

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ ГОРОДА ВОЛОГДЫ
МУНИЦИПАЛЬНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«ДЕТСКО-ЮНОШЕСКИЙ ЦЕНТР «ЕДИНСТВО»

Рассмотрено на педагогическом совете
МУ ДО «ДЮЦ «Единство»
Протокол №4 от 31 мая 2024 г.

УТВЕРЖДАЮ
Директор МУ ДО «ДЮЦ «Единство»

Приказ №43 от 31 мая 2024 г.

И.Н. Курина



**Дополнительная общеобразовательная общеразвивающая
программа технической направленности**

УЛУБЛЕННЫЙ РУТНОН

Уровень программы: углубленный

Возраст детей: 13–16 лет
Срок реализации: 9 месяцев

Автор-составитель:
Меньшиков Андрей Иванович,
педагог дополнительного образования
МУ ДО ДЮЦ «Единство»

Вологда
2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Углубленный Python» имеет *техническую направленность*.

Программа разработана в соответствии со следующими нормативными документами:

- Федеральный Закон РФ от 29.12.2012 г. № 273 «Об образовании в Российской Федерации» (с последующими изменениями);
- Указ Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года» от 07 мая 2024 года № 309;
- Указ Президента Российской Федерации от 01.12.2016 г. № 642 «О Стратегии научно-технического развития Российской Федерации»;
- Указ Президента Российской Федерации от 21.07.2020 г. № 474 «О национальных целях развития Российской Федерации на период до 2030 года»;
- Указ Президента Российской Федерации от 9 ноября 2022 г. № 809 «Об утверждении Основ государственной политики по сохранению и укреплению традиционных российских духовно-нравственных ценностей»;
- Постановление Правительства Российской Федерации от 26.12.2017 г. № 1642 «Об утверждении государственной программы Российской Федерации «Развитие образования»;
- Распоряжение Правительства Российской Федерации от 31.03.2022 г. № 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года»;
- Федеральный проект «Успех каждого ребенка», утвержденный президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам (протокол от 24 декабря 2018 года № 16);
- Паспорт Национального проекта «Образование»;
- Приказ Министерства образования и науки Российской Федерации от 23.08.2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
- Приказ Минпросвещения России от 27.07.2022 N 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмо Министерства просвещения Российской Федерации от 19.03.2020 г. № ГД-39/04 «О направлении методических рекомендаций» («Методические рекомендации по реализации образовательных программ начального общего, основного общего, среднего общего образования, образовательных программ среднего профессионального образования, дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий»);
- Письмо Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242 «О направлении информации» («Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);
- Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (утверждены Постановлением Главного государственного санитарного врача РФ от 28.09.2020 г. №28).

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.

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

Цель программы

Личностное и интеллектуальное развитие обучающихся в процессе обучения программированию 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.
- Овладеть приёмами самостоятельной и творческой деятельности при разработке собственных приложений.

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

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 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 углубленный.

Адресат программы

Программа предназначена для обучающихся 13-16 лет, уже освоивших

базовые конструкции языка Python.

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 72 часа в год: на практические занятия – 46 часов, теоретические – 26 часов.

СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНЫЙ ПЛАН

№ п/п	Тематический блок	Количество часов			Форма контроля итогов
		теория	пр-ка	всего	
1.	Введение. Техника безопасности	2	0	2	
2.	Повторение основных конструкций языка Python. О типах данных в Python	2	2	4	программный продукт
3.	Вложенные списки	2	6	8	программный продукт
4.	Множества	2	4	6	программный продукт
5.	Словари	2	4	6	программный продукт
6.	Парадигмы программирования, введение в функциональное программирование. Функции как объекты	4	6	10	программный продукт
7.	Введение в рекурсию	1	3	4	программный продукт
8.	Файловый ввод/вывод. Работа с текстовыми файлами	2	2	4	программный продукт
9.	Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python	2	2	4	программный продукт
10.	Встроенные модули: random, decimal и fractions. Модуль turtle	4	6	10	программный продукт
11.	Работа с датой и временем. Модули date и time	3	5	8	программный продукт
12.	Работа над проектом	0	6	6	программный продукт
	ИТОГО	26	46	72	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
1. Введение. Техника безопасности		2	0	2
1	Как устроен курс. Техника безопасности	1	0	1
2	Входная диагностика	1	0	1
2. Повторение основных конструкций языка Python. О типах данных в Python		2	2	4
3	Повторение основных конструкций языка Python. Типы алгоритмов и типы данных	1	0	1
4	<i>Решение задач на условия, циклы, строки, списки, функции</i>	0	1	1
5	Логический тип данных bool. Булевы значения как числа. Функция bool(). Функции-предикаты. Функция isinstance(). Тип данных NoneType. Понятие «пустого значения». Литерал None. Проверка на None. Сравнение None с другими типами данных	1	0	1
6	<i>Таблицы истинности. Предикат делимости. Задачи на сравнения с None</i>	0	1	1
3. Вложенные списки		2	6	8
7	Глубина вложенности. Двумерные списки	1	0	1
8	Индексация двумерных списков. <i>Создание вложенных списков. Считывание вложенных списков</i>	1	1	2
9	<i>Перебор и вывод элементов вложенного списка</i>	0	2	2
10	<i>Обработка вложенных списков</i>	0	3	3
4. Множества		2	4	6
11	Множества в математике: обозначения множеств, конечные и бесконечные множества, равенство множеств, подмножество и надмножество, пустое множество. Диаграммы Эйлера-Венна	1	0	1
12	<i>Числовые множества: натуральных чисел, целых чисел, рациональных чисел, иррациональных чисел, вещественных чисел. Операции над множествами</i>	0	1	1
13	Множества в Python. Создание множества. Встроенная функция set()	1	0	1
14	<i>Перебор элементов множества. Сравнение множеств</i>	0	1	1

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
15	<i>Методы множеств: добавление элементов, удаление элемента.</i>	0	1	1
16	<i>Методы множеств: операции над множествами. Методы множеств, изменяющие текущие множества. Подмножества и надмножества</i>	0	1	1
5. Словари		2	4	6
17	Словари в Python. Создание словаря. Обращение к элементу словаря. Функция dict(). Особенности словарей	1	0	1
18	<i>Вывод словаря. Сравнение словарей</i>	0	1	1
19	<i>Перебор элементов словаря. Методы keys(), values(), items()</i>	0	1	1
20	<i>Распаковка ключей словаря. Сортировка словаря</i>	0	1	1
21	Вложенные словари, генераторы словарей. Итерация по вложенным словарям	1	0	1
22	<i>Методы словарей. Добавление и изменение элементов в словаре: метод get(), метод update(), метод setdefault(). Удаление элементов из словаря: оператор del, метод pop(), метод popitem(), метод clear()</i>	0	1	1
6. Парадигмы программирования, введение в функциональное программирование. Функции как объекты		4	6	10
23	Парадигмы программирования: императивное, структурное, объектно-ориентированное, логическое и функциональное программирование	1	0	1
24	<i>Необязательные и именованные аргументы. Позиционные аргументы</i>	0	1	1
25	Функции с переменным количеством аргументов	1	0	1
26	Функции как объекты. Функции в качестве аргументов других функций	1	0	1
27	<i>Функции высшего порядка. Функции map(), filter(), reduce()</i>	0	1	1
28	<i>Модуль operator</i>	0	1	1

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
29	Анонимные функции. Определение. Однократное использование функции. Передача анонимных функций в качестве аргументов другим функциям. Возвращение функции в качестве результата другой функции. Ограничения анонимных функций	1	0	1
30	<i>Условный оператор в теле анонимной функции</i>	0	1	1
31	<i>Передача аргументов в анонимную функцию</i>	0	1	1
32	<i>Встроенные функции <code>any()</code>, <code>all()</code>, <code>zip()</code>, <code>enumerate()</code></i>	0	1	1
7. Введение в рекурсию		1	3	4
33	Введение в рекурсию. Хвостовая рекурсия	1	0	1
34	<i>Решение задач на рекурсию</i>	0	1	1
35	<i>Решение задач на рекурсию. Мемоизация</i>	0	1	1
36	<i>Рекурсивный обход коллекций. Настройка глубины рекурсии в Python</i>	0	1	1
8. Файловый ввод/вывод. Работа с текстовыми файлами		2	2	4
37	Понятие файл. Имена файлов. Местоположение файлов. Относительные и абсолютные пути. Типы файлов	1	0	1
38	<i>Открытие и закрытие файлов. Чтение данных из файла</i>	0	1	1
39	Методы доступа к файлам. Кодировка файлов	1	0	1
40	<i>Позиция (курсор) в файле. Менеджеры контекста. Запись данных в файл</i>	0	1	1
9. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python		2	2	4
41	Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы	1	0	1
42	<i>Разработка алгоритма программ с учетом возможных ошибок выполнения. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам</i>	0	1	1

