

Принята на заседании
педагогического совета
Протокол № 7
от 25.05.2024 г.

УТВЕРЖДАЮ

Директор

 Тимошина Е.А.

Приказ № 174-1
от 30.05.2024 г.

М.П.



**Дополнительная общеобразовательная
общеразвивающая программа
технической направленности
«Основы программирования на языке Python»**

Уровень программы: базовый

Вид: модифицированная

Возрастная категория: от 13 до 15 лет

Состав группы: 10 человек

Срок реализации: 1 год

ИД-номер программы в Навигаторе: 31379

Автор-составитель:

Кобелева Е.А.

учитель информатики,

педагог дополнительного

образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная (общеразвивающая) программа «Основы программирования на языке Python» имеет техническую направленность.

Программа составлена в соответствии с Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, на основании материалов специальной методической и научно-популярной литературы, учитывает возрастные и психологические особенности и интересы детей, запросы социума, материальные возможности семей обучающихся, конкретные условия учреждения, личный опыт педагога.

При разработке программы соблюдены Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы).

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python» приведена в соответствие с нормативно-правовой базой:

- Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (ст. 2, ст. 15, ст.16, ст.17, ст.75, ст. 79);
- Концепция развития дополнительного образования до 2030 года, утвержденная Распоряжением Правительства РФ от 31 марта 2022 года № 678-р;
- Приказ Министерства просвещения Российской Федерации от 27 июля 2022 года №629 «Об утверждении порядка организации образовательной деятельности по дополнительным общеобразовательным программам»
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (письмо Министерства образования и науки Российской Федерации от 18.11.2015г. № 09-3242);
- Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденные постановлением Главного государственного санитарного врача РФ от 28 сентября 2020 года №28.

В современную жизнь человека все шире внедряются компьютеры и информационные технологии. Поэтому все большее значение приобретает компьютерная грамотность.

Курс изучения компьютерной грамотности состоит из двух разделов: пользовательского курса и программирования. Раздел «Программирование» в школьном курсе представлен языком программирования Pascal, а многим учащимся хочется познакомиться с другими языками программирования, самим попробовать разработать программы, которые можно использовать на уроках и во внеурочной деятельности. Данная Программа позволяет реализовать эти желания, так как уделяется большое внимание практической работе учащихся на компьютере, самостоятельной разработке ими программ для решения практических задач.

Данная Программа разработана на основании программы «Программирование на языке Python» (разработчик Киселева Н.Н., педагог дополнительного образования ГБОУ Школа № 1375).

Новизна программы

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Актуальность программы

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Педагогическая целесообразность Программы

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Цель Программы: обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентно способной личности.

Задачи Программы

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

Развивающие:

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

Воспитательные:

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

Отличительные особенности данной программы

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Возрастная категория обучающихся по программе от 13 до 15 лет.

Срок реализации программы составляет 1 год. Общее количество часов в год составляет 108 часов.

Формы и режим занятий - групповые – для всей группы при изучении общих практических и теоретических вопросов. Наполняемость группы до 10 человек.

В ходе реализации программы применяется дифференцированный, индивидуальный подход к каждому обучающемуся.

Занятия проводятся 2 раза в неделю по 1 часу (1 раз по 40 минут) и по 2 часа (2 раза по 40 минут с перерывом 10 минут).

ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ И СПОСОБЫ ОПРЕДЕЛЕНИЯ ИХ РЕЗУЛЬТАТИВНОСТИ

По итогам реализации программы, учащиеся *будут знать*:

- принципы программирования на языке Python;
- основы дизайнерского оформления созданных программ.

По итогам реализации программы, учащиеся *будут уметь*:

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой, благодаря иллюстрированной среде программирования, мотивации к обучению и познанию;
- сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов.
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Предметные:

- владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- владение стандартными приёмами написания программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ на языке Python;
- владение навыками и опытом разработки программ на Python, включая тестирование и отладку программ;
- владение элементарными навыками формализации прикладной задачи

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, понимая, что в программировании длинная программа – не значит лучшая программа;
- 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, принятия решений;
- способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания.

СОДЕРЖАНИЕ ПРОГРАММЫ.

1. История языков программирования. Компиляция и интерпретация.
2. Знакомство с Python и средами программирования.

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

3. Типы данных в программировании. Определение переменной.

Язык Python. Среда программирования.

Теория. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы.

Практика. Установка языка программирования Python 3.5 и среды программирования WingIDE 100.

