

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования» имеет техническую направленность.

Актуальность программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирование невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

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. Как считают многие, один из самых используемых языков программирования в мире.

Актуальность программы «Основы программирования» вызвана потребностью современного информационного общества в высокообразованных, адаптированных к изменениям специалистах в IT-сфере. Для удовлетворения данной потребности перед дополнительным образованием стоит задача развития человеческого потенциала через выявление талантливых детей, развитие их мотивации и способностей.

Изучение языка программирования Python поможет ребенку получить более целостное представление о профессии программиста, разработчика, инженера.

Педагогическая целесообразность

Данная дополнительная общеразвивающая программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Цель программы.

Основной целью данного учебного курса является ознакомление

слушателя с объектно-ориентированным языком программирования Python, с возможностями, синтаксисом языка, технологией и методами программирования в среде Python, обучение практическим навыкам программирования на языке Python для решения типовых задач математики и информатики, а также при разработке простейших игр.

Задачи программы

Реализация поставленной цели предусматривает решение следующих задач:

- развитие интереса учащихся к изучению программирования;
- знакомство учащихся с основами программирования в среде Python;
- овладение базовыми понятиями теории алгоритмов при решении математических задач;
- формирование навыков работы в системе программирования Python;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- формирование навыков алгоритмического и логического мышления;
- формирование навыков грамотной разработки программ;
- формирование практических навыков решения прикладных задач;
- формирование практических навыков разработки игр.

Отличительные особенности данной программы

На занятиях обучающиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python.

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python. На втором году обучения закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

Организационные условия реализации программы

Возраст обучающихся: 14-17 лет.

Наполняемость группы –15 человек.

Общее количество часов: 144.

Режим занятий: 1 раз в неделю по 3 учебных часа.

Продолжительность учебного часа - 40 минут.

Форма обучения: очная.

Уровень освоения программы: продвинутый.

Язык обучения: русский.

Место проведения занятий – МАУ ДО ДЮОЦ.

Технологии обучения: игровые технологии, системно-деятельностный подход, лично-ориентированные технологии,

здоровьесберегающие технологии, информационно-коммуникационные технологии, дистанционные образовательные технологии.

Условия набора в объединение:

Набор свободный, при наличии заявления от родителей.

Программа включает 1 образовательный модуль.

Планируемые (ожидаемые) результаты

В результате изучения дисциплины получают дальнейшее развитие личностные регулятивные, коммуникативные и познавательно-универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ–компетентность обучающихся, составляющая психолого-педагогическую, инструментальную основу формирования способности и готовности к освоению систематических знаний, к их самостоятельному пополнению, переносу и интеграции, способности к сотрудничеству и коммуникации, решению личностно и социально значимых проблем и воплощению решений в практику, способности к самоорганизации, саморегуляции и рефлексии.

В ходе изучения данного курса в основном формируются и получают развитие следующие **метапредметные результаты:**

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вместе с тем, вносятся существенный вклад в развитие **личностных результатов:**

- формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной,

общественно-полезной, учебно- исследовательской, творческой и других видах деятельности.

В части развития **предметных результатов** наибольшее влияние изучение курса оказывает на:

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойства;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической;
- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
- формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

В результате изучения дисциплины обучающиеся должны знать:

- основные типы алгоритмов;
- иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;
- содержание этапов разработки программы: алгоритмизация-кодирование-

отладка- тестирование;

- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;
- основы разработки простых игр в системе программирования Python.

Обучающиеся должны уметь:

- записывать алгоритмические структуры на языке программирования Python;
- 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;
- решать простые, сложные и нестандартные задачи;
- создавать простые игры;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

Формы подведения итогов реализации программы

Основной формой подведения итогов дополнительной общеразвивающей программы «Основы программирования Python» является решения задач (разработка и отладка программ на языке программирования Python).

