки	1	1		
6.2	Занятие 23. Срезы строк	1		1	
6.3	Занятие 24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	3		3	

7.	Раздел 7. Сложные типы данных	9	4	5	Тестирование, решение практических задач
7.1	Занятие 25. Списки Практическая работа: Списки	1	1		
7.2	Занятие 26. Срезы списков	1	1		
7.3	Занятие 27. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2	1	1	
7.4	Занятие 28. Матрицы	1	1		
7.5	Тест № 7. Списки	1		1	
7.6	Занятие 29. Кортежи	1		1	
7.7	Занятие 30. Введение в словари	1		1	
7.8	Занятие 31. Множества в языке Python	1		1	
8.	Раздел 8. Стиль программирования и отладка программ	8	4	4	Решение практических задач, презентация проекта, рефлексия
8.1	Занятие 32. Стиль программирования	1	1	1	
8.2	Занятие 33. Отладка программ	1	1	1	
8.3	Занятие 34. Зачет по курсу «Программирование на языке Python»	1	1	1	
8.4	Занятие 35. Что дальше?	1	1	1	
ИТОГО:		72	23	49	

Содержание учебного плана

Вводное занятие

Теория: Знакомство с планом работы объединения, инструктаж по ТБ.

Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python
Практическая работа 1.2. Режимы работы с Python
Тест № 1. Знакомство с языком Python

Учащиеся должны знать / понимать: – понятие программы; – структура программы на Python; – режимы работы с Python.

Учащиеся должны уметь: – выполнить установку программы; – выполнить простейшую программу в интерактивной среде; – написать комментарии в программе.

Раздел 2. Переменные и выражения

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест № 2. Выражения и операции.

Учащиеся должны знать / понимать: – общую структуру программы; – типы данных; – целые, вещественные типы данных и операции над ними; – оператор присваивания.

Раздел 3. Условные предложения

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения

Практическая работа 3.2. "Условный оператор"

Практическая работа 3.3. Множественное ветвление

Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы".

Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

Учащиеся должны знать / понимать: – назначение условного оператора; – способ записи условного оператора; – логический тип данных; – логические операторы or, and, not;

Учащиеся должны уметь: – использовать условный оператор; – создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи"

Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом.

Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы Творческая работа № 1. "Циклы"

Учащиеся должны знать / понимать: – циклы с условием и их виды; – правила записи циклов условием; – назначение и особенности использования цикла с параметром; – формат записи цикла с параметром; – примеры использования циклов различных типов.

Учащиеся должны уметь: – определять вид цикла, наиболее удобный для решения поставленной задачи; – использовать цикл с условием; – определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции"

Тест № 5. Функции

Учащиеся должны знать / понимать: – понятие функции; – способы описания функции; – принципы структурного программирования; – понятие

локальных переменных подпрограмм; – понятие формальных и фактических параметров подпрограмм; – способ передачи параметров.

Учащиеся должны уметь: – создавать и использовать функции; – использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать / понимать: – назначение строкового типа данных; – операторы для работы со строками; – процедуры и функции для работы со строками; – операции со строками.

Учащиеся должны уметь: – описывать строки; – соединять строки; – находить длину строки; – вырезать часть строки; – находить подстроку в строке; – находить количество слов в строке.

Раздел 7. Сложные типы данных

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками.

Тест № 7. Списки

Учащиеся должны знать / понимать: – сложные типы данных; – способ описания списка; – способ доступа к элементам списка; – способ описания кортежа; – способ описания словаря; – операции, выполняемые со списками, кортежами и словарями; – понятие множества; – способы описания множества; – операторы работы с множествами.

Учащиеся должны уметь: – описывать списки; вводить элементы списка; – выводить элементы списка; – выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; – использовать вложенные списки; – приводить примеры использования вложенных списков (матриц); – описывать множества; – определять принадлежность элемента множеству; – вводить элементы множества; – выводить элементы множества.

Раздел 8. Стил ь программирования и отладка программ

Стил ь программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать: – что такое стил ь программирования; – правила именования объектов; – основные рекомендации при написании программ.

Учащиеся должны уметь: – определять вид ошибок и находить ошибки в программе. – выполнять тестирование и отладку программ.

1.5 Формы аттестации и их периодичность

Для отслеживания результативности образовательной деятельности по проводятся: текущий, промежуточный и итоговый контроль.

Текущий контроль – оценка уровня и качества освоения тем/разделов программы и личностных качеств обучающихся. Осуществляется на занятиях в течение всего времени обучения по данной программе.