4. Ввод данных с клавиатуры.

Язык Python. Среда программирования. Особенности ввода – вывода

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

5. Логические выражения.

Основные понятия: логический тип данных, логические операции (and, or, not), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

6. Условный оператор. Инструкция if.

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

7. Множественное ветвление.

8. Цикл While.

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

9. Цикл For.

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

10. Строки как последовательности символов.

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

11. Списки — изменяемые последовательности. Массивы.

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание. Основные задачи обработки массивов: поиск, сортировка, реверс.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

12. Введение в словари.

Основные понятия: словарь, множество, ключ, кодирование.

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

13. Функции в программировании. Параметры и аргументы функций.

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

14. Локальные и глобальные переменные. Процедуры.

15. Файлы. Чтение текстового файла. Запись в файл.

16. Алгоритм Евклида (нахождение наибольшего общего делителя)

17. Вычисление факториала на языке программирования Python

18. Двоичный (бинарный) поиск элемента в массиве

19. Замена элементов в списке

20. Перевод чисел из десятичной системы счисления в двоичную

21. Решето Эратосфена - алгоритм определения простых чисел

22. Сортировка выбором (поиск минимума и перестановка)

23. Сортировка методом пузырька

24. Сумма и произведение цифр числа
25. Тестирование простоты числа методом перебора делителей
26. Числа Фибоначчи (вычисление с помощью цикла while и рекурсии)

| № п/п | Наименование тем, разделов | Теория | Практика |
|----------|--|-----------|-----------|
| 1 | Техника безопасности при работе в компьютерном классе. | 1 | 0 |
| 2 | Введение в программирование. | 1 | 0 |
| 3 | Знакомство с языком Python. | 5 | 13 |
| | Понятия кода, интерпретатора, программы | 1 | 3 |
| | Интегрированные среды, исполнение кода и отладка | 1 | 4 |
| | Переменные, основные операторы | 2 | 4 |
| | Базовые типы данных, ветвления | 1 | 1 |
| | Контрольное тестирование | 0 | 1 |
| 4 | Базовые конструкции Python. | 10 | 20 |
| | Циклы, срезы, списочные выражения | 5 | 10 |
| | Методы списков и строк. Функции | 5 | 8 |
| | Контрольное тестирование | 0 | 2 |
| 5 | Прикладные задачи Python. | 16 | 24 |
| | Функциональный стиль программирования, лямбда-функции, модуль functools. | 1 | 3 |
| | Хэш-таблицы. Модуль Collections | 2 | 2 |
| | Обработка исключений | 1 | 2 |
| | Принципы устройства и механика создания модулей и пакетов | 2 | 3 |
| | ООП: инкапсуляция, наследование, полиморфизм | 2 | 3 |
| | Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования | 2 | 2 |
| | Итераторы и генераторы | 2 | 2 |
| | Автоматизированное тестирование в Python | 2 | 2 |
| | Создание графического интерфейса | 1 | 2 |
| | Работа со звуками | 1 | 2 |
| | Контрольное тестирование | 0 | 1 |
| 6 | Проектная деятельность | 4 | 12 |
| 7 | Итоговое занятие | 1 | 1 |
| | ВСЕГО: | 38 | 70 |

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

| № п/п | Тема | Виды учебной деятельности | Количество часов |
|-------|--|--|------------------|
| 1. | История языков программирования. Язык Python. Простейшие программы. Реализация вычислений и ветвлений. | знать место языка Python среди языков программирования высокого уровня, знать особенности структуры программы, представленной на языке Python, <ul style="list-style-type: none"> ◦ иметь представление о модулях, входящих в состав среды Python, знать возможности и ограничения использования готовых модулей, иметь представление о величине, ее характеристиках, знать что такое операция, опе- | 20 |

