

Муниципальное общеобразовательное учреждение
Романовская средняя общеобразовательная школа
р.п. Романовка Романовского района Саратовской области
имени полного кавалера орденов Славы И.В. Серещенко
Центр образования цифрового и гуманитарного профилей «Точка роста»



ПРИНЯТО
на заседании педагогического
совета школы
(Протокол от ____ 20__ г. №__)



**Дополнительная общеобразовательная
общеразвивающая программа
«Учимся программировать на языке PYTHON»**

Направленность: техническая
Возраст обучающихся: 14-17 лет
Срок реализации программы: 1 год

Автор-составитель:
Дедов Юрий Алексеевич,
педагог дополнительного образования

Романовка, 2024

1. Комплекс основных характеристик образовательной программы

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Учимся программировать на языке Python» относится к **технической направленности** и направлена на развитие и поддержку подростков 14-17 лет, проявляющих интерес и определенные способности к программированию.

Уровень реализуемой программы: – базовый.

Актуальность

С развитием современных информационных технологий, сегодня на рынке труда профессия программиста востребована. Актуальность программы отвечает современным потребностям в получении необходимых теоретических знаний и навыков в сфере программирования для личностного развития, ранней профориентации и успешной адаптации к требованиям жизни в современном информационном обществе. Изучение программирования развивает логическое мышление, креативность, исследовательские и творческие качества. Писать программы — занятие увлекательное и временами непростое, однако полученный опыт пригодится и в школе, и дома, да и на рынке труда профессия программиста востребована.

Отличительной особенностью данной программы является выбранный для изучения язык Python – это один из самых популярных, быстро развивающихся языков в современном мире, при этом он имеет достаточно простой синтаксис и подходит для начального обучения. Основное количество часов отводится практическому написанию программ. Каждый обучающийся в результате освоения программы реализует индивидуальный проект. Полученный продукт имеет прикладной характер и может быть использован по необходимости.

В программе предусмотрена **методология наставничества в форме «ученик – ученик»**, так как возраст детей позволяет попробовать себя в роли старшего друга. На первых занятиях по реализации программы предполагается формирование групп «наставник-наставляемые», затем планирование основного процесса работы в течение учебного года, и на последних занятиях курса программы – определение лучшего наставника по итогам года.

Педагогическая целесообразность программы

Программа «Учимся программировать на языке PYTHON» является целостной и непрерывной в течение всего процесса обучения, и позволяет обучающемуся шаг за шагом раскрывать в себе творческие возможности и само реализоваться в современном мире, развить компьютерную грамотность.

Адресат программы

Возраст детей, участвующих в реализации программы с 14 до 17 лет. Дети данного возраста способны на общекультурном уровне выполнять предлагаемые задания по образцу и создавать свои простые программы. Для успешной реализации программы учащиеся должны владеть базовыми навыками работы в текстовом редакторе. Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Наполняемость группы: 12-13 человек.

Объём программы: - 144 часа в год.

Срок реализации программы: – 1 учебный год.

Режим работы: - 2 раза в неделю по 2 учебных часа с 10-минутным перерывом.

Форма обучения: - очная.

1.2. Цель и задачи программы

Цель программы: формирование и развитие имеющихся навыков операционного и логического стиля мышления, исследовательских и творческих качеств подростков; формирование 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 программы:

Обучающие:

- обучить синтаксису языка и основным базовым алгоритмическим конструкциям;
- обучить навыкам разработки, тестирования и отладки несложных программ;
- познакомить с понятием проекта, его структуры, дизайна и разработки;
- формировать мотивы к познавательной и творческой деятельности.

Развивающие:

- развивать логическое мышление и творческие способности;
- способствовать развитию интереса к выбранному профилю деятельности;
- развивать трудолюбие и целеустремленность.

Воспитательные:

- формировать навыки работы в группе;
- формировать культуру общения между учащимися;
- создавать условия для самоопределения учащихся в профессиональном выборе.

Планируемые результаты программы

Предметные результаты:

должны знать

- основы языка программирования Python;
- структуру программы на языке Python;
- типы данных в Python, понятия - переменная, список, кортеж, словарь, множество;
- циклы и ветвления;
- функции и модули в Python;
- зачем нужны классы и объекты в Python;

должны уметь:

- писать самостоятельно несложные программы на языке программирования Python;
- подключать стандартные модули в свои программы;
- работать с переменными, списками, кортежами, словарями, множествами, конструкциями циклов и ветвлений;
- создавать графический интерфейс пользователя (GUI) с помощью модулей;
- создавать простые игры.

Метапредметные:

- умение планировать свою деятельность,
- умение искать информацию и использовать её в планируемой деятельности,
- наличие образного мышления, творческого воображения, художественного вкуса,
- наличие коммуникативных навыков.

