

МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «СРЕДНЯЯ  
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА с.НОВАЯ ПОРУБЕЖКА ПУГАЧЕВСКОГО  
РАЙОНА САРАТОВСКОЙ ОБЛАСТИ»

Рекомендовано к утверждению  
на заседании методического совета  
МОУ «СОШ с.Новая Порубежка»  
« 30 » августа 20 21 г.  
Протокол № 1

Утверждаю  
Директор  
МОУ «СОШ с.Новая Порубежка»  
И.А. Палагутина  
« 31 » августа 20 21 г.  
Приказ № 980



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

Целевая аудитория: обучающиеся 10 – 17 лет  
Срок реализации: 1 год

Педагог дополнительного образования:  
Палагутин Иван Александрович

с.Новая Порубежка - 2021

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

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

Программа разработана в соответствии со следующими нормативными документами:

- Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Распоряжение Правительства Российской Федерации от 28 июля 2017 г. № 1632-р «Об утверждении программы «Цифровая экономика Российской Федерации»;
- Концепция развития дополнительного образования детей, распоряжение Правительства Российской Федерации от 4 сентября 2014г. № 1726-р;
- Приказ Минпросвещения России от 9 ноября 2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) Письмо Министерства образования и науки России от 18 ноября 2015 года №09-3242;
- СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

В современных условиях «Цифровая экономика» - один из стратегических принципов развития Российской Федерации. Федеральный проект «Кадры для цифровой экономики» национальной программы «Цифровая экономика Российской Федерации» и государственная программа Российской Федерации «Развитие образования» направлены на совершенствование системы образования, которая должна создать ключевые условия для подготовки кадров и обеспечивать цифровую экономику компетентными кадрами. В условиях модернизации российского образования наиболее остро стоит вопрос о поисках резервов совершенствования качества образования, в том числе резервов дополнительного образования в интересах цифровой экономики.

Дополнительное образование, дополняя базовое образование, может быть направлено на формирование у обучающихся представлений о принципах функционирования вычислительных систем, возможностях программирования для развития логического мышления; способностей к формализации; элементов системного мышления и воспитания чувства ответственности за результаты своего труда; установки на недопустимость действий, нарушающих правовые, этические нормы; развития творческих способностей и самостоятельности при решении различных вопросов.

«Основы программирования на языке Python» рассматривается как базовый курс, развивающий знания школьников в области владения вычислительной техникой и информационно-коммуникационных технологий, а также дающий базовые знания о программировании.

## **Цель программы**

Личностное и интеллектуальное развитие обучающихся в процессе обучения основам программирования на языке Python.

## **Задачи программы**

### *1. Обучающие*

- Обучение синтаксису языка и основным базовым алгоритмическим конструкциям.
- Обучение навыкам разработки, тестирования и отладки несложных программ.
- Знакомство с понятием проекта, его структуры, дизайна и разработки.
- Формирование мотивов к познавательной и творческой деятельности.

### *2. Воспитательные*

- Создание условий для формирования навыков работы в группе.
- Создание условий для формирования культуры общения между учащимися.
- Создания условий для возникновения потребности к самостоятельному изучению материала.
- Создание условий для самоопределения обучающихся в профессиональном выборе.

### *3. Развивающие*

- Развитие логического мышления.
- Развитие творческих способностей.
- Развитие интереса к выбранному профилю деятельности.
- Развитие трудолюбия и целеустремленности.

## **Отличительные особенности программы**

В связи с отсутствием утвержденных Министерством образования и науки программ по программированию для учреждений дополнительного образования была взята за основу программа для высшей школы.

Программа предусматривает подготовку обучающихся в области вычислительной техники и программирования.

Содержание программы систематизирует и расширяет знания по основам функционирования компьютерных систем. Программа позволяет слушателям с минимальным знанием информатики освоить основы программирования, научиться проектировать и разрабатывать приложения, используя базовые возможности языка программирования Python.

В основу программы положены такие принципы как:

- *Целостность и непрерывность*, означающие, что данная ступень является важным звеном единой подготовки школьников в сфере информационных технологий.
- *Научность в сочетании с доступностью*, строгость и систематичность изложения (включение в содержание фундаментальных положений современной

- науки с учетом возрастных особенностей обучаемых).
- *Практико-ориентированность*, обеспечивающая отбор содержания, направленного на решение практических задач. При этом исходным является положение о том, что компьютер может многократно усилить возможности человека, но не заменить его.
  - *Принцип развивающего обучения* (обучение ориентировано не только на получение новых знаний в области программирования, но и на активизацию мыслительных процессов, формирование и развитие у воспитанников обобщенных способов деятельности, формирование навыков самостоятельной работы).

Используемые в программе задачи взяты из различных источников: сайты с задачами по программированию; печатная литература; задачи составленные разработчиком курсов.

Программа *вариативная* так, как в рамках ее содержания можно разрабатывать разные учебно-тематические планы и для ее освоения возможно выстраивание индивидуальных программ, индивидуальных траекторий (маршрутов) обучения. Программа *открытая*, предполагает совершенствование, изменение в соответствии с потребностями обучающихся и тенденциями развития этой области знаний.

Настоящая программа рассчитана на работу в детском объединении в системе дополнительного образования.

**Уровень программы базовый.**

#### **Адресат программы**

Программа предназначена для обучающихся 10 - 17 лет, в том числе без предыдущего опыта работы с персональным компьютером.