| № п/п | Тема | Виды учебной деятельности | Количество часов |
|--------|---|--|------------------|
| | | ранд и их характеристики, знать принципиальные отличия величин структурированных и неструктурированных, иметь представление о логических выражениях и входящих в них операндах, операциях и функциях, уметь записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить, знать основные операторы языка Python, их синтаксис, иметь представление о процессе исполнения каждого из операторов, знать принципиальные отличия между формальными, локальными и глобальными переменными | |
| 2. | Реализация циклических, вспомогательных алгоритмов. Рекурсия. | знать основные операторы языка Python, их синтаксис, иметь представление о процессе исполнения каждого из операторов, уметь разрабатывать программы обработки числовой и символьной информации, уметь разрабатывать программы (линейные, разветвляющиеся и с циклами), иметь представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня, знать правила описания процедур и функций в Python и построение вызова процедуры, знать область действия описаний в процедурах, иметь представление о рекурсии, знать ее реализацию на Python, владеть основными приемами формирования процедуры и функции | 30 |
| 3. | Множества. Словари | знать основные операторы языка Python, их синтаксис, иметь представление о процессе исполнения каждого из операторов, уметь разрабатывать программы обработки числовой и символьной информации, уметь разрабатывать программы (линейные, разветвляющиеся и с циклами) | 20 |
| 4. | Символьные строки. Обработка символьных строк. | знать основные операторы языка Python, их синтаксис, иметь представление о процессе исполнения каждого из операторов, уметь разрабатывать программы обработки числовой и символьной информации, уметь разрабатывать программы (линейные, разветвляющиеся и с циклами) | 16 |
| 5. | Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов. | знать основные операторы языка Python, их синтаксис, иметь представление о процессе исполнения каждого из операторов, уметь разрабатывать программы обработки числовой и символьной информации, уметь разрабатывать программы (линейные, разветвляющиеся и с циклами), определять в программе тип «массив», «матрица», знать свойства данных типа «массив», «матрица», уметь воспроизводить алгоритмы сортировки массивов и матриц, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах, уметь читать и записывать текстовые файлы в заданном формате | 20 |
| 6. | Итоговое занятие | решать задачи в среде Python. | 2 |
| Итого: | | | 108 |

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

| № п/п | Название раздела, темы | Всего часов | Дата проведения | |
|-----------|--|-------------|-----------------|-------|
| | | | План. | Факт. |
| I | История языков программирования. Язык Python. Простейшие программы. Реализация вычислений и ветвлений | 20 | | |
| 1. | История языков программирования. Инструктаж по ТБ. | 1 | 02.09 | |
| 2. | Язык Python | 1 | 05.09 | |
| 3. | Оператор ввода – input | 1 | 05.09 | |
| 4. | Оператор вывода – print | 1 | 09.09 | |
| 5. | Оператор ввода – input . Оператор вывода – print | 1 | 12.09 | |
| 6. | Типы данных | 1 | 12.09 | |
| 7. | Тип данных int (целочисленный) | 1 | 16.09 | |
| 8. | Операции над типом int (целочисленное деление, остаток от деления) | 1 | 19.09 | |
| 9. | Решение задач | 1 | 19.09 | |
| 10. | Тип данных float (вещественный) | 1 | 23.09 | |
| 11. | Операции над типом float | 1 | 26.09 | |
| 12. | Решение задач | 1 | 26.09 | |
| 13. | Ветвление. Полное ветвление | 1 | 30.09 | |
| 14. | Ветвление. Неполное ветвление | 1 | 03.10 | |
| 15. | Знакомство с конструкцией if...elif...else | 1 | 03.10 | |
| 16. | Решение задач | 1 | 07.10 | |
| 17. | Логические операнды | 1 | 10.10 | |
| 18. | Решение задач с помощью and, or | 1 | 10.10 | |
| 19. | Вложенное ветвление | 1 | 14.10 | |
| 20. | Решение задач | 1 | 17.10 | |
| II | Реализация циклических, вспомогательных алгоритмов. Рекурсия | 30 | | |
| 21. | Знакомство с понятием «цикл» | 1 | 17.10 | |
| 22. | Цикл for, итерация цикла | 1 | 21.10 | |
| 23. | Решение задач с помощью for | 1 | 24.10 | |
| 24. | Решение задач с помощью for | 1 | 24.10 | |
| 25. | Выход из цикла for с помощью break | 1 | 28.10 | |
| 26. | Конструкция for...else | 1 | 31.10 | |
| 27. | Конструкция for...else | 1 | 31.10 | |
| 28. | Решение задач | 1 | 04.11 | |
| 29. | Цикл с условием | 1 | 07.11 | |
| 30. | Решение задач на цикл с условием | 1 | 07.11 | |
| 31. | Решение задач на цикл с условием | 1 | 11.11 | |
| 32. | Цикл while | 1 | 14.11 | |
| 33. | Решение задач с помощью while | 1 | 14.11 | |
| 34. | Решение задач с помощью while | 1 | 18.11 | |
| 35. | Выход из цикла while с помощью break | 1 | 21.11 | |
| 36. | Понятие вложенного цикла | 1 | 21.11 | |
| 37. | Решение задач на все типы циклов | 1 | 25.11 | |
| 38. | Решение задач на все типы циклов | 1 | 28.11 | |
| 39. | Понятие функции – def | 1 | 28.11 | |
| 40. | Понятие функции – def | 1 | 02.12 | |
| 41. | Описание переменных в функции, локальные и глобальные переменные | 1 | 05.12 | |
| 42. | Решение задач на вызов функции | 1 | 05.12 | |
| 43. | Решение задач на вызов функции | 1 | 09.12 | |