Промежуточный контроль – оценка уровня и качества освоения обучающимися дополнительной общеразвивающей программы по итогам изучения раздела, темы.

Итоговый контроль – оценка уровня и качества освоения обучающимися дополнительной общеразвивающей программы по завершению учебного модуля или всего периода обучения по программе.

Основными формами проверки знаний обучающихся является:

- устный опрос;
- беседа;
- наблюдение;
- практическая работа на ПК;
- защита проекта.

Текущий контроль усвоения материала осуществляется путем устного опроса или беседы. Задачей устного опроса является не столько оценивание знаний обучающихся, сколько определение проблемных мест в усвоении материала и фиксирование внимания учеников на сложных понятиях, явлениях, процессе. Оценка за устный опрос не выставляется.

Промежуточный контроль проводится в форме практических работ на ПК. По результатам проверки практической работы оценка не выставляется.

Педагог озвучивает обучающимся ошибки, допущенные в процессе практической работы, и дает рекомендации по их исправлению.

Итоговый контроль проводится в форме защиты индивидуальных проектов. На итоговом занятии обучающиеся презентуют свои проекты. По итогам защиты проекта выставляется «Зачет».

2. Комплекс организационно-педагогических условий

2.1 Методическое обеспечение

Занятия по программе «Программирование на языке Python» проводятся 1 раз в неделю по 2 часа. Образовательный процесс организуется в соответствии с возрастными особенностями обучающихся, с применением разнообразных форм и методов обучения. На занятиях используется групповая и индивидуальная работа под руководством педагога. Занятия в учебном кабинете состоят из теоретической и практической части.

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой

проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

2.2 Условия реализации программы

Материально-техническое обеспечение:

Материально-техническое обеспечение: – столы для компьютера; – компьютерные стулья; – шкафы для дидактических материалов, пособий; – специальная и научно-популярная литература для педагога и учащихся; – канцтовары;

Информационное обеспечение: – персональный компьютер (на каждого участника); – мультимедийный проектор; – видеоматериалы разной тематики по программе; – оргтехника; – выход в сеть Internet;

Аппаратное обеспечение: – Процессор не ниже Core2 Duo; – Объем оперативной памяти не ниже 4 ГбDDR3; – Дисковое пространство на менее 128 Гб; – Монитор диагональю на менее 19”;

Программное обеспечение: – Операционная система Windows 10; – Интерпретатор Python версии 4.3; – IDE JetBrains PyCharm; – Foxit Reader или другой просмотрщик PDF файлов; – WinRAR; – Пакет офисных программ; – Любой браузер для интернет серфинга.

Кадровое обеспечение (педагог, работающий по данной программе, должен иметь высшее образование, обладать необходимыми знаниями программированию и иметь курсы повышения квалификации по данному направлению).

2.3 Оценочные материалы:

Теоретические знания проверяются посредством небольших презентаций, индивидуальных и групповых бесед. Материал считается усвоенным, если обучающийся грамотно знает теорию и выполняет практическую работу. Не усвоенным считается материал, если обучающийся не может выполнить практическую работу или не может ответить пройденный материал. В случае, если практическая работа выполнена с педагогической поддержкой или обучающийся не может полностью изложить теорию, материал считается усвоенным не до конца.

Итоговая аттестация проводится в форме практической работы по учебной программе. По итогам работы ставится итоговая оценка – «Зачет».

Итоговая работа по программе «Программирование на языке Python»:
В течение курса обучающиеся выполняют разные задания. На итоговом занятии обучающиеся презентуют свои проекты.

Критерии оценки итоговой работы:

Оценка теоретических знаний и практических умений и навыков осуществляется по трем уровням: высокий, средний, низкий.

Высокий уровень – учащиеся должны знать правила техники безопасности при работе, грамотно излагать программный материал, знать основные блоки команд, уметь выполнять тестирование программы и самостоятельно находить, и исправлять ошибки кода. Работать с разными функциями, добавлять и выполнять программы для решения алгоритмических задач в программе Python.

Средний уровень – учащиеся должны знать основные блоки команд, типы данных. Уметь работать с переменными и совершать операции над числовыми типами данных, грамотно и по существу излагать программный материал, не допуская существенных неточностей в ответе.

Низкий уровень – учащиеся не знают значительной части материала, допускают существенные ошибки, с большими затруднениями выполняют практические задания.

При обработке результатов учитываются критерии для выставления уровней:

Высокий уровень – выполнение 100% - 70% заданий;

Средний уровень – выполнение от 50% до 70% заданий;

Низкий уровень – выполнение менее 50% заданий.

