

**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**МКОУ Кипельская средняя общеобразовательная школа**

РАССМОТРЕНО

на заседании ПС

Протокол № 2

от «26» августа 2024 г.

СОГЛАСОВАНО

Зам. по УВР

Клепинина С.В.

от «26» августа 2024 г.

УТВЕРЖДЕНО

Директор школы

Пожарицкая Е.В.

Приказ № 50

от «26» августа 2024 г.

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 64FF60E34844DB0AA86FFB80382BE6F7  
Владелец: Пожарицкая Евгения Викторовна  
Действителен: с 27.07.2023 до 19.10.2024

РАБОЧАЯ ПРОГРАММА

внеурочной деятельности

по информатике

«Юный программист»

6 - 11 классы

Составитель: Лоскутников Виталий Николаевич,

учитель информатики

## **Пояснительная записка**

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования – необходимое условие подготовки современных учащихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

**Актуальность программы** состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

### **Отличительные особенности программы:**

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

**Целесообразность** программы выражена в подборе интерактивных и практикоориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

### **Практическая значимость:**

В рамках предлагаемого курса «Программирования на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и

навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизируют процесс индивидуально-личностного становления учащихся.

### **Адресат программы**

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 12–18 лет, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей учащихся (самих учащихся с 14 лет). Группы формируются из школьников разного возраста на добровольной внеконкурсной основе.

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

### **Методические принципы программы**

- принцип развития: развитие индивидуальных способностей, общей культуры, навыков творческой продуктивной деятельности обучающихся;
- принцип демократизма: право каждого субъекта системы технического творчества на выбор своей траектории развития;
- принцип дифференциации и индивидуализации образования: выявление и развитие способностей обучающихся в области технического творчества, обеспечение их развития в соответствии с потенциалом, индивидуальными возможностями и интересами;
- принцип культуросообразности: ориентация на потребности общества и личности учащихся, единство человека и социокультурной среды, адаптация детей к современным условиям жизни общества;
- принцип единства учебного и воспитательного процесса;
- принцип систематичности и последовательности: вначале используется репродуктивный и интерактивный методы усвоения знаний с постепенным введением проблемного метода обучения, метода проекта.

### **Формы обучения и занятий**

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе – обучение учащихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п. Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

### **Цель и задачи программы**

**Цель:** Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

## **Задачи**

### **1. Образовательные:**

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

### **2. Воспитательные:**

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

### **3. Развивающие:**

- развить познавательные процессы (внимание, восприятие, логическое мышление, память),
- развить креативность,
- развить способности к самореализации.

## **Планируемые результаты**

### ***Личностные результаты***

К личностным результатам освоения информационных и коммуникационных технологий как инструмента в учёбе и повседневной жизни можно отнести:

1. критическое отношение к информации и избирательность её восприятия;
1. уважение к информации о частной жизни и информационным результатам других людей;
1. осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
1. начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

***Метапредметными результатами*** изучения курса являются формирование следующих универсальных учебных действий:

### ***Регулятивные УУД:***

1. планирование последовательности шагов алгоритма для достижения цели; поиск ошибок в плане действий и внесение в него изменений.

### ***Познавательные УУД :***

- моделирование – преобразование объекта из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- анализ объектов с целью выделения признаков (существенных, несущественных);
- синтез – составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбор оснований и критериев для сравнения, сериации, классификации объектов; подведение под понятие; установление причинно-следственных связей;

- построение логической цепи рассуждений.

### **Коммуникативные УУД :**

аргументирование своей точки зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;

1. выслушивание собеседника и ведение диалога;

1. признание возможности существования различных точек зрения и права каждого иметь свою.

### **Предметные результаты:**

Обучающийся:

Овладеет этапами проектной деятельности.

Научится использованию различных методов создания, отладки и корректировки проектов. Научится использованию инструментов встроенного графического редактора.

Получит знания основных видов и задач творческих олимпиад по креативному программированию.