Учебно-тематический план

№ п/п	Тема занятий	Количество часов		
		Всего	Теория	Практика
Инструктаж по технике безопасности в компьютерном классе		1	1	
1.	Раздел 1. Знакомство с языком Python	6	2	4
1.1	Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	2	1	1
1.2.	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	4	1	3
2.	Раздел 2. Переменные и выражения	16	4	12
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	4	1	3
2.2	Занятие 4. Выражения Практическая работа: Выражения	4	1	3
2.3	Занятие 5. Ввод и вывод	4	1	3
2.4	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	4	1	3
3	Раздел 3. Условные предложения	10	3	7
3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	3	1	2
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	3	1	2
3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление Практическая работа: «Условные операторы»	4	1	3
4	Раздел 4. Циклы	15	4	11
4.1.	Занятие 10. Оператор цикла с условием Практическая работа «Числа Фибоначчи»	3	1	3
4.2.	Занятие 11. Оператор цикла for Практическая работа Решение задачи с циклом for.	4	1	3
4.3.	Занятие 12. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	4	1	3
4.4.	Занятие 13. Случайные числа Практическая работа: Случайные числа	4	1	2
5	Раздел 5. Функции	14	4	10
5.1.	Занятие 14. Создание функций Практическая работа «Создание функций»	4	1	3
5.2.	Занятие 15. Локальные переменные Практическая работа «Локальные переменные»	3	1	2
5.3.	Занятие 16. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	13	1	12
5.4	Занятие 17. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	4	1	3
6	Раздел 6. Строки - последовательности символов	23	3	20

6.1.	Занятие 18. Строки Практическая работа: Строки	4	1	3
6.2.	Занятие 19. Срезы строк	4	1	3
6.3	Занятие 20. Примеры решения задач со строками Практическая работа: Решение задач со строками.	15	1	14
7	Раздел 7. Сложные типы данных	33	7	26
7.1.	Занятие 21. Списки Практическая работа: Списки	3	1	2
7.2	Занятие 22. Срезы списков	3	1	2
7.3	Занятие 23. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	9	1	8
7.4	Занятие 24. Матрицы	3	1	2
7.5	Занятие 25. Кортежи	3	1	2
7.6	Занятие 26. Введение в словари	4	1	3
7.7	Занятие 27. Множества в языке Python	14	1	13
8	Раздел 8. Стиль программирования и отладка программ	26	2	24
8.1	Занятие 28. Стиль программирования	4	1	3
8.2	Занятие 29. Отладка программ	4	1	3
8.3	Занятие 30. Зачет по курсу «Основы программирования»	2		2
ВСЕГО		144	30	114

Содержание программы

Вводное занятие

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

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

Обучающиеся должны знать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Обучающиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения

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

Выражения. Операции. Порядок выполнения операций. Математические функции. 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.

Обучающиеся должны знать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения

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».

Обучающиеся должны знать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;

Обучающиеся должны уметь:

- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы

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»

Обучающиеся должны знать:

- 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.

Обучающиеся 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

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.

Обучающиеся должны знать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Обучающиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов

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.

Обучающиеся должны знать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Обучающиеся 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

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

Обучающиеся должны знать:

- 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.

Обучающиеся 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. Стиль программирования и отладка программ

Стиль программирования. Отладка программ.

Зачет по курсу «Основы программирования»

Обучающиеся должны знать:

- стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Обучающиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

1.4. Планируемые результаты освоения дополнительной общеразвивающей программы «Основы программирования»

**Реализация программы предполагает достижение обучающимися
следующих личностных, метапредметных и предметных результатов:**

Предметные:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с

основными алгоритмическими структурами — линейной, условной и циклической;

- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.
- Личностные:
- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;

- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Материально-техническое обеспечение:

Помещение: МАУ До ДЮОЦ. Кабинет, соответствующий санитарно-гигиеническим требованиям.

Для реализации образовательной программы используются:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары.

Информационно-методическое обеспечение:

- персональный компьютер (12 шт.);
- 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 мене 19”;

Программное обеспечение:

1. Операционная система Windows 7 Профессиональная или выше;
2. Интерпретатор Python версии 3.7 и выше;
3. IDE JAtBrains PyCharm;
4. Foxit Reader или другой просмотрщик PDF файлов;
5. WinRAR;
6. Пакет офисных программ;
7. Adobe Photoshop или другой растровый графический редактор;
8. Любой браузер для интернет серфинга.