Оценка «Зачет» ставится при выполнении итогового задания на 70%-100%.

Оценочные материалы:

Для учащихся – 15-17 лет

Фамилия, Имя - _____

1. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

2. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок.

Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания: Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

5. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Промежуточная аттестация:

1. Составьте выражение для вычисления в интерпретаторе Python и вставьте в поле ответа результат вычисления:

11111·11111111

— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц).

2. Запишите число $1.2345e3$ в виде десятичной дроби.

3. Составьте и запишите выражение для вычисления:

2014.0^{14} (Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

4. Приведите к целому типу число 2.99

5. Расставьте скобки в выражении `a and b or not a and not b` в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

6. Найдите результат выражения для заданных значений `a` и `b`. Учитывайте регистр символов при ответе. `a = True`; `b = False`; `a and b or not a and not b`

7. Отметьте выражения, значения которых равны `True`:

• `"239" < "30" and 239 < 30`

• `"239" < "30" and 239 > 30`

• `"239" > "30" and 239 < 30`

• `"239" > "30" and 239 > 30`

8. Укажите результат выражения: `"123" + "42"`

2.4 Календарный учебный график программы

№	Месяц, дата занятия	Тема занятия	Количество часов	Формы контроля	Место проведения
1	06.09.22	Вводное занятие. Инструктаж по ТБ при работе за ПК.	2	Опрос	МБОУ «СОШ № 1 р. п. Самойловка»

		Раздел 1. Знакомство с языком Python	3		
2	13.09.22	Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
3	13.09.22	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
4	20.09.22	Тест № 1. Знакомство с языком Python	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 2. Переменные и выражения	7		
5	20.09.22	Занятие 3. Переменные Практическая работа: Работа со справочной системой	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
6	27.09.22	Практическая работа: Переменные	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
7	27.09.22	Занятие 4. Выражения Практическая работа: Выражения	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
8	04.10.22	Занятие 5. Ввод и вывод	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
9	04.10.22	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»

		элементарные действия с числами			
10	11.10.22	Тест № 2. Выражения и операции.	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 3. Условные предложения	14		
11	18.10.22	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
12	18.10.22 25.10.22	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
13	25.10.22 08.11.22	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
14	08.11.22	Занятие 10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
15	15.11.22 22.11.22 29.11.22	Самостоятельная работа № 1 по теме «Условные операторы».	5	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
16	29.11.22 06.12.22	Занятие 11. Зачетная работа № 1. "Составление программ с ветвлением".	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
17	06.12.22	Тест № 3. "Условные операторы".	1	Тестирование, решение	МБОУ «СОШ № 1

				практическ их задач	р. п. Самойловка»
		Раздел 4. Циклы	17		
18	13.12. 22	Занятие 12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	Тестирова ние, решение практическ их задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
19	20.12. 22	Занятие 13. Оператор цикла for Практическая работа Решение задачи с циклом for.	2	Тестирова ние, решение практическ их задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
20	27.12. 22	Занятие 14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	2	Тестирова ние, решение практическ их задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
21	10.01. 23	Занятие 15. Случайные числа Практическая работа: Случайные числа	1	Тестирова ние, решение практическ их задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
22	10.01. 23 17.01. 23	Занятие 16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	3	Тестирова ние, решение практическ их задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
23	24.01. 23 31.01. 23	Самостоятельная работа № 2 "Составление программ с циклом"	4	Тестирова ние, решение практическ их задач,	МБОУ «СОШ № 1 р. п. Самойловка»

				творческая работа	
24	07.02.23	Тест № 4. Циклы	1	Тестирование, решение практических задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
25	07.02.23 14.02.23	Занятие 17. Творческая работа № 1. "Циклы"	2	Тестирование, решение практических задач, творческая работа	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 5. Функции	8		
26	14.02.23	Занятие 18. Создание функций Практическая работа Создание функций	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
27	21.02.23	Занятие 19. Локальные переменные Практическая работа Локальные переменные	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
28	21.02.23	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
29	28.02.23	Самостоятельная работа № 3 по теме "Функции"	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
30	07.03.23	Занятие 21. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»