#### **Срок освоения программы**

9 месяцев (36 недель). Программа реализуется в течение всего календарного года, включая каникулярное время, с 1 сентября по 31 мая.

#### **Формы и виды занятий**

Обучение очное с элементами дистанционного обучения. Формы занятий: лекции, индивидуальные практические занятия, индивидуальные консультации, групповые консультации, групповые практические занятия.

#### **Режим занятий**

Занятия проводятся в очной форме еженедельно. Занятия проводятся один раз в неделю по 2 часа. Продолжительность одного академического часа — 45 минут.

**Численность обучающихся в группе 8 – 12 человек.**

#### **Объём программы**

Распределение часов по темам дано из расчета максимум 72 часа в год.

## СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНЫЙ ПЛАН

№ п/п	Тематический блок	Количество часов			Форма контроля итогов
		теория	пр-ка	всего	
1.	Классификация языков программирования. Компиляция, интерпретация. Среда разработки	1	1	2	практическое задание
2.	Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов	2	2	4	программный продукт
3.	Встроенные типы данных. Переменные. Основные операции с данными	3	5	8	программный продукт
4.	Ввод/вывод данных в программу	2	2	4	программный продукт
5.	Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор	2	2	4	программный продукт
6.	Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python	2	2	4	программный продукт
7.	Управляющие инструкции – циклы	3	7	10	программный продукт
8.	Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.	4	6	10	программный продукт
9.	Строки. Списки. Словари. Кортежи	6	6	12	программный продукт
10.	Файловый ввод/вывод	2	2	4	программный продукт
11.	Основы объектно-ориентированного программирования	3	3	6	программный продукт
12.	Итоговая аттестация		4	4	программный продукт
	<b>ИТОГО</b>	<b>30</b>	<b>42</b>	<b>72</b>	

## УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
<b>1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки</b>		<b>1</b>	<b>1</b>	<b>2</b>
1	Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Области применения.	1		1
2	<i>Дистрибутивы интерпретатора языка Python. Установка официального дистрибутива интерпретатора языка Python 3.x. Запуск интерактивного режима Python. Работа в официальном IDLE Python. Работа в среде PyScripter, запуск/остановка скриптов на языке Python.</i>		1	1
<b>2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов</b>		<b>2</b>	<b>2</b>	<b>4</b>
3	Документация на язык программирования – PEP8. Стиль написания кода: отступы. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Комментарии.	1		1
4	Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.	1		1
5	<i>Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().</i>		1	1
6	<i>Выполнение команд в интерактивном режиме. Создание скриптов. Сохранение скриптов. Программа приветствие.</i>		1	1
<b>3. Встроенные типы данных. Переменные. Основные операции с данными</b>		<b>3</b>	<b>5</b>	<b>8</b>
7	Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. <i>Основные операции с числовыми данными.</i>	1	1	2
8	<i>Базовые функции языка Python для работы с числовыми данными. Ввод/вывод числовых данных. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.</i>		1	1
9	Основы строкового типа данных. <i>Склеивание строк.</i>	1	1	2

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
10	Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций.	1		1
11	<i>Использование логических операций. Взаимодействие с числовыми и строковыми типами данных. Битовые операции.</i>		2	2
<b>4. Ввод/вывод данных в программу.</b>		<b>2</b>	<b>2</b>	<b>4</b>
12	Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). <i>Использование функции input(). Преобразование типов вводимых данных.</i>	1	1	2
13	Функция print() для вывода данных на экран и в файл. <i>Использование функции print(). Формирование строки. Вывод псевдографики.</i>	1	1	2
<b>5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор.</b>		<b>2</b>	<b>2</b>	<b>4</b>
14	Понятие алгоритма их типы. Описание задачи в виде алгоритма.	1		1
15	Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. <i>Применение условного оператора if...else для решения прикладных задач.</i>	1	1	2
16	<i>Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.</i>		1	1
<b>6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.</b>		<b>2</b>	<b>2</b>	<b>4</b>
17	Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы.	2		2
18	<i>Разработка алгоритма программ с учетом возможных ошибок выполнения. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам.</i>		1	1

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
19	<i>Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.</i>		1	1
<b>7. Управляющие инструкции – циклы.</b>		<b>3</b>	<b>7</b>	<b>10</b>
20	Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций.	1		1
21	Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. <i>Решение задач с помощью цикла for.</i>	1	2	3
22	Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. <i>Бесконечный цикл. Формулировка условия. Решение задач с использованием циклов while.</i>	1	2	3
23	<i>Решение задач различного типа с помощью циклических алгоритмов.</i>		3	3
<b>8. Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.</b>		<b>4</b>	<b>6</b>	<b>10</b>
24	Понятие функция. Назначение. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям.	2		2
25	Математические функции. Подключение модулей. Модуль Math. <i>Использование функций модуля Math для решения математических задач.</i>	1	1	2
26	Создание собственных функций: определение и использование. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. <i>Использование функций в программах. Создание собственных модулей.</i>	1	1	2
27	<i>Создание игры «Волк, коза и капуста» с использованием циклического алгоритма.</i>		2	2
28	<i>Знакомство с модулем NumPy и matplotlib.</i>		1	1
29	<i>Знакомство с модулем tkinter.</i>		1	1
<b>9. Строки. Списки. Словари. Кортежи.</b>		<b>6</b>	<b>6</b>	<b>12</b>