Личностные:

- стремление к разумной организации своего свободного времени;
- наличие усидчивости, аккуратности, трудолюбия;
- умение взаимодействовать в коллективе, доброжелательное отношение к сверстникам и взрослым, уважение к чужому труду.

1.3. Содержание программы Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в Python	14	6	8	
1.1	Язык Python. Среда программирования. Особенности ввода вывода. Проведение отбора «Лидер – безынициативный». Формирование групп «наставник-наставляемые»	2	1	1	Анкетирование. Входной тест
1.2.	Типы данных, операции. Оператор присваивания	4	2	2	Практическое задание
1.3.	Числа. Стандартные операции	4	1	3	Практическое задание
2.	Алгоритмические конструкции	22	6	16	
2.1.	Условный оператор	2	2	-	Опрос
2.2.	Цикл while. Теория чисел	10	2	8	Практическое задание
2.3.	Цикл for	10	2	8	Практическое задание
3.	Строки	20	8	12	
3.1.	Литералы строк	4	2	2	Практическое задание
3.2.	Срезы строк	4	2	2	Практическое задание
3.3.	Методы строк	12	4	8	Практическое задание
4.	Функции	20	8	12	

4.1.	Парадигма и преимущества структурного программирования	2	2	-	Опрос
4.2.	Граф и стек вызовов функций. Области видимости переменных	4	2	2	Практическое задание
4.3.	Прямая рекурсия	10	2	8	Практическое задание
4.4.	Косвенная рекурсия	4	2	2	Практическое задание
5.	Списки и кортежи	20	8	12	
5.1.	Списки и кортежи в Python. Сходства и различия	2	2	-	Опрос
5.2.	Операции со списками	10	2	8	Практическое задание
5.3.	Срезы списков	4	2	2	Практическое задание
5.4.	Матрицы. Операции над матрицами.	4	2	2	Практическое задание
6	Словари и множества	8	4	4	
6.1.	Словари	4	2	2	Практическое задание
6.2.	Множества	4	2	2	Практическое задание
7.	Объектно-ориентированное программирование	38	4	34	
7.1.	Классы в Python	2	2	-	Опрос
7.2.	Разработка собственного класса	6	2	4	Практическое задание
7.3.	Разработка и программирование собственного проекта	30	-	30	-
8.	Заключительное занятие	2	-	2	
8.1.	Подведение итогов. Выбор лучшего наставника Индивидуальный проект	2		2	Защита проекта
Всего		144	44	100	

Содержание учебного плана

1. ВВЕДЕНИЕ В PYTHON

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.

1.1. Язык Python. Среда программирования. Особенности ввода вывода

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

Практика. Проведение отбора «Лидер – безынициативный». Формирование групп «наставник-наставляемые».

Входной тест.

Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

1.2. Типы данных, операции. Оператор присваивания

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. АЛГОРИТМИЧЕСКИЕ ИНСТРУКЦИИ

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

2.1. Условный оператор

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

2.2. 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.

2.3. 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.

3. СТРОКИ.

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.

3.1. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

3.2. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

3.3. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. ФУНКЦИИ.

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.

4.1. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.2. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.3. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.4. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

Практические занятия к темам 4.2.,4.3.,4.4.: Нахождение суммы чисел. Вычисление степени. Числа Фибоначчи. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

5. СПИСКИ И КОРТЕЖИ

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.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

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

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

6. СЛОВАРИ И МНОЖЕСТВА

Основные понятия: словарь, множество, ключ, кодирование.

6.1. Словари

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.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП)

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

Практика. Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.

8. ПОДВЕДЕНИЕ ИТОГОВ.

8.1 Выбор лучшего наставника по итогам года.

8.2. Защита индивидуального проекта.

1.4. Формы аттестации / контроля планируемых результатов

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

В процессе реализации программы предусмотрены следующие виды контроля:

- *входной контроль* проводится с целью определения уровня знаний учащихся;
- *промежуточный контроль* проводится регулярно на занятиях с целью определения степени усвоения материала в форме педагогического наблюдения, опроса, собеседования, тестирования и практических заданий;
- *итоговый контроль* – защита проекта.

Формы подведения итогов реализации программы:

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

1.5. Календарный учебный график

Количество учебных недель – 36.

Количество учебных дней – 72.

Учебный период: сентябрь-май.

Календарный учебный график разрабатывается ежегодно и является составной частью рабочей программы. (*Приложение № 1*)

Место проведения занятий – кабинет «Формирования цифровых и гуманитарных компетенций».