31	07.03. 23	Тест № 5. Функции	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 6. Строки - последовательности символов	5		
32	14.03. 23	Занятие 22. Строки Практическая работа: Строки	1	Решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
33	14.03. 23	Занятие 23. Срезы строк	1	Решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
34	21.03. 23 04.04. 23	Занятие 24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	3	Решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 7. Сложные типы данных	9		
35	04.04. 23	Занятие 25. Списки Практическая работа: Списки	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
36	11.04. 23	Занятие 26. Срезы списков	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
37	11.04. 23 18.04. 23	Занятие 27. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
38	18.04. 23	Занятие 28. Матрицы	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»

39	25.04. 23	Тест № 7. Списки	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
40	25.04. 23	Занятие 29. Кортежи	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
41	02.05. 23	Занятие 30. Введение в словари	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
42	02.05. 23	Занятие 31. Множества в языке Python	1	Тестирование, решение практических задач	МБОУ «СОШ № 1 р. п. Самойловка»
		Раздел 8. Стиль программирования и отладка программ	8		
44	09.05. 23	Занятие 32. Стиль программирования	2	Решение практических задач, презентация проекта, рефлексия	МБОУ «СОШ № 1 р. п. Самойловка»
45	16.05. 23	Занятие 33. Отладка программ	2	Решение практических задач, презентация проекта, рефлексия	МБОУ «СОШ № 1 р. п. Самойловка»
46	23.05. 23	Занятие 34. Зачет по курсу «Программирование на языке Python»	2	Решение практических задач, презентация проекта, рефлексия	МБОУ «СОШ № 1 р. п. Самойловка»
47	30.05. 23	Занятие 35. Что дальше?	2	Решение практических задач,	МБОУ «СОШ № 1

				презентаци я проекта, рефлексия	р. п. Самойловка»
--	--	--	--	---------------------------------------	----------------------

2.5 Список литературы

Список литературы для педагога

1. Банкрашков А. Программирование для детей на языке Python. – Москва: АСТ, 2017.
2. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
3. Бриггс Дж. Python для детей. Самоучитель по программированию. – Москва: Манн, 2018.
4. Бэрри П. Изучаем программирование на Python. – Москва: Эксмо, 2017.
5. ВандерПлас Дж. Python для сложных задач. Наука о данных и машинное обучение. – Санкт-Петербург: Питер, 2018.
6. Васильев А. Python на примерах. Практический курс. Москва: Наука и техника, 2018.
7. Вордэрман К. Программирование на Python. Иллюстрированное руководство для детей. – Москва: Манн, Иванов и Фербер, 2018. 29
8. Гэддис Т. Начинаем программировать на Python. – Санкт-Петербург: БВХ-Петербург, 2019.
9. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
10. Доусон М. Програмируем на Python. – Санкт-Петербург: Прогресс книга, 2019.
11. Жуков Р. Язык программирования Python: практикум. Учебное пособие. – Москва: Инфра-М, 2020.
12. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
13. Златопольский Д.М. Основы программирования на языке Python. – Москва: ДМК Пресс, 2018.
14. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
15. Лекции А. В. Умнова, прочитанные в Школе анализа данных Яндекса – <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh5OpdwVl>
16. Лутц М. Изучаем Python. В 2-х т. – Москва: Вильямс, 2019.
17. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
18. Любанович Б. Простой Python. Современный стиль программирования. – Санкт-Петербург: Питер, 2019.
19. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.

20. МакГрат М. Программирование на Python для начинающих. Москва: Эксмо, 2015.
21. Маккини У. Python и анализ данных. – Москва: ДМК Пресс, 2018.
22. Мартелли А., Рейвенскрофт А., Холден С. Python. Справочник. Полное описание языка. – Москва: Диалектика, 2018.
23. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – Санкт-Петербург: Питер, 2020.
24. Мюллер Дж.П. Python для чайников. – Москва: Вильямс, 2019.
25. Окулов С. М. Основы программирования. М.: Бином. Лаборатория знаний, 2012.
26. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
27. Персиваль Г. Python. Разработка на основе тестирования. –Москва: ДМК Пресс, 2018.
28. Поляков К. Ю., Еремин Е. А. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.
29. Пратик Дж. Искусственный интеллект с примерами на Python. Создание приложений искусственного интеллекта с помощью Python для взаимодействия с окружающим миром. – Москва: Вильямс, 2019.
30. Пэйн Б. Python для детей и родителей. Играй и программируй. – Москва, Эксмо, 2018.
31. Рамальо Л. Python к вершинам мастерства. – Москва: ДМК Пресс, 2016.
32. Сайт «Python 3 для начинающих» – pythonworld.ru.
33. Сайт «Питонтьютор» – pythontutor.ru.
34. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
35. Свейгарт Э. Автоматизация рутинных задач с помощью Python. Практическое руководство для начинающих. – Москва: Вильямс, 2018.
36. Свейгарт Э. Учим Python, делая крутые игры. – Москва: Бомбора, 2018. 30
37. Содем Я.Э. Программирование компьютерного зрения на языке Python. – Москва: ДМК Пресс, 2016.
38. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
39. Томашевский П. Привет, Python. Моя первая книга по программированию. - Москва: Наука и техника, 2018.
40. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
41. Харрисон М. Как устроен Python. – Санкт-Петербург: Питер, 2019.
42. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. – М.: МПГУ, 2015.