| № п/п | Название раздела, темы | Всего часов | Дата проведения | |
|------------|---|-------------|-----------------|-------|
| | | | План. | Факт. |
| 44. | Понятие рекурсии в Python | 1 | 12.12 | |
| 45. | Понятие рекурсии в Python | 1 | 12.12 | |
| 46. | Решение типовых задач на рекурсию | 1 | 16.12 | |
| 47. | Решение задач на замену цикла for рекурсией | 1 | 19.12 | |
| 48. | Решение задач на замену цикла for рекурсией | 1 | 19.12 | |
| 49. | Решение задач на замену рекурсии циклом for | 1 | 23.12 | |
| 50. | Решение задач на замену рекурсии циклом for | 1 | 27.12 | |
| III | Множества. Словари | 20 | | |
| 51. | Понятие «Множества» в Python | 1 | 27.12 | |
| 52. | Ввод множества с клавиатуры | 1 | 30.12 | |
| 53. | Запись в множества с помощью if | 1 | | |
| 54. | Запись в множества с помощью if | 1 | | |
| 55. | Проход по значениям множества с помощью for | 1 | | |
| 56. | Решение задач по теме «Множества» | 1 | | |
| 57. | Понятие «Словари» в Python | 1 | | |
| 58. | Ввод с клавиатуры в словари | 1 | | |
| 59. | Ввод с клавиатуры в словари | 1 | | |
| 60. | Понятие значения в словарях. Инструктаж по ТБ. | 1 | | |
| 61. | Понятие значения ключа в словарях | 1 | | |
| 62. | Решение задач | 1 | | |
| 63. | Запись в множества через условия | 1 | | |
| 64. | Добавления элементов в множество | 1 | | |
| 65. | Знакомство с методами множества | 1 | | |
| 66. | Решение задач | 1 | | |
| 67. | Решение задач | 1 | | |
| 68. | Проход по ключам-значениям в множестве | 1 | | |
| 69. | Решение задач по множествам | 1 | | |
| 70. | Решение задач по множествам | 1 | | |
| IV | Символьные строки. Обработка символьных строк | 16 | | |
| 71. | Ввод с клавиатуры | 1 | | |
| 72. | Операции над строками | 1 | | |
| 73. | Срезы в строках | 1 | | |
| 74. | Проход по элементам в строках с помощью for | 1 | | |
| 75. | Создание пустой строки | 1 | | |
| 76. | Добавление элементов в новую строку через условие | 1 | | |
| 77. | Вывод элементов строки в одну строку | 1 | | |
| 78. | Методы строк. Проверка на числа | 1 | | |
| 79. | Методы строк. Проверка на заглавные символы | 1 | | |
| 80. | Методы строк. Поиск по значению | 1 | | |
| 81. | Методы строк. Метод count | 1 | | |
| 82. | Методы строк. Метод reversed | 1 | | |
| 83. | Методы строк. Перевод из строки в число | 1 | | |
| 84. | Методы строк. Перевод из типа int в тип str | 1 | | |
| 85. | Решение задач | 1 | | |
| 86. | Решение задач | | | |
| V | Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов | 20 | | |
| 87. | Понятие массива в программировании | 1 | | |
| 88. | Понятие list (список) в Python | 1 | | |
| 89. | Ввод списка с клавиатуры | 1 | | |