2. Комплекс организационно-педагогических условий

2.1. Методическое обеспечение программы

Методические пособия, разработанные с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Электронное приложение к учебникам К. Ю. Полякова Информатика и ИКТ. Набор цифровых образовательных ресурсов – дидактические материалы, интерактивные тесты, анимационные плакаты.

2.2. Условия реализации программы

Материально-техническое обеспечение:

Наименование имущества	Кол-во
Ученические столы	6
Ученические стулья	12
Компьютер	10

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 Гб DDR3;
- Дисковое пространство на менее 128 Гб;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JupyterPyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернета серфинга.

Кадровое обеспечение: педагог дополнительного образования, обладающий знаниями программирования.

2.3. Оценочные материалы

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

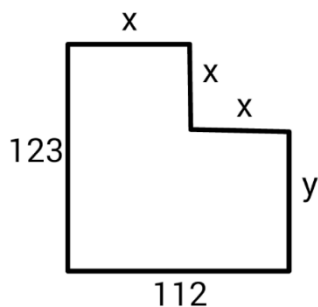
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 14-17 лет

Фамилия, Имя	
--------------	--

1. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



2. 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?

3. 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.

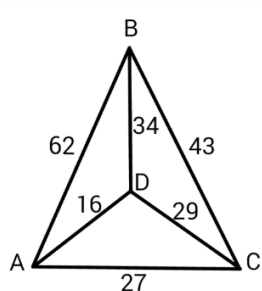
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

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

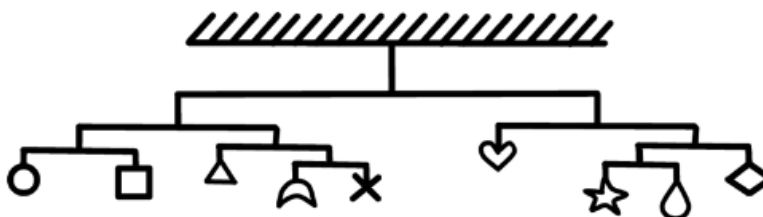
4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке.

Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. 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?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

8. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

Да

Нет

9. 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?

Вопросы для промежуточного контроля по усвоению материала

(Промежуточный контроль проводится в последнюю неделю декабря)

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления: 11111 .1111111
— **произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)**

Запишите число **1.2345e3** в виде десятичной дроби.

Составьте и запишите выражение для вычисления:
2014.0^14(Возвестив 14 степень)
Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

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).

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

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

Укажите результат выражения:
"123" + "42"

Какое значение будет у переменной *i* после выполнения фрагмента программы?
i = 0 while i <= 10:
i = i + 1

if i > 7:
i = i + 2

Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0 while i <= 10:  
i = i + 1 if i > 7:  
i = i + 2
```

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