Список литературы для обучающихся

1. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВПетербург, 2019. – 768 с.
2. Доусон М. Програмируем на Python. – СПб.: Питер, 2014. – 416 с.
3. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестселлеры O'Reilly»).
4. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
5. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
6. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
7. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный
8. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.

Критерии оценки результативности образовательной деятельности на аттестационном занятии

Низкий уровень:

Учащиеся должны знать / понимать: – технику безопасного поведения во время занятий; – правила поведения в общественных местах, – понятие программы Python; общую структуру программы; – основные типы данных; – оператор присваивания; – назначение условного оператора; – способ записи условного оператора; – логический тип данных; – логические операторы or, and, not; – основные циклы с условием; – основные правила записи циклов с условием; – формат записи цикла с параметром; – понятие функции; – основные принципы структурного программирования; – понятие локальных переменных подпрограмм; – способ передачи параметров. – назначение строкового типа данных; – операторы для работы со строками; – операции со строками; – способ описания списка; – способ описания кортежа; – способ описания словаря; – основные операции, выполняемые со списками, кортежами и словарями; – понятие множества; – способы описания множества; – операторы работы с множествами. – что такое стиль программирования; – правила именования объектов; – основные рекомендации при написании программ.

Учащиеся должны уметь: – понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога; – работать в паре, малой группе; – выполнить установку программы под руководством педагога; – выполнить простейшую программу в интерактивной среде; – написать комментарии в программе под руководством педагога; – решать задачи на элементарные действия с числами; – использовать условный оператор; – определять вид цикла, наиболее удобный для решения поставленной задачи; – использовать цикл с условием под руководством педагога; – определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога; – создавать и использовать основные функции; – описывать и соединять строки; – находить подстроку в строке с помощью педагога; – находить количество слов в строке; – вводить и выводить элементы списка под руководством педагога; – приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога; – описывать множества под руководством педагога; – определять принадлежность элемента множеству по наводящим вопросам; – определять вид ошибок и находить ошибки в программе под руководством педагога; – составлять элементарные алгоритмы для решения задач; – реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень:

Учащиеся должны знать / понимать: – основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности; – основные приемы взаимодействия в группе сверстников; – понятие программы; – структуру программы на Python; – режимы работы с Python. – типы данных; – целые, вещественные типы данных и операции над ними; – оператор присваивания; – назначение условного оператора; – способ записи условного оператора; – логический тип данных; – логические операторы or, and, not; – циклы с условием и их виды; – назначение и особенности использования цикла с параметром; – формат записи цикла с параметром; – примеры использования циклов различных типов. – понятие функции; – основные способы описания функции; – принципы структурного программирования; – понятие локальных переменных подпрограмм; – понятие формальных и фактических параметров подпрограмм; – способ передачи параметров. – назначение строкового типа данных; – операторы для работы со строками; – процедуры и функции для работы со строками; – операции со строками; – сложные типы данных; – способ описания списка; – способ доступа к элементам списка; – способ описания кортежа; – способ описания словаря; – операции, выполняемые со списками, кортежами и словарями; – понятие множества; – способы описания множества; – операторы работы с множествами. – что такое стиль программирования; – правила именования объектов; – основные рекомендации при написании программ; – основные шаги работы над проектом, его презентации.