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
43	<i>Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток</i>	0	1	1
44	<i>Пользовательские исключения. Методики LBYL и EAF. Оператор assert</i>	0	1	1
10. Встроенные модули: random, decimal и fractions. Модуль turtle		4	6	10
45	Случайные числа. Псевдослучайные числа	1	0	1
46	<i>Модуль random. Методы shuffle(), choice(), sample(). Модуль string</i>	0	1	1
47	Ещё раз о числовых типы данных. Числа с плавающей точкой float.	1	0	1
48	<i>Модуль decimal. Тип данных Decimal</i>	0	1	1
49	Модуль fractions. Рациональное число. Тип данных Fraction	1	0	1
50	<i>Арифметические операции над Fraction числами. Свойства numerator и denominator. Метод limit_denominator()</i>	0	1	1
51	Черепашья графика. Модуль turtle	1	0	1
52	<i>Рисование отрезков прямой. Поворот черепашки. Установка углового направления черепашки. Изменение внешнего вида черепашки</i>	0	1	1
53	<i>Поднятие и опускание пера. Рисование кругов и точек. Изменение размера пера. Изменение цвета рисунка. Изменение цвета фона. Создание штампа. Возвращение экрана в исходное состояние. Получение текущей позиции черепашки. Управление скоростью анимации черепашки</i>	0	1	1
54	<i>Вывод текста в графическое окно. Заполнение геометрических фигур. Создание нескольких черепашек. Отслеживание нажатия клавиши. Отслеживание нажатия мыши</i>	0	1	1
11. Работа с датой и временем. Модули date и time		3	5	8
55	Модуль datetime. Типы данных модуля datetime	1	0	1
56	<i>Тип данных date. Тип данных time</i>	0	1	1
57	<i>Сравнение дат и времени. Функции str() и repr()</i>	0	1	1
58	Форматирование даты и времени	1	0	1

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
59	<i>Использование локализации. Преобразование строки в дату и время</i>	0	1	1
60	<i>Тип данных timedelta. Атрибуты days, seconds, microseconds. Метод total_seconds(). Сравнение временных интервалов. Операции над временными интервалами timedelta</i>	0	1	1
61	Модуль calendar. Атрибуты day_name, day_abbr, month_name, month_abbr	1	0	1
62	<i>Функции setfirstweekday(), firstweekday(). Функции isleap(), leapdays(). Функции weekday(), monthrange(), monthcalendar(). Функции month(), calendar(), prmonth(), prcal()</i>	0	1	1
12. Работа над проектом		0	6	6
<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>		0	6	6
ИТОГО		26	46	72

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

1. Введение. Техника безопасности

Теория:

Как устроен курс. Техника безопасности.

Практические занятия:

1. Входная диагностика: решение математических и логических задач с использованием IDLE

2. Повторение основных конструкций языка Python. О типах данных в Python

Теория:

Повторение основных конструкций языка Python. Типы алгоритмов и типы данных. Логический тип данных bool. Булевы значения как числа. Функция bool(). Функции-предикаты. Функция isinstance(). Тип данных NoneType. Понятие «пустого значения». Литерал None. Проверка на None. Сравнение None с другими типами данных.

Практические занятия:

1. Решение задач на условия, циклы, строки, списки, функции
2. Таблицы истинности.
3. Предикат делимости.
4. Задачи на сравнения с None.