№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
30	Строка как последовательность. Функция len. Перебор элементов строки. Срезы строк. Поиск. Подсчет. <i>Методы строк. Сравнение строк. Работа с текстовыми файлами.</i>	1	1	2
31	Списки. Перебор элементов списка. Операции со списками. Срезы списков. <i>Методы списков. Отображение, фильтрация и сокращение. Удаление элементов списка. Создание синонимов в списке.</i>	1	1	2
32	Словари. Словари как счетчики. Циклы и словари. Обратный поиск. <i>Словари и списки. Глобальные переменные.</i>	2	2	4
33	Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. <i>Кортежи с переменным числом аргументов. Списки и кортежи. Словари и кортежи. Сравнение кортежей. Последовательности последовательностей.</i>	2	2	4
<b>10. Файловый ввод/вывод.</b>		<b>2</b>	<b>2</b>	<b>4</b>
34	Понятие файл. Имена файлов. Место положение файлов. <i>Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл.</i>	1	1	2
35	Базы данных. <i>Сериализация. Десериализация. Конвейеры.</i>	1	1	2
<b>11. Основы объектно-ориентированного программирования.</b>		<b>3</b>	<b>3</b>	<b>6</b>
36	Основы объектно-ориентированного подхода. Объекты в реальном мире, объекты в Python. Объект = атрибуты + методы. <i>Создание объектов. Примеры класса.</i>	1	1	2
37	Скрываем данные. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения. <i>Использование классов в программах.</i>	2	2	4
<b>12. Итоговая аттестация.</b>			<b>4</b>	<b>4</b>
<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>			4	4
<b>ИТОГО</b>		<b>30</b>	<b>42</b>	<b>72</b>

## СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

### **1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки.**

#### Теория:

Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим. Официальный сайт языка Python. Дистрибутивы языка Python.

#### Практические занятия:

1. Дистрибутивы интерпретатора языка Python.
2. Установка официального дистрибутива интерпретатора языка Python 3.x.
3. Запуск интерактивного режима Python.
4. Работа в официальном IDLE Python.
5. Работа в среде PyScripter, настройка, запуск/остановка скриптов на языке Python.

### **2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов.**

#### Теория:

Документация на язык программирования – PEP8. Стиль написания кода: отступы, базовый синтаксис. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Ключевые (служебные/зарезервированные слова). Первая программа. Сообщения интерпретатора.

#### Практические занятия:

1. Первая программа.
2. Подсветка синтаксиса в средах IDLE.
3. Сообщения интерпретатора.
4. Знакомство с функциями print() и input().
5. Выполнение команд в интерактивном режиме.
6. Создание скриптов. Сохранение скриптов.
7. Программа приветствие.

### **3. Встроенные типы данных. Переменные. Основные операции с данными.**

#### Теория:

Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754. Основы строкового типа данных. Склеивание

строк. Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ.

#### Практические занятия:

1. Основные операции с числовыми данными.
2. Базовые функции языка Python для работы с числовыми данными.
3. Ввод/вывод числовых данных.
4. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.
5. Склеивание строк.
6. Использование логических операций.
7. Взаимодействие с числовыми и строковыми типами данных.
8. Битовые операции.

#### **4. Ввод/вывод данных в программу.**

##### Теория:

Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Функция print() для вывода данных на экран и в файл. Формирование строки. Вывод псевдографики.

#### Практические занятия:

1. Использование функции input().
2. Преобразование типов вводимых данных.
3. Использование функции print().
4. Формирование строки.
5. Вывод псевдографики.

#### **5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием.**

##### **Условный оператор.**

##### Теория:

Понятие алгоритма их типы. Описание задачи в виде алгоритма. Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий.

#### Практические занятия:

1. Применение условного оператора if...else для решения прикладных задач.
2. Вложенные условия.
3. Множественные условия.
4. Ленивая оценка условий.
5. Сложные условия.
6. Формулировка условий.

7. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.

## **6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.**

### Теория:

Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам. Использование исключений для обработки ошибок выполнения без аварийного завершения программы. Разработка алгоритма программ с учетом возможных ошибок выполнения. Отладка программы с семантическими ошибками.

### Практические занятия:

1. Сообщения интерпретатора об ошибках.
2. Поиск документации по ошибкам.
3. Разработка алгоритма программ с учетом возможных ошибок выполнения.
4. Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.

## **7. Управляющие инструкции – циклы.**

### Теория:

Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций. Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. Бесконечный цикл. Формулировка условия. Составление оптимального алгоритма программы.

### Практические занятия:

1. Решение задач с помощью цикла for.
2. Бесконечный цикл.
3. Формулировка условия.
4. Решение задач с использованием циклов while.
5. Решение задач различного типа с помощью циклических алгоритмов.

## **8. Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.**

### Теория:

Понятие функция. Назначение. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям. Математические функции. Подключение модулей. Модуль Math. Создание собственных функций: определение и использование. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. Создание собственных модулей.

### Практические занятия:

1. Использование функций модуля Math для решения математических задач.
2. Использование функций в программах.
3. Создание собственных модулей.
4. Создание игры «Волк, коза и капуста» с использованием циклического алгоритма.
5. Знакомство с модулем NumPy и matplotlib.
6. Знакомство с модулем tkinter.

## **9. Строки. Списки. Словари. Кортежи.**

### Теория:

Строка как последовательность. Функция len. Срезы строк. Методы строк. Работа с текстовыми файлами. Списки. Перебор элементов списка. Операции со списками. Срезы списков. Отображение, фильтрация и сокращение. Удаление элементов. Создание синонимов. Словари. Словари как счетчики. Циклы и словари. Обратный поиск. Словари и списки. Глобальные переменные. Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. Кортежи с переменным числом аргументов.