Учащиеся должны уметь: – уважительно относиться к преподавателям и сверстникам; – применять некоторые приемы логического (абстрактное) мышления; – концентрировать внимание на одном или двух объектах; – понимать причины успеха/неуспеха с помощью анализа педагога; – выполнить установку программы; – выполнить простейшую программу в интерактивной среде; – написать комментарии в программе; – решать задачи на элементарные действия с числами; – использовать условный оператор; – создавать сложные условия с помощью логических операторов; – определять вид цикла, наиболее удобный для решения поставленной задачи; – использовать цикл с условием; – определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; – создавать и использовать функции; – использовать механизм параметров для передачи значений; – описывать строки; – соединять строки; – находить длину строки; – вырезать часть строки; – находить подстроку в строке; – находить количество слов в строке; – описывать списки; – вводить элементы списка; – выводить элементы списка; – выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; – использовать вложенные списки; – приводить примеры использования вложенных списков (матриц); – описывать множества; – определять принадлежность элемента множеству; – вводить элементы множества; – выводить элементы множества. – определять вид ошибок и находить ошибки в программе. – составлять алгоритмы для решения задач; – реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; – отлаживать и тестировать

программы, написанные на языке Python; – понимать и сохранять цели и задачи учебной деятельности в ходе занятия; – планировать свою деятельность с помощью взрослого; – сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи; – понимать причины успеха/неуспеха учебной деятельности под руководством взрослого; – делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога; – выражать творческие идеи, разработать творческий проект на основе образца; – конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень:

Учащиеся должны знать / понимать: – основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий; – приемы конструктивного взаимодействия в группе сверстников; – понятие программы; – структуру программы на Python; – режимы работы с Python. – общую структуру программы; – типы данных; – целые, вещественные типы данных и операции над ними; – оператор присваивания; – назначение условного оператора; – способ записи условного оператора; – логический тип данных; – логические операторы or, and, not; – циклы с условием и их виды; – правила записи циклов условием; – назначение и особенности использования цикла с параметром; – формат записи цикла с параметром; – примеры использования циклов различных типов. – понятие функции; – способы описания функции; – принципы структурного программирования; – понятие локальных переменных подпрограмм; – понятие формальных и фактических параметров подпрограмм; – способ передачи параметров. – назначение строкового типа данных; – операторы для работы со строками; – процедуры и функции для работы со строками; – операции со строками; – сложные типы данных; – способ описания списка; – способ доступа к элементам списка; – способ описания кортежа; – способ описания словаря; – операции, выполняемые со списками, кортежами и словарями; – понятие множества; – способы описания множества; – операторы работы с множествами. – что такое стиль программирования; – правила именования объектов; – основные рекомендации при написании программ; – правила и этапы работы над проектом; – приемы успешной презентации проекта.

Учащиеся должны уметь: – выполнить установку программы; – выполнить простейшую программу в интерактивной среде; – написать комментарии в программе; – решать задачи на элементарные действия с числами; – использовать условный оператор; – создавать сложные условия с помощью логических операторов; – определять вид цикла, наиболее удобный для решения поставленной задачи; – использовать цикл с условием; – определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; – создавать и использовать функции; – использовать механизм параметров для передачи значений; – описывать строки; – соединять строки; – находить длину строки; – вырезать

часть строки; – находить подстроку в строке; – находить количество слов в строке; – описывать списки; – вводить элементы списка; – выводить элементы списка; – выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; – использовать вложенные списки; – приводить примеры использования вложенных списков (матриц); – описывать множества; – определять принадлежность элемента множеству; – вводить элементы множества; – выводить элементы множества. – определять вид ошибок и находить ошибки в программе. – составлять алгоритмы для решения задач; – реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; – отлаживать и тестировать программы, написанные на языке Python; – понимать учебную задачу, анализировать достижение результата; – делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений; – понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога; – планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи; – создать проектировочную команду и организовать ее деятельность; – разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы; – самостоятельно подготовить презентацию и защитить проект.

ПРОТОКОЛ

РЕЗУЛЬТАТОВ _____ АТТЕСТАЦИИ
вид аттестации

обучающихся МБОУ «СОШ № 1 р. п. Самойловка»

от «___» _____ 20___ г.

Ф.И.О. педагога

_____ год обучения _____ количество учащихся

Форма проведения

Члены аттестационной комиссии (Ф.И.О., должность):

№	Фамилия, имя ребенка	Высокий уровень	Средний уровень	Низкий уровень
1.				
2.				
3.				
4.				
5.				
6.				

Всего аттестовано _____ учащихся. Из них по результатам аттестации:

высокий уровень _____ чел.

средний уровень _____ чел.

низкий уровень _____ чел.

Подпись педагога

Члены аттестационной
 комиссии

Приложение 3

Итоговый контроль Диагностическая карта

№	Ф. И. О.	Входящая диагностика	Промежуточная диагностика	Итоговая диагностика
1				
2				
3				
4				
5				
	ИТОГО в %	Н – С – В -	Н – С – В -	Н – С – В -

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов.

Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).