3. Вложенные списки

Теория:

Глубина вложенности. Двумерные списки. Индексация двумерных списков

Практические занятия:

1. Создание вложенных списков.
2. Считывание вложенных списков
3. Перебор и вывод элементов вложенного списка
4. Обработка вложенных списков

4. Множества

Теория:

Множества в математике: обозначения множеств, конечные и бесконечные множества, равенство множеств, подмножество и надмножество, пустое множество. Диаграммы Эйлера-Венна.

Множества в Python. Создание множества. Встроенная функция set()

Практические занятия:

1. Числовые множества: натуральных чисел, целых чисел, рациональных чисел, иррациональных чисел, вещественных чисел. Операции над множествами.
2. Перебор элементов множества. Сравнение множеств.

3. Методы множеств: добавление элементов, удаление элемента.
4. Методы множеств: операции над множествами.
5. Методы множеств, изменяющие текущие множества.
6. Подмножества и надмножества.

5. Словари

Теория:

Словари в Python. Создание словаря. Обращение к элементу словаря. Функция dict(). Особенности словарей. Вложенные словари, генераторы словарей. Итерация по вложенным словарям.

Практические занятия:

1. Вывод словаря. Сравнение словарей.
2. Перебор элементов словаря. Методы keys(), values(), items().
3. Распаковка ключей словаря. Сортировка словаря.
4. Методы словарей. Добавление и изменение элементов в словаре: метод get(), метод update(), метод setdefault().
5. Удаление элементов из словаря: оператор del, метод pop(), метод popitem(), метод clear().

6. Парадигмы программирования, введение в функциональное программирование. Функции как объекты.

Теория:

Парадигмы программирования: императивное, структурное, объектно-ориентированное, логическое и функциональное программирование. Функции с переменным количеством аргументов. Функции как объекты. Функции в качестве аргументов других функций. Анонимные функции. Определение. Однократное использование функции. Передача анонимных функций в качестве аргументов другим функциям. Возвращение функции в качестве результата другой функции. Ограничения анонимных функций.

Практические занятия:

1. Необязательные и именованные аргументы. Позиционные аргументы.
2. Функции высшего порядка. Функции map(), filter(), reduce().
3. Модуль operator.
4. Условный оператор в теле анонимной функции.
5. Передача аргументов в анонимную функцию.
6. Встроенные функции any(), all(), zip(), enumerate(). Функция all()

7. Введение в рекурсию

Теория:

Понятие рекурсии. Хвостовая рекурсия

Практические занятия:

1. Решение задач на рекурсию.
2. Мемоизация.
3. Рекурсивный обход коллекций. Настройка глубины рекурсии в Python

8. Файловый ввод/вывод. Работа с текстовыми файлами

Теория:

Понятие файл. Имена файлов. Местоположение файлов. Относительные и абсолютные пути. Типы файлов. Методы доступа к файлам. Кодировка файлов.

Практические занятия:

1. Открытие и закрытие файлов.
2. Чтение данных из файла.
3. Позиция (курсор) в файле.
4. Менеджеры контекста.
5. Запись данных в файл.

9. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python

Теория:

Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы.

Практические занятия:

1. Разработка алгоритма программ с учетом возможных ошибок выполнения.
2. Сообщения интерпретатора об ошибках.
3. Поиск документации по ошибкам.
4. Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.
5. Пользовательские исключения.
6. Методики LBYL и EAF.
7. Оператор assert.

10. Встроенные модули: random, decimal и fractions. Модуль turtle

Теория:

Случайные числа. Псевдослучайные числа. Ещё раз о числовых типах данных. Числа с плавающей точкой float. Модуль fractions. Рациональное число. Тип данных Fraction. Черепашня графика. Модуль turtle.

Практические занятия:

1. Модуль random.
2. Методы shuffle(), choice(), sample().
3. Модуль string.