### Практические занятия:

1. Методы строк.
2. Сравнение строк.
3. Работа с текстовыми файлами.
4. Методы списков.
5. Отображение, фильтрация и сокращение.
6. Удаление элементов списка.
7. Создание синонимов в списке.
8. Словари и списки.
9. Глобальные переменные.
10. Кортежи с переменным числом аргументов.
11. Списки и кортежи.
12. Словари и кортежи.
13. Сравнение кортежей.
14. Последовательности последовательностей.

## **10. Файловый ввод/вывод.**

### Теория:

Понятие файл. Имена файлов. Место положение файлов. Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл. Базы данных. Сериализация. Десериализация. Конвейеры.

### Практические занятия:

1. Открытие файла.
2. Чтение файла.

3. Текстовые и бинарные файлы.
4. Запись в файл.
5. Сериализация.
6. Десериализация.
7. Конвейеры.

## **11. Основы объектно-ориентированного программирования.**

### Теория:

Основы объектно-ориентированного подхода. Объекты в реальном мире, объекты в Python. Объект = атрибуты + методы. Создание объектов. Скрываем данные. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения.

### Практические занятия:

1. Создание объектов.
2. Примеры класса.
3. Использование классов в программах.

## **12. Итоговая аттестация**

### Практические занятия:

1. Индивидуальное задание.
2. Разработка концепции программы по заданию.
3. Написание и отладка программы.
4. Презентация программного продукта.

## ПРОГНОЗИРУЕМЫЕ РЕЗУЛЬТАТЫ

### *1. Обучающие*

- знание синтаксиса языка и базовых алгоритмических конструкций, умение применять теоретические знания для решения конкретных проблем;
- умение разрабатывать, тестировать и отлаживать несложные программы;
- понимание, что такое проект, структура и дизайн проекта;
- устойчивая мотивация к познавательной и творческой деятельности.

### *2. Воспитательные*

- сформированность коммуникативной культуры обучающихся, навыков работы в команде;
- самостоятельность в изучении дополнительного материала по пройденным темам;
- профориентация.

### *3. Развивающие*

- навыки применения логики при решении поставленных задач;
- проявление творческой активности обучающихся.
- создание продуктов оригинальной, творческой деятельности;
- проявление интереса обучающихся к программированию;
- проявление трудолюбия и целеустремленности.

Основными критериями оценки эффективности образовательного процесса являются:

- степень сформированности основных знаний, умений и навыков, предусмотренных программой;
- способность практически применять знания при создании программного продукта, при решении задач;
- писать программы без ошибок или устранять ошибки при отладке программы.

Обучающиеся, освоившие программу в полном объеме, должны:

- уметь качественно формулировать и излагать свои мысли;
- уметь работать как самостоятельно, так и в команде;
- знать средства языка программирования Python в рамках курса;
- организовывать самостоятельное обучение, стремиться к самосовершенствованию;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

# КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

## УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### Материально-техническое обеспечение

Для реализации программы имеется компьютерный класс, мультимедийный проектор, необходимое программное обеспечение (среда программирования языка Python, операционная система Windows, пакет Microsoft Office), компьютеры, принтер и копировальный аппарат, интернет.

### Кадровое обеспечение

Дополнительную общеразвивающую программу реализуют педагоги дополнительного образования с образованием в области программирования.

### Информационное обеспечение

Официальный сайт языка Python - <https://www.python.org/>

Интерактивный учебник языка Питон - <http://pythontutor.ru/>

Python 3 для начинающих - <https://pythonworld.ru/>

База знаний, примеры по программированию на языке Python - <https://pythonru.com>

Сайт «Школа программиста» — <https://acmp.ru/>

Подборка материалов по языку Python - <https://proglib.io/p/learning-python/>

Выполнение программы онлайн - <https://rextester.com/1/python3>

## ФОРМЫ АТТЕСТАЦИИ

Программа предполагает обязательную промежуточную и итоговую аттестацию по всем темам программы обучения, для осуществления мониторинга освоения образовательной программы. Помимо итоговой аттестации проводятся периодические проверки знаний в форме индивидуальных заданий после изучения одной или нескольких тем, с целью выявить возможные проблемы с усвоением материала курса.

Итоговая аттестация предполагает дифференциальный зачет в виде презентации программы. Обучающимся предлагается небольшое индивидуальное задание для создания программы, предполагается игра с графическим интерфейсом. По результатам аттестации учащийся может получить одну из трёх оценок: *незачёт*, *зачёт* и *зачёт с повышенным освоением программы*. Критерием проверки индивидуальных работ: программа записана без ошибок, программа написана полностью самостоятельно, или взята готовая, понята и модифицирована. Повышенным уровнем освоения программы будет считаться применение оригинальных решений в итоговой работе.



## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### Системный мониторинг результативности обучения по дополнительной общеразвивающей программе «Основы программирования на языке Python»

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
Обучающие			
Обучение синтаксису языка и основным базовым алгоритмическим конструкциям	Знание синтаксиса языка и базовых алгоритмических конструкций, умение применять теоретические знания для решения конкретных проблем.	Использование знаний, умений, навыков в самостоятельной деятельности и выполнении работ по образцу. Умение работать самостоятельно и в группе. Расширение кругозора в различных областях знаний. Способность передавать знания сверстникам.	Анализ занятий. Наблюдение. Анализ продуктов деятельности воспитанников. Система практических работ, выполнение индивидуальных заданий по изученным темам.
		Создание программ без постоянного обращения к документации. Понимание чужого кода и самостоятельная модификация готовых программ.	
Обучение навыкам разработки, тестирования и отладки несложных программ	Умение разрабатывать, тестировать и отлаживать несложные программы.	Использование знаний, умений, навыков по разработке, тестированию и отладке программ.	Анализ занятий. Наблюдение.
		Обнаружение и устранение ошибок различного типа в программном коде. Сокращение ошибок в текстах программ.	
Знакомство с понятием проекта, его структуры, дизайна и разработки	Понимание, что такое проект, структура и дизайн проекта.	Умение разработать простой проект самостоятельно или при помощи педагога. Умение работать самостоятельно и в группе при разработке групповых проектов.	Анализ занятий. Анализ продуктов деятельности воспитанников. Система практических работ, выполнение индивидуальных заданий по изученным темам
		Планирование структуры приложения. Декомпозиция задачи на части. Успешная работа в коллективе над различными частями проекта.	

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
Формирование мотивов к познавательной и творческой деятельности	Устойчивая мотивация обучающихся к познавательной и творческой деятельности.	Принятие образовательной программы и способность обучающихся осваивать её. Потребность заниматься в профиле сверх предложенной образовательной программы. Проявление обучающимися инициативы и способности реализовывать свои идеи в творческой деятельности.	Изучение контингента обучающихся (беседа). Удовлетворенность обучающихся занятиями (беседа).
		Проявление интереса и самостоятельное изучение новых тем и расширение знаний по изученным в аудитории темам.	
Воспитательные			
Создание условий для формирования навыков работы в группе	Сформированность навыков работы в группе.	Наличие коммуникативной культуры обучающихся. Взаимопомощь.	Наблюдение. Беседа и тестирование на выявление коммуникативных склонностей.
		Уважительное и корректное отношение к одноклассникам. Навык формулирования ответов и вопросов к одноклассникам и преподавателям.	
Создание условий для формирования культуры общения между учащимися	Сформированность коммуникативной культуры обучающихся	Наличие коммуникативной культуры обучающихся.	Наблюдение на уровень воспитанности.
		Отсутствие конфликтов в группах.	
Создания условий для возникновения потребности к самостоятельному изучению материала	Сформированность умения самостоятельно изучать дополнительный материал по пройденным темам.	Степень самостоятельности (участие педагога) Качество усвоения.	Самоанализ Беседа Проверка работ
		Самостоятельный поиск решений поставленной задачи. Понимание чужих решений и способность своей интерпретации.	
Создание условий для самоопределения обучающихся в профессиональном выборе	Направленность личности в профессиональном выборе.	Наличие обучающихся, выбравших допрофессиональную или профессиональную подготовку в соответствии с осваиваемым предметом.	Беседы на выявление мотивов обучающихся к занятию по данному профилю, беседы на выявление профессиональных намерений. Статистический учет профессионального выбора обучающихся.
		Оценка перспективы применения полученных знаний в дальнейшей учебной и профессиональной деятельности.	
Развивающие			
Развитие логического мышления	Применение логики при решении поставленных задач	Способность использовать логику при решении конкретных задач	Наблюдение. Анализ продуктов деятельности детей.

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
		Выбор оптимального метода решения задачи.	
Развитие творческих способностей	Проявление творческой активности обучающихся. Наличие продуктов оригинальной, творческой деятельности обучающихся.	<p>Постановка необычных, нестандартных вопросов и проблем. Большое количество выдвигаемых идей, нестандартных решений, замыслов.</p> <p>Использование оригинальных решений для рутинных задач, комбинирование условий.</p>	Анализ и оценивание детских творческих достижений.
Развитие интереса обучающихся к выбранному ими профилю деятельности	Развитие интереса обучающихся к программированию	<p>Устойчивость интереса обучающихся к программированию. Наличие внутренней мотивации у обучающихся для занятий программированием.</p> <p>Удовлетворенность обучающихся собственными достижениями. Успешность в освоении образовательной программы. Активность в образовательном процессе.</p> <p>Применение знаний полученных в процессе обучения в решении смежных задач в процессе обучения и для бытового применения.</p>	<p>Беседы с учащимися.</p> <p>Беседы на выявление мотивов обучающихся к занятию по данному профилю, беседы на выявление профессиональных намерений.</p> <p>Статистический учет сохранности контингента.</p> <p>Анализ мотивов ухода обучающихся из коллектива.</p>
Развитие трудолюбия и целеустремленности	Развитие трудолюбия и целеустремленности	<p>Целеустремленность, трудолюбие, проявляемые при выполнении индивидуальных заданий.</p> <p>Учащийся не бросает попытки решить сложные задачи.</p>	Наблюдение.