```
i = 0 while  
i < 5:  
print('*') if i % 2  
== 0:  
print('**') if i > 2:  
print('***') i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0 s = 0 while  
i < 10: i = i + 1  
s = s + i if s  
> 15: break  
i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0 s = 0 while  
i < 10: i = i + 1  
s = s + i if s  
> 15: continue  
i = i + 1
```

Итоговый контроль Диагностическая карта

№/п	Имя	Входящая Диагностика	Промежуточная диагностика	Итоговая диагностика
1				
2				
3				
	Итого 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).

Показателем эффективности реализации наставничества являются повышение мотивации наставляемого в результате работы наставника, выявленные методом опроса, результаты участия в конкурсах разного уровня.

Опрос для наставляемого

	Часто	Средне	Редко	Никогда
1. У меня появилось желание чаще принимать участие в олимпиадах и конкурсах, мероприятиях.				
2. Мне стало легче готовиться к олимпиадам и конкурсам				
3. Я хочу принять участие в олимпиадах и конкурсах, мероприятиях, в которых ранее не принимал участие				
4. Я понимаю, зачем мне нужно участвовать в олимпиадах, конкурсах, мероприятиях.				

Список литературы

Для педагога

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.

Для родителей и обучающихся

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

**Календарный учебный график
дополнительной общеобразовательной общеразвивающей программы
«Учимся программировать на языке PYTHON»**

№ п/п	Дата Планируемая (число, месяц)	Дата Фактическая (число, месяц)	Тема занятия	Кол-во часов	Форма занятия	Формы контроля
1			Вводное занятие. Инструктаж по ТБ. Входной тест. Проведение отбора «Лидер – безынициативный». Формирование групп «наставник-наставляемые» Язык программирования Python. Достоинства и недостатки.	1	Теоретическое	Педагогическое наблюдение, собеседование, тест
2			Области применения. Интерактивный режим работы программы.	1	Комбинированное	Педагогическое наблюдение, собеседование
3			Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.	1	Практическое	Практическая работа
4			Особенности ввода-вывода. Входной контроль.	1	Комбинированное	Практическая работа Тест
5-6			Ввод и вывод числовой информации.	2	Теоретическое	Педагогическое наблюдение, собеседование
7-8			Тренировочное задание на ввод и вывод числовой информации.	2	Практическое	Практическая работа
9-10			Стандартные операции с целыми и действительными числами. Стилль программирования Python (PEP 8).	2	Теоретическое	Педагогическое наблюдение, собеседование

11-14			Решение простых задач в интерактивном режиме.	4	Практическое	Практическая работа
15-16			Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.	2	Теоретическое	Опрос
17-18			Цикл while. Теория чисел. Фильтрация потока чисел.	2	Теоретическое	Педагогическое наблюдение, собеседование
19-20			Решение задач на анализ чисел потока и целочисленной арифметики (НОД и НОК)	2	Практическое	Практическая работа
21-22			Решение задач на анализ чисел потока и целочисленной арифметики (проверка числа на простоту)	2	Практическое	Практическая работа
23-24			Решение задач на анализ чисел потока и целочисленной арифметики (Алгоритм Евклида)	2	Практическое	Практическая работа
25-26			Решение задач на анализ чисел потока и целочисленной арифметики (Нахождение максимума и минимума)	2	Практическое	Практическая работа
27-28			Цикл for. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.	2	Теоретическое	Педагогическое наблюдение, собеседование
29-30			Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (НОД и НОК).	2	Практическое	Практическая работа
31-32			Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Проверка числа на простоту).	2	Практическое	Практическая работа
33-34			Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Алгоритм Евклида).	2	Практическое	Практическая работа

35-36			Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Нахождение максимума и минимума).	2	Практическое	Практическая работа
37-38			Понятие «литералы строк».	2	Теоретическое	Педагогическое наблюдение, собеседование
39-40			Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.	2	Практическое	Практическая работа
41-42			Форматирование строки.	2	Теоретическое	Педагогическое наблюдение, собеседование
43-44			Преобразование строки. Решение упражнений.	2	Практическое	Практическая работа
45-46			Методы работы со строкой.	2	Теоретическое	Педагогическое наблюдение, собеседование
47-48			Методы работы со строкой. Решение задач.	2	Практическое	Практическая работа
49-50			Применение методов строки.	2	Теоретическое	Педагогическое наблюдение, собеседование
51-56			Применение методов строки. Решение задач.	6	Практическое	Практическая работа
57-58			Обзор парадигм программирования. Особенности применения языков программирования.	2	Теоретическое	Опрос
59-60			Применение стек и граф вызовов функций.	2	Теоретическое	Педагогическое наблюдение, собеседование

61-62			Выполнение тренировочных упражнений.	2	Практическое	Практическая работа
63-64			Понятие прямой рекурсии и ее применение.	2	Теоретическое	Педагогическое наблюдение, собеседование

65-72			Выполнение тренировочных упражнений.	8	Практическое	Практическая работа
73-74			Понятие косвенной рекурсии и ее применение.	2	Теоретическое	Педагогическое наблюдение, собеседование
75-76			Выполнение тренировочных упражнений.	2	Практическое	Практическая работа
77-78			Представление списка и кортежа в памяти компьютера, сходства и различия.	2	Теоретическое	Опрос
79-80			Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).	2	Теоретическое	Педагогическое наблюдение, собеседование
81-88			Решение задач на ввод-вывод элементов одномерного и многомерного списка.	8	Практическое	Практическая работа
89-90			Методы работы со списком и кортежем. Методы сортировки списка	2	Теоретическое	Педагогическое наблюдение, собеседование
91-92			Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.	2	Практическое	Практическая работа

93-94			Вычисление сложности алгоритма. Многомерные списки.	2	Теоретическое	Педагогическое наблюдение, собеседование
95-96			Решение задач повышенной трудности.	2	Практическое	Практическая работа
97-98			Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями.	2	Теоретическое	Педагогическое наблюдение, собеседование
			Кодирование и декодирование текста.			
99-100			Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.	2	Практическое	Практическая работа
101-102			Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.	2	Теоретическое	Педагогическое наблюдение, собеседование
103-104			Решение задач повышенной трудности.	2	Практическое	Практическая работа
105-106			Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «событие».	2	Теоретическое	Опрос
107-108			Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.	2	Теоретическое	Педагогическое наблюдение, собеседование
109-112			Создание собственного класса.	4	Практическое	Практическая работа
113-142			Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.	30	Практическое	Практическая работа

143-144			Защита индивидуального проекта. Выбор лучшего наставника	2	Практическое	Практическая работа
			Итого	144		