4. Модуль decimal. Тип данных Decimal.
5. Арифметические операции над Fraction числами.
6. Свойства numerator и denominator.
7. Метод limit_denominator().
8. Рисование отрезков прямой.
9. Поворот черепашки. Установка углового направления черепашки.
10. Изменение внешнего вида черепашки.
11. Поднятие и опускание пера. Рисование кругов и точек.
12. Изменение размера пера. Изменение цвета рисунка. Изменение цвета фона.
13. Создание штампа. Возвращение экрана в исходное состояние.
14. Получение текущей позиции черепашки.
15. Управление скоростью анимации черепашки.
16. Вывод текста в графическое окно.
17. Заполнение геометрических фигур.
18. Создание нескольких черепашек.
19. Отслеживание нажатия клавиш. Отслеживание нажатия мыши.

11. Работа с датой и временем. Модули date и time

Теория:

Модуль datetime. Типы данных модуля datetime. Форматирование даты и времени. Модуль calendar. Атрибуты day_name, day_abbr, month_name, month_abbr.

Практические занятия:

1. Тип данных date.
2. Тип данных time.
3. Сравнение дат и времени.
4. Функции str() и repr().
5. Использование локализации. Преобразование строки в дату и время
6. Тип данных timedelta.
7. Атрибуты days, seconds, microseconds.
8. Метод total_seconds().
9. Сравнение временных интервалов.
10. Операции над временными интервалами timedelta
11. Функции setfirstweekday(), firstweekday().
12. Функции isleap(), leapdays().
13. Функции weekday(), monthrange(), monthcalendar().
14. Функции month(), calendar(), prmonth(), prcal().

12. Работа над проектом

Практические занятия:

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

ПЛАНИРУЕМЫЕ 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), ноутбуки, 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 КОНТРОЛЯ

Программа предполагает обязательный промежуточный и итоговый контроль по всем темам программы обучения, для осуществления мониторинга освоения образовательной программы. Помимо итогового контроля проводятся периодические проверки знаний в форме индивидуальных заданий после изучения одной или нескольких тем, с целью выявить возможные проблемы с усвоением материала курса.

Итоговый контроль предполагает дифференциальный зачет в виде презентации программы. Обучающимся предлагается небольшое индивидуальное задание для создания программы, предполагается игра с графическим интерфейсом. По результатам обучающийся 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

«Углубленный 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.	Постановка необычных, нестандартных вопросов и проблем. Большое количество выдвигаемых идей, нестандартных решений, замыслов.	Анализ и оценивание детских творческих достижений.
		Использование оригинальных решений для рутинных задач, комбинирование условий.	
Развитие интереса обучающихся к выбранному ими профилю деятельности	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.	Беседы с учащимися. Беседы на выявление мотивов обучающихся к занятию по данному профилю, беседы на выявление профессиональных намерений. Статистический учет сохранности контингента. Анализ мотивов ухода обучающихся из коллектива.
		Применение знаний полученных в процессе обучения в решении смежных задач в процессе обучения и для бытового применения.	
Развитие трудолюбия и целеустремленности	Развитие трудолюбия и целеустремленности	Целеустремленность, трудолюбие, проявляемые при выполнении индивидуальных заданий.	Наблюдение.
		Учащийся не бросает попытки решить сложные задачи.	

**Личная карта обучающегося по дополнительной общеразвивающей
программе**

«Углубленный 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	Степень самостоятельности (участие педагога) Качество усвоения.		

Параметры	Критерии	1-е полугодие	2-е полугодие
дополнительный материал по пройденным темам.	Самостоятельный поиск решений поставленной задачи. Понимание чужих решений и способность своей интерпретации.		
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 контрольных заданий

Задание 1

Вложенные списки

На вход программе подается число n . Напишите программу, которая создает и выводит построчно список, состоящий из n списков `[[1, 2, ..., n], [1, 2, ..., n], ..., [1, 2, ..., n]]`.

На входе: на вход программе подается натуральное число n .

На выходе: программа должна вывести построчно указанный список

Например:

```
>>> 3
>>> [1, 2, 3]
      [1, 2, 3]
      [1, 2, 3]
```

Задание 2

Множества

На вход программе подаются две строки, состоящие из цифр. Необходимо определить, верно ли, что для записи этих строк были использованы одинаковые наборы цифр?

На входе: на вход подаются две строки, состоящие из цифр.