**Личная карта учащегося по дополнительной общеразвивающей программе  
«Основы программирования на языке Python»  
(заполняется педагогом как дневник наблюдений)**

**Ф.И. ученика** \_\_\_\_\_

Параметры	Критерии	1-е полугодие	2-е полугодие
Знание синтаксиса языка и базовых алгоритмических конструкций, умение применять теоретические знания для решения конкретных проблем.	Использование знаний, умений, навыков в самостоятельной деятельности и выполнении работ по образцу. Умение работать самостоятельно и в группе. Расширение кругозора в различных областях знаний. Способность передавать знания сверстникам.		
	Создание программ без постоянного обращения к документации. Понимание чужого кода и самостоятельная модификация готовых программ.		
Умение разрабатывать, тестировать и отлаживать несложные программы.	Использование знаний, умений, навыков по разработке, тестированию и отладке программ.		
	Обнаружение и устранение ошибок различного типа в программном коде. Сокращение ошибок в текстах программ.		
Понимание, что такое проект, структура и дизайн проекта.	Умение разработать простой проект самостоятельно или при помощи педагога. Умение работать самостоятельно и в группе при разработке групповых проектов.		
	Планирование структуры приложения. Декомпозиция задачи на части. Успешная работа в коллективе над различными частями проекта.		
Устойчивая мотивация обучающихся к познавательной и творческой деятельности.	Принятие образовательной программы и способность обучающихся осваивать её. Потребность заниматься в профиле сверх предложенной образовательной программы. Проявление обучающимися инициативы и способности реализовывать свои идеи в творческой деятельности.		
	Проявление интереса и самостоятельное изучение новых тем и расширение знаний по изученным в аудитории темам.		
Сформированность навыков работы в группе.	Наличие коммуникативной культуры обучающихся. Взаимопомощь.		
	Уважительное и корректное отношение к одноклассникам. Навык формулирования ответов и вопросов к одноклассникам и преподавателям.		
Сформированность коммуникативной культуры обучающихся	Наличие коммуникативной культуры обучающихся.		
	Отсутствие конфликтов в группах.		
Сформированность умения самостоятельно изучать дополнительный материал по пройденным темам.	Степень самостоятельности (участие педагога) Качество усвоения.		
	Самостоятельный поиск решений поставленной задачи. Понимание чужих решений и способность своей интерпретации.		

Параметры	Критерии	1-е полугодие	2-е полугодие
Направленность личности в профессиональном выборе.	Наличие обучающихся, выбравших допрофессиональную или профессиональную подготовку в соответствии с осваиваемым предметом.		
	Оценка перспективы применения полученных знаний в дальнейшей учебной и профессиональной деятельности.		
Применение логики при решении поставленных задач	Способность использовать логику при решении конкретных задач		
	Выбор оптимального метода решения задачи.		
Проявление творческой активности обучающихся. Наличие продуктов оригинальной, творческой деятельности обучающихся.	Постановка необычных, нестандартных вопросов и проблем. Большое количество выдвигаемых идей, нестандартных решений, замыслов.		
	Использование оригинальных решений для рутинных задач, комбинирование условий.		
Развитие интереса обучающихся к программированию	Устойчивость интереса обучающихся к программированию. Наличие внутренней мотивации у обучающихся для занятий программированием. Удовлетворенность обучающихся собственными достижениями. Успешность в освоении образовательной программы. Активность в образовательном процессе.		
	Применение знаний полученных в процессе обучения в решении смежных задач в процессе обучения и для бытового применения.		
Развитие трудолюбия и целеустремленности	Целеустремленность, трудолюбие, проявляемые при выполнении индивидуальных заданий.		
	Учащийся не бросает попытки решить сложные задачи.		

# ОБРАЗЦЫ ИТОГОВЫХ АТТЕСТАЦИОННЫХ ЗАДАНИЙ

## Задание 1

### ВВОД И ВЫВОД ДАННЫХ

Программа запрашивает у пользователя имя и фамилию, после чего выводит приветственное сообщение.

**На входе:** две строковые переменные

**На выходе:** строка

**Например:**

>>>Ваше имя? Иван

>>>Ваша фамилия? Петров

*Здравствуйте, Петров Иван!*

## Задание 2

### РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ

Программа находит минимальное значение из трёх.

**На входе:** три целых числа.

**На выходе:** одно целое число.

**Пример:**

>>> Введите первое число: 1

>>> Введите второе число: 2

>>> Введите третье число: 6

Минимальное число: 1

## Задание 3

### ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ

Напишите программу, выводящую на экран все числа от А до В включительно.

**Пояснение:**  $A \leq B$ .

**На входе:** в первой строке вводится А, во второй вводится В.

**На выходе:** последовательность целых чисел, разделённых пробелом.

**Например:**

>>>Введите А: -3

>>>Введите В: 5

-3 -2 -1 0 1 2 3 4 5

## МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

**Python** поддерживает множество различных парадигм программирования: объектно-ориентированное, структурное, функциональное и др.

Python – язык с динамической типизацией данных, интерпретируемый язык, язык сценариев (скриптов). Это означает, что программы написанные на python требуют для своего запуска компилятор (интерпритатор) языка. Python обладает простым и лаконичным синтаксисом, а богатая стандартная библиотека является одной из причин широкой популярности языка в совершенно различных сферах деятельности.

Разработка языка Python была начата в конце 1980-х годов сотрудником голландского института CWI Гвидо ван Россумом. Разработка языка идет по двум направлениям (версии языка), не сохраняющих совместимость – это версия 2.7 и 3.x. Подробно об этом на официальном сайте Python. В настоящее время актуальной версией языка является 3.8.

- Официальный сайт: [www.python.org](http://www.python.org)
- Страница загрузки: [www.python.org/download/](http://www.python.org/download/)

Язык программирования **Python 3.x** входит в состав допустимых языков программирования на Всероссийской олимпиаде школьников по информатике. В демонстрационной версии ЕГЭ — Python один из языков на котором представлен программный код.