### **Планируемые результаты:**

В результате реализации программы, обучающиеся получат возможность знать: устройство программной среды;

процесс составления алгоритма, блок-схемы;

алгоритм составления проекта; процесс составления алгоритма программы и её отладки; уметь: использовать сенсоры, списки, переменные, случайные числа для составления скриптов; создавать алгоритмы словесно и на языке python;

создавать свои проекты и уметь

презентовать их; работать в команде.

Результативность освоения программы отслеживается на практических занятиях, на которых выполняются определенные задания и после каждого изученного раздела заполняется диагностическая карта успеваемости.

## СОДЕРЖАНИЕ КУРСА

2. История языков программирования. Компиляция и интерпретация.
3. Знакомство с Python и средами программирования.
4. Типы данных в программировании. Определение переменной.
5. Ввод данных с клавиатуры.
6. Логические выражения.
7. Условный оператор. Инструкция if.
8. Множественное ветвление.
9. Цикл While и For.
10. Строки как последовательности символов.
11. Списки — изменяемые последовательности. Массивы.
2. Основные задачи обработки массивов: поиск, сортировка, реверс.
12. Введение в словари.
13. Функции в программировании. Параметры и аргументы функций.
3. Локальные и глобальные переменные. Процедуры.
14. Файлы. Чтение текстового файла. Запись в файл.
15. Алгоритм Евклида (нахождение наибольшего общего делителя)
16. Вычисление факториала на языке программирования Python
17. Двоичный (бинарный) поиск элемента в массиве
18. Замена элементов в списке
19. Перевод чисел из десятичной системы счисления в двоичную
20. Решето Эратосфена - алгоритм определения простых чисел
21. Сортировка выбором (поиск минимума и перестановка)
22. Сортировка методом пузырька
23. Сумма и произведение цифр числа
24. Тестирование простоты числа методом перебора делителей
25. Числа Фибоначчи (вычисление с помощью цикла while и рекурсии)

## УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Тема	Виды учебной деятельности
2.	История языков программирования. Язык Python. Простейшие программы. Реализация вычислений и ветвлений.	знать место языка Python среди языков программирования, особенности структуры программы, представленной на представление о модулях, входящих в состав среды Python, знать воз использования готовых модулей, иметь представление о величине, е что такое операция, операнд и их характеристики, знать принципа структурированных и неструктурированных, иметь предста выражениях и входящих в них операндах, операциях и функ примеры арифметических и логических выражений всех атрибу входить, знать основные операторы языка Python, их синтаксис, процессе исполнения каждого из операторов, знать принципал формальными, локальными и глобальными переменными
3.	Реализация циклических, вспомогательных алгоритмов. Рекурсия.	знать основные операторы языка Python, их синтаксис, иметь пр исполнения каждого из операторов, уметь разрабатывать программы символьной информации, уметь разрабатывать программы (линейн с циклами), иметь представление о значении полноценных пр структурно-ориентированного языка высокого уровня, знать правил функций в Python и построение вызова процедуры, знать обла процедурах, иметь представление о рекурсии, знать ее реализа основными приемами формирования процедуры и функции
4.	Множества. Словари	знать основные операторы языка Python, их синтаксис, иметь пр исполнения каждого из операторов, уметь разрабатывать программы символьной информации, уметь разрабатывать программы (линейн с циклами)
5.	Символьные строки. Обработка символьных строк.	знать основные операторы языка Python, их синтаксис, иметь пр исполнения каждого из операторов, уметь разрабатывать программы символьной информации, уметь разрабатывать программы (линейн с циклами)
6.	Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов.	знать основные операторы языка Python, их синтаксис, иметь пр исполнения каждого из операторов, уметь разрабатывать программы символьной информации, уметь разрабатывать программы (линейн с циклами), определять в программе тип «массив», «матрица», зна «массив», «матрица», уметь воспроизводить алгоритмы сортиров поиска в упорядоченном массиве, распространять эти алгоритмы нечисловых массивах, уметь читать и записывать текстовые файлы в
7.	Итоговое занятие	решать олимпиадные задачи в среде Python.