| № п/п | Название раздела, темы | Всего часов | Дата проведения | |
|----------|--|-------------|-----------------|-------|
| | | | План. | Факт. |
| 90. | Вывод списка | 1 | | |
| 91. | Понятие индекса и значение массива (списка) | 1 | | |
| 92. | Методы списка. Add | 1 | | |
| 93. | Методы списка. Pop | 1 | | |
| 94. | Методы списка. Copy | 1 | | |
| 95. | Методы списка. Clear | 1 | | |
| 96. | Методы списка. Reverse | 1 | | |
| 97. | Методы списка. Index | 1 | | |
| 98. | Понятие файла в программирование | 1 | | |
| 99. | Открытие файла в python | 1 | | |
| 100. | Запись в файл в python | 1 | | |
| 101. | Прохождение по элементам списка | 1 | | |
| 102. | Понятие «Список в списке» (двумерный массив) | 1 | | |
| 103. | Перевод элемента списка в тип данных int | 1 | | |
| 104. | Перевод из типа данных str в list | 1 | | |
| 105. | Перевод из типа данных list в str | 1 | | |
| 106. | Знакомство с методом join | 1 | | |
| V | Итоговое занятие | 2 | | |
| 107. | Решение задач | 1 | | |
| 108. | Решение задач. Подведение итогов | 1 | | |

ФОРМЫ АТТЕСТАЦИИ И КОНТРОЛЯ

Виды контроля:

- входящий контроль: осуществляется при комплектовании группы в начале учебного года с целью определения исходного уровня знаний учащихся, определить формы и методы работы с учащимися. Форма контроля: тестирование.

- текущий контроль: осуществляется в процессе проведения опроса учащихся, выполнения практических работ, тестирования, а также выполнения индивидуальных заданий на каждом занятии, а так же по завершении каждой темы — контрольное тестирование;

- промежуточный контроль: проверяется степень усвоения учащимися пройденного за первое полугодие материала;

- итоговая аттестация: выполнение и защита курсовой проект подведение итогов в конце обучения.

Формы контроля:

- педагогическое наблюдение;
- устный опрос;
- выполнение практического задания
- тестирование;
- письменная контрольная работа
- выполнение и защита курсовой проект;
- участие в конкурсах, олимпиадах, соревнованиях.

Способом оценки достижений является гибкая рейтинговая система.

Оценка результативности прохождения программы осуществляется по умению учащихся решать нетривиальные задачи от составления собственного алгоритма до отладки программного кода. Здесь может учитываться не только время, затрачиваемое на выполнение задания, но и оптимальность программного кода, оригинальность решения, соблюдение стандартов – «читаемость» кода.

Для контроля за освоением материала, по завершении каждой темы, проводится контрольное тестирование, которая позволяет оценить степень усвоения материала. Организация контроля знаний происходит на основе анализа контрольных тестов (работ).

К видимым результатам следует отнести выполняемые учащимися курсовые (творческие) проекты. Проекты учащихся могут быть продемонстрированы на итоговом занятии своим друзьям, родителям, учителям. Тем самым достигаются и невидимые внешне результаты, такие как повышение уровня самооценки учащегося, осознание собственной значимости в обществе, умение работать в коллективе.

Программа включает в себя воспитательную работу, направленную на сплочение коллектива, посредством совместных экскурсий, участие в конкурсах, олимпиадах, соревнованиях по программированию различных уровней, в том числе в мероприятия организуемых в заочной форме. Участие в фестивалях и научно-практических конференциях.

Критерии оценки и уровни освоения программного материала

Критерии оценки результативности обучения

- теоретической подготовки обучающихся: соответствие уровня теоретических знаний программным требованиям; широта кругозора; свобода восприятия теоретической информации; развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии;

- практической подготовки учащихся: соответствия уровня развития практических умений и навыков программным требованиям; свобода владения специальным оснащением; качество выполнения практического задания; технологичность практической деятельности;

- развития учащихся: культура организации практической деятельности; культура поведения; творческое отношение к выполнению практического задания; аккуратность и ответственность при работе.

Показатели соответствия теоретической и практической подготовки учащихся определяются степенью освоения программных требований:

- высокий уровень – при успешном освоении более 75% содержания образовательной программы, подлежащей аттестации;

- средний уровень – при успешном освоении от 50% до 75% содержания образовательной программы, подлежащей аттестации;

- низкий уровень – при усвоении менее 50% содержания образовательной программы, подлежащей аттестации.

Требования к результатам выполнения курсового проекта:

- умение планировать и осуществлять проектную и исследовательскую деятельность;
- способность презентовать достигнутые результаты, включая умение определять приоритеты целей с учетом ценностей и жизненных планов;

- самостоятельно реализовывать, контролировать и осуществлять коррекцию своей деятельности на основе предварительного планирования;

- способность использовать доступные ресурсы для достижения целей;

- осуществлять выбор конструктивных стратегий в трудных ситуациях;

- способность создавать продукты своей деятельности, востребованные обществом, обладающие выраженными потребительскими свойствами;

- сформированность умений использовать все необходимое многообразие информации и полученных в результате обучения знаний, умений и компетенций для целеполагания, планирования и выполнения индивидуального проекта.