### *Методические материалы*

- Allen Downey. Думать на языке Python.
- Федоров Д. Ю. Основы программирования на примере языка Python.

### *Задачники по программированию*

- М. Э. Абрамян 1000 задач по программированию. Часть I. Скалярные типы данных, управляющие операторы, процедуры и функции.
- М. Э. Абрамян 1000 задач по программированию. Часть II. Минимумы и максимумы, одномерные и двумерные массивы, символы и строки, двоичные файлы.
- М. Э. Абрамян 1000 задач по программированию. Часть III. Текстовые файлы, составные типы данных в процедурах и функциях, рекурсия, указатели и динамические структуры.
- Златопольский Д. М. Сборник задач по программированию.

Методическая основа программы «Основы программирования на языке Python» предназначен для начального знакомства с основными алгоритмами и с программированием на языке Python в интегрированных средах разработки (IDE) Python и PyScripter. В программе изучаются типовые задачи в рамках базового курса информатики, алгоритмы их решения и реализация этих алгоритмов на языке Python. Дополнительно в курсе описываются основы программирования графики в Python с использованием кросс-платформенной библиотеки Tkinter.

## ЛИТЕРАТУРА ДЛЯ ОБУЧАЮЩИХСЯ

1. Allen Downey. Думать на языке Python. Green Tea Press. 2012. Перевод на русский язык Николай Орехов 2017. [https://bitbucket.org/thinkpython\\_ru/](https://bitbucket.org/thinkpython_ru/) book/src
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
4. Доусен М. Програмируем на Python. - СПб.: Питер, 2014. - 416 с.: ил.
5. Пейн, Брайсон. Python для детей и родителей / Брайсон Пейн. – Москва: Издательство «Э». 2017. – 352 с.: ил.
6. Хайнеман, Джордж, Пояяис, Гэри, Сеяков, Стэнли. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО —Альфа-книга, 2017. — 432 с. : ил. — Парал. тит. англ.
7. Седжвик, Роберт, Уэйн, Кевин, Дондеро, Роберт. Программирование на языке Python: учебный курс. : Пер. с англ. - СПб. : ООО "Альфа-книга": 2017. - 736 с. : ил. - Парал. тит. англ.
8. У. Сэнд, К. Сэнд. Hello World! Занимательное программирование. — СПб.: Питер, 2016. — 400 с.: ил. — (Серия «Вы и ваш ребенок»).

## ЛИТЕРАТУРА ДЛЯ ПЕДАГОГА

1. Бизли Д. М. Язык программирования Python : справочник :пер. с англ. / Д. М. Бизли. – Киев : ДиаСофт, 2000
2. Гифт Н. Python в системном администрировании UNIX и Linux : пер. с англ. / Н. Гифт, Д. Джонс. – СПб. : Символ-Плюс, 2009
3. Лейнингем И. Освой самостоятельно Python за 24 часа : пер. с англ. / И. Лейнингем. – М. : Издательский дом «Вильямс», 2001
4. Лесса А. Python. Руководство разработчика : пер. с англ. / А. Лесса. – СПб. : ДиасофтЮП, 2001
5. Лутц М. Изучаем Python : пер. с англ. / М. Лутц. – СПб. : Символ-Плюс, 2009
6. Лутц М. Программирование на Python : пер.с англ. / М. Лутц. – СПб. : Символ-Плюс, 2002
7. Саммерфельд М. Программирование на Python 3 Подробное руководство : пер. с англ. / М. Саммерфельд. – СПб. : Символ-Плюс, 2009
8. Сузи Р. А. Python / Р. А. Сузи. – СПб. : БХВ-Петербург, 2002
9. Сузи Р. А. Язык Python и его применения : учеб. пособие / Р.А. Сузи. – М. : Интернет-Университет информационных технологий: БИНОМ. Лаборатория знаний, 2006
10. Язык программирования Python / Г. Россум [и др.]. – СПб. : АНО «Институт логики» – Невский диалект, 2001
11. Сэнд У., Сэнд К. Hello World! Занимательное программирование. — СПб.: Питер, 2016



12. Шоу, Зед. Легкий способ выучить Python / Зед Шоу ; [пер. с англ. М. А. Райтмана]. — М: Издательство «Э», 2017.
13. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017.
14. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. – СПб.: БХВ-Петербург, 2016
15. Саммерфилд М. Программирование на Python 3. Подробное руководство. СПб.: Символ-Плюс — 2009
16. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017
17. Пэйн, Брайсон Python для детей и родителей. М.: «Э», 2017
18. Лучано Рамальо Python. К вершинам мастерства / Пер. с англ. Слинкин А. А. — М.: ДМК Пресс, 2016
19. Гифт Н., Джонс Д. Python в системном администрировании UNIX и Linux. СПб.: Символ-Плюс, 2009
20. М.Э. Абрамян. 1000 задач по программированию. Часть I. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
21. М.Э. Абрамян. 1000 задач по программированию. Часть II. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
22. М.Э. Абрамян. 1000 задач по программированию. Часть III. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
23. Златопольский Д.М. Сборник задач по программированию. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 304 с.: ил. – (ИиИКТ)

## КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

<b>Название программы</b> _____ <b>ФИО педагога</b> _____ <b>Учебный год:</b> 2021/2022 <b>Продолжительность реализации программы</b> _____ <b>Количество часов</b> _____ <b>Группа №</b> _____ <b>Расписание занятий</b> _____ <b>Промежуточная аттестация:</b> октябрь, декабрь, февраль <b>Итоговая аттестация:</b> май				
Дата	Тематический блок / Тема	Количество часов		
		теор	прак	всего
<b>1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки</b>		<b>1</b>	<b>1</b>	<b>2</b>
1	Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Области применения.	1		1
2	<i>Дистрибутивы интерпретатора языка Python. Установка официального дистрибутива интерпретатора языка Python 3.x. Запуск интерактивного режима Python. Работа в официальном IDLE Python. Работа в среде PyScripter, запуск/остановка скриптов на языке Python.</i>		1	1
<b>2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов</b>		<b>2</b>	<b>2</b>	<b>4</b>
3	Документация на язык программирования – PEP8. Стиль написания кода: отступы. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Комментарии.	1		1
4	Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.	1		1
5	<i>Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().</i>		1	1
6	<i>Выполнение команд в интерактивном режиме. Создание скриптов. Сохранение скриптов. Программа приветствие.</i>		1	1
<b>3. Встроенные типы данных. Переменные. Основные операции с данными</b>		<b>3</b>	<b>5</b>	<b>8</b>

7	Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. <i>Основные операции с числовыми данными.</i>	1	1	2
8	<i>Базовые функции языка Python для работы с числовыми данными. Ввод/вывод числовых данных. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.</i>		1	1
9	Основы строкового типа данных. <i>Склеивание строк.</i>	1	1	2
10	Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций.	1		1
11	<i>Использование логических операций. Взаимодействие с числовыми и строковыми типами данных. Битовые операции.</i>		2	2
<b>4. Ввод/вывод данных в программу.</b>		<b>2</b>	<b>2</b>	<b>4</b>
12	Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). <i>Использование функции input(). Преобразование типов вводимых данных.</i>	1	1	2
13	Функция print() для вывода данных на экран и в файл. <i>Использование функции print(). Формирование строки. Вывод псевдографики.</i>	1	1	2
<b>5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор.</b>		<b>2</b>	<b>2</b>	<b>4</b>
14	Понятие алгоритма их типы. Описание задачи в виде алгоритма.	1		1
15	Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. <i>Применение условного оператора if...else для решения прикладных задач.</i>	1	1	2
16	<i>Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.</i>		1	1
<b>6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.</b>		<b>2</b>	<b>2</b>	<b>4</b>

17	Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы.	2		2
18	<i>Разработка алгоритма программ с учетом возможных ошибок выполнения. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам.</i>		1	1
19	<i>Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.</i>		1	1
<b>7. Управляющие инструкции – циклы.</b>		<b>3</b>	<b>7</b>	<b>10</b>
20	Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций.	1		1
21	Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. <i>Решение задач с помощью цикла for.</i>	1	2	3
22	Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. <i>Бесконечный цикл. Формулировка условия. Решение задач с использованием циклов while.</i>	1	2	3
23	<i>Решение задач различного типа с помощью циклических алгоритмов.</i>		3	3
<b>8. Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.</b>		<b>4</b>	<b>6</b>	<b>10</b>
24	Понятие функция. Назначение. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям.	2		2
25	Математические функции. Подключение модулей. Модуль Math. <i>Использование функций модуля Math для решения математических задач.</i>	1	1	2
26	Создание собственных функций: определение и использование. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. <i>Использование функций в программах. Создание собственных модулей.</i>	1	1	2
27	<i>Создание игры «Волк, коза и капуста» с использованием циклического алгоритма.</i>		2	2
28	<i>Знакомство с модулем NumPy и matplotlib.</i>		1	1

29	<i>Знакомство с модулем tkinter.</i>		1	1
<b>9. Строки. Списки. Словари. Кортежи.</b>		<b>6</b>	<b>6</b>	<b>12</b>
30	Строка как последовательность. Функция len. Перебор элементов строки. Срезы строк. Поиск. Подсчет. <i>Методы строк. Сравнение строк. Работа с текстовыми файлами.</i>	1	1	2
31	Списки. Перебор элементов списка. Операции со списками. Срезы списков. <i>Методы списков. Отображение, фильтрация и сокращение. Удаление элементов списка. Создание синонимов в списке.</i>	1	1	2
32	Словари. Словари как счетчики. Циклы и словари. Обратный поиск. <i>Словари и списки. Глобальные переменные.</i>	2	2	4
33	Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. <i>Кортежи с переменным числом аргументов. Списки и кортежи. Словари и кортежи. Сравнение кортежей. Последовательности последовательностей.</i>	2	2	4
<b>10. Файловый ввод/вывод.</b>		<b>2</b>	<b>2</b>	<b>4</b>
34	Понятие файл. Имена файлов. Место положение файлов. <i>Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл.</i>	1	1	2
35	Базы данных. <i>Сериализация. Десериализация. Конвейеры.</i>	1	1	2
<b>11. Основы объектно-ориентированного программирования.</b>		<b>3</b>	<b>3</b>	<b>6</b>
36	Основы объектно-ориентированного подхода. Объекты в реальном мире, объекты в Python. Объект = атрибуты + методы. <i>Создание объектов. Примеры класса.</i>	1	1	2
37	Скрываем данные. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения. <i>Использование классов в программах.</i>	2	2	4
<b>12. Итоговая аттестация.</b>			<b>4</b>	<b>4</b>
<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>			4	4
<b>ИТОГО</b>		<b>30</b>	<b>42</b>	<b>72</b>