### Календарно-тематическое планирование

№ п/п	Название раздела, темы
1	<b>История языков программирования. Язык Python. Простейшие программы. Реализ вычислений и ветвлений</b>
2.	История языков программирования
3.	Язык Python
4.	Оператор ввода – input
5.	Оператор вывода – print
6.	Типы данных
7.	Тип данных int (целочисленный)
8.	Операции над типом int (целочисленное деление, остаток от деления)
9.	Решение задач
10.	Тип данных float (вещественный)

11.	Операции над типом float
12.	Решение задач
13.	Ветвление. Полное ветвление
14.	Ветвление. Неполное ветвление
15.	Знакомство с конструкцией if...elif...else
16.	Логические операнды
17.	Решение задач с помощью and, or
18.	Вложенное ветвление
19.	Решение задач
<b>II</b>	<b>Реализация циклических, вспомогательных алгоритмов. Рекурсия</b>
20.	Знакомство с понятием «цикл»
21.	Цикл for, итерация цикла
22.	Решение задач с помощью for
23.	Выход из цикла for с помощью break
24.	Конструкция for...else
25.	Решение задач
26.	Цикл с условием
27.	Решение задач на цикл с условием
28.	Цикл while
29.	Решение задач с помощью while
30.	Выход из цикла while с помощью break
31.	Понятие вложенного цикла
32.	Решение задач на все типы циклов
33.	Понятие функции – def
34.	Описание переменных в функции, локальные и глобальные переменные
35.	Решение задач на вызов функции
36.	Понятие рекурсии в python
37.	Понятие рекурсии в python
38.	Решение типовых задач на рекурсию
39.	Решение задач на замену цикла for рекурсией
40.	Решение задач на замену рекурсии циклом for
<b>III</b>	<b>Множества. Словари</b>
41.	Понятие «Множества» в python
42.	Ввод множества с клавиатуры
43.	Запись в множества с помощью if
44.	Проход по значениям множества с помощью for
45.	Решение задач по теме «Множества»
46.	Понятие «Словари» в python
47.	Ввод с клавиатуры в словари
48.	Понятие значения в словарях
49.	Понятие значения ключа в словарях
50.	Решение задач
51.	Запись в множества через условия
52.	Добавления элементов в множество
53.	Знакомство с методами множества
54.	Решение задач
55.	Проход по ключам-значениям в множестве

56. Решение задач по множествам

**IV**

**Символьные строки. Обработка символьных строк**

57. Ввод с клавиатуры

58. Операции над строками

59. Решение задач

60. Срезы в строках

61. Проход по элементам в строках с помощью for

62. Решение задач

63. Создание пустой строки

64. Добавление элементов в новую строку через условие

65. Вывод элементов строки в одну строку

66. Решение задач

67. Методы строк. Проверка на числа

68. Методы строк. Проверка на буквы

69. Решение задач

70. Методы строк. Проверка на заглавные символы

71. Методы строк. Поиск по значению

72. Методы строк. Метод count

73. Методы строк. Метод reversed

74. Решение задач

75. Методы строк. Перевод из строки в число

76. Методы строк. Перевод из типа int в тип str

77. Решение задач

**V**

**Матрицы. Ввод, вывод, обработка матриц.**

**Чтение и запись текстовых файлов**

78. Понятие массива в программировании

79. Понятие list (список) в python

80. Ввод списка с клавиатуры

81. Вывод списка

82. Решение задач

83. Понятие индекса и значение массива (списка)

84. Методы списка. Add

85. Методы списка. Pop

86. Решение задач

87. Методы списка. Copy

88. Методы списка. Clear

89. Решение задач

90. Методы списка. Reverse

91. Методы списка. Index

92. Решение задач

93. Понятие файла в программировании

94. Открытие файла в python

95. Запись в файл в python

96. Решение задач

97. Прохождение по элементам списка

98. Понятие «Список в списке» (двумерный массив)



100.	Перевод элемента списка в тип данных int
101.	Перевод из типа данных str в list
102.	Перевод из типа данных list в str
103.	Знакомство с методом join

### **Список литературы**

#### ***Нормативно-правовые акты и документы***

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей»,
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.
7. Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

#### ***Литература для родителей и учащихся***

1. Сайт / справочные материалы[Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python[Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
4. 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](http://openbookproject.net), свободный.