Итог программы

По окончании обучения по программе учащимся, успешно закончившим обучение, выдается документ (сертификат), установленного образовательным учреждением образца о том, что учащиеся прошли обучение по программе. В документе указываются список изученных тем, названия выполненных проектов, достижения учащегося за период обучения по программе

Материально-техническое обеспечение программы

Для проведения занятий по программе используется специализированный компьютерный класс, имеющий специализированную мебель и технику для проведения занятий:

- учебный компьютерный класс (компьютерные столы + компьютерные стулья 10 мест).

- рабочее место преподавателя (компьютерный стол + компьютерный стул 1 место)

- шкаф для хранения оборудования и методических материалов;

- персональный компьютер. Компьютеры объединены в локальную сеть, подключены к серверу и имеют выход в Интернет. Каждый учащийся – зарегистрированный пользователь сети – имеет сетевой адрес,

пароль и личное пространство на диске.

- мультимедийный проектор + настенно-потолочный экран (1 шт.);
- сканер;
- принтер;
- колонки / наушники.

Информационно-методическое обеспечение программы

Методическая разработка курса содержит:

- лекции по каждой теме, раскрывающие основные вопросы курса, содержащие как теоретический, так и личностно-ориентированный материал;
- схемы и таблицы, иллюстрирующие изучаемые темы и используемые в качестве наглядных пособий в процессе обучения, позволяющие структурировать изучаемый материал, активизировать наглядно-образное мышление учащихся;
- сборники задач, содержащий информацию по отдельным типам задач учебного курса, используемые для самостоятельной и индивидуальной работы учащихся на занятиях;
- задания практикумов, содержащие теоретические и практические вопросы и задачи, позволяющие учащимся закрепить полученные знания, сформировать необходимые навыки;
- практические работы, предусмотренные программой, позволяющие учащимся получить навыки работы со средой программирования, компилятором, написанием, отладкой программы, также для изучения будут использоваться электронные учебники, игровые приложения для изучения языка программирования Python.

Кадровое обеспечение программы

Реализация программы обеспечивается педагогическими кадрами, имеющими среднее профессиональное образование или высшее образование, соответствующее направленности дополнительной общеобразовательной программы.

Требования к педагогам дополнительного образования и преподавателям:

- среднее профессиональное образование – программы подготовки специалистов среднего звена или высшее образование – бакалавриат, направленность (профиль) которого, как правило, соответствует направленности дополнительной общеобразовательной программы;
- дополнительное профессиональное образование – профессиональная переподготовка, направленность (профиль) которой соответствует направленности дополнительной общеобразовательной программы;

При отсутствии педагогического образования – дополнительное профессиональное педагогическое образование; дополнительная профессиональная программа может быть освоена после трудоустройства.

Рекомендуется обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже чем один раз в три года.

СПИСОК ЛИТЕРАТУРЫ

Литература для педагога

1. Федеральный закон № 273-ФЗ от 29.12.2012 «Об образовании в Российской Федерации».
2. СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» (Главный государственный санитарный врач РФ, Постановление от 4 июля 2014 года №41).
3. Бондаренко А.М. Проектная деятельность – запуск механизма развития личности ребёнка // Эксперимент и инновации в школе, 2011. – №3.
4. Гура В.В. Теоретические основы педагогического проектирования личностно-ориентированных образовательных ресурсов и сред. – Ростов н/Д: Изд-во ЮФУ, 2007.
5. Новожилова М.М. Как корректно провести учебное исследование: от замысла к открытию / М. М. Новожилова, С.Г. Воровщиков, И.В. Таврель; науч. ред. Т.И. Шамова. – 3-е изд. – М.: 5 за знания, 2008.
6. Страхова И.А. Проектная деятельность как один из способов формирования универсальных учебных действий // Методист. – 2012. – № 4.
7. Федеральный портал «Российское образование». Каталог образовательных Интернет-ресурсов [Электронный ресурс]. – режим доступа: www.edu.ru. Литература для учащихся
1. Абрамов С.А, Гнездилова Г.Г., Капустина Е.Н., М.И. Селюн. Задачи по программированию. – М.: Наука, 1998.
2. Аллен И. Голуб. С и C++. Правила программирования. – М.: БИНОМ, 1996.
3. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
4. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
5. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.

6. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
7. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
8. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016.
9. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014.