Дидактические материалы:

- таблицы;
- схемы;
- памятки;
- видеозаписи, мультимедийные материалы;
- упражнения.

Формы итоговой аттестации – защита проекта.

Оценочные материалы

Диагностика результативности образовательного процесса.

Система оценки и фиксирования результатов

В начале учебного года при комплектовании групп осуществляется входной контроль (в форме собеседования и стартового тестирования) для определения уровня развития детей и их творческих способностей и распределения по группам базового и продвинутого уровня.

Аттестацию учащихся в процессе реализации программ рекомендуется проводить с использованием диагностических методов.

Цель проведения диагностики – определение изменения уровня развития учащихся, их творческих способностей, получение сведений для совершенствования образовательной программы и методов обучения.

Промежуточная диагностика или текущий контроль позволяет выявить и проанализировать уровень усвоения материала реализуемого модуля и внести необходимые коррективы, в том числе и индивидуально.

Итоговая диагностика является необходимым завершающим элементом программы и проводится при завершении реализации программы. В качестве итоговой формы аттестации рекомендуется проведение научно-практической конференции, на которой обучающиеся представляют результаты проектной деятельности.

Одним из показателей результативности реализации программ технической направленности является участие в выставках, конкурсах, конференциях муниципального, регионального, федерального уровней. Формой фиксации результатов в данном случае являются свидетельства (сертификаты участия), грамоты и дипломы, портфолио, статьи в информационных источниках).

Сравнительный анализ различных форм контроля позволяет оценить в целом достижение планируемых результатов и уровни освоения программы учащимися.

Уровни	Параметры	Показатели
Высокий уровень (80-100%)	Теоретические знания.	Обучающийся глубоко и всесторонне усвоил навыки работы; уверенно, логично, последовательно и грамотно излагает материал; умело обосновывает и аргументирует выдвигаемые им идеи; делает выводы и обобщения; свободно владеет понятиями.
	Практические умения и навыки.	Способен применять практические умения и навыки во время выполнения самостоятельных заданий. Работу выполняет с соблюдением правил техники безопасности, аккуратно, доводит ее до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
Средний уровень (50-79%)	Теоретические знания.	Тема раскрыта недостаточно четко и полно, то есть обучающийся освоил проблему, по существу излагает ее, но допускает несущественные ошибки и неточности; слабо аргументирует научные положения; затрудняется в формулировании выводов и обобщений; частично владеет системой понятий.
	Практические умения и навыки.	Владеет базовыми навыками и умениями, но не всегда может выполнить самостоятельное задание, затрудняется и просит помощи педагога. В работе допускает небрежность, делает ошибки, но может устранить их после наводящих вопросов или самостоятельно. Оценить результаты своей деятельности может с подсказкой педагога.
Низкий уровень (меньше 50%)	Теоретические знания.	Обучающийся не усвоил значительной части проблемы, допускает существенные ошибки и неточности при рассмотрении ее; не может аргументировать научные положения; не формулирует выводов и обобщений; не владеет понятийным аппаратом.
	Практические умения и навыки.	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. В работе допускает грубые ошибки, не может их найти даже после указания. Не способен самостоятельно оценить результаты своей работы.

Электронные образовательные ресурсы

1. Центр онлайн-обучения «Фоксфорд» <http://foxford.ru/>;
2. Национальный Открытый Университет «ИНТУИТ» <http://www.intuit.ru/>;
3. Образовательная платформа и конструктор онлайн-курсов Stepik.
<https://stepik.org/course/67/syllabus>;
4. Всероссийский портал «Дистанционная подготовка по информатике»
(informatics.mcsme.ru);
5. Официальная документация языка Python (docs.python.org).

Список литературы

Литература для обучающихся:

Сайт/справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.

Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.

Сайт/Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.

Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.

Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Литература для педагога

Бизли, Дэвид М. Python. Подробный справочник. – М. – СПб.: Символ-Плюс, 2022.

Лутц, Марк Python. Справочник. – М.: Вильямс, 2021.

Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.

Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.