На выходе: программа должна вывести YES, если для записи этих строк были использованы одинаковые наборы цифр и NO, в противном случае.

Пример:

```
>>> 0943
      9304
>>> YES
```

Задание 3

Словари

На вход программе подается список стран и городов каждой страны. Затем даны названия городов. Напишите программу, которая для каждого города выводит, в какой стране он находится.

На входе: Программа получает на вход количество стран n . Далее идет n строк, каждая строка начинается с названия страны, затем идут названия городов этой страны. В следующей строке записано число m , далее идут m запросов — названия каких-то m городов, из перечисленных выше.

На выходе: Программа должна вывести название страны, в которой находится данный город в соответствии с примером.

Например:

>>> 2

Германия Берлин Мюнхен Гамбург Дортмунд
Нидерланды Амстердам Гаага Роттердам Алкмар

4

Амстердам

Алкмар

Гамбург

Гаага

>>> Нидерланды

Нидерланды

Германия

Нидерланды

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 3.12.

- Официальный сайт: www.python.org
- Страница загрузки: 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.
- Доусен М. Програмируем на 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 «Углубленный Python» предназначена для продолжения знакомства с основными алгоритмами и с программированием на языке Python в интегрированных средах разработки (IDE) Python и PyCharm. В программе изучаются типовые задачи в рамках базового курса информатики, алгоритмы их решения и реализация этих алгоритмов на языке Python.

ВОСПИТАТЕЛЬНЫЙ КОМПОНЕНТ

Особенностью программы является компонентность образовательно-воспитательного процесса, взаимосвязь между ними:

- I компонент – система дополнительного образования. Реализация дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python». Целью первого компонента является формирование образовательного пространства и реализация в рамках образовательной программы дополнительного образования детей, в первую очередь, задач воспитания. При реализации программы взрослые выступают в роли педагогов дополнительного образования, тренеров, наставников, педагогов – психологов, мастеров, а дети и подростки – в роли обучающихся, наставников (в системе «ребенок – ребенок»). В зависимости от темы, формы организации занятий строится адекватная система отношений, определяются нормы поведения в образовательном пространстве: ученичество, сотворчество и т.п.
- II компонент – система воспитательных мероприятий. Предназначение второго компонента – обеспечение создания воспитательного пространства, в котором реализуются проекты, мероприятия и акции по основным направлениям воспитательной деятельности с использованием разнообразных форм организации.
- III компонент – психолого-педагогическая поддержка и сопровождение обучающихся.

КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ

Название мероприятия, события	Форма проведения: <i>беседа, конкурс, игра, круглый стол, акция и др.</i>	Сроки
Неделя технического творчества	беседа на тему «Компьютерная грамотность»	ноябрь-декабрь
Международный день программиста	беседа о роли, особенностях и традициях	7 января, 13 сентября
День космонавтики	подготовка проектов на космическую тему	12 апреля
Научно-практическая конференция «Мир науки +»	конференция (подготовка докладов в соответствии с Положением)	март-апрель

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/ 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

Название программы _____ ФИО педагога _____ Учебный год: _____ Продолжительность реализации программы _____ Количество часов _____ Расписание занятий _____ Праздничные дни: 1, 2, 3, 4, 5, 6, 7, 8 января, 23 февраля, 8 марта, 1, 9 мая, 4 ноября; 31 декабря, Промежуточный контроль: декабрь Итоговый контроль: май			
Дата	П/п	Тематический блок / Тема	Часы
			всего
1. Введение. Техника безопасности			2
	1	Как устроен курс. Техника безопасности	1
	2	Входная диагностика	1
2. Повторение основных конструкций языка Python. Еще раз о типах данных в Python			4
	3	Повторение основных конструкций языка Python. Типы алгоритмов и типы данных	1
	4	<i>Решение задач на условия, циклы, строки, списки, функции</i>	1
	5	Логический тип данных bool. Булевы значения как числа. Функция bool(). Функции-предикаты. Функция isinstance(). Тип данных NoneType. Понятие «пустого значения». Литерал None. Проверка на None. Сравнение None с другими типами данных	1
	6	<i>Таблицы истинности. Предикат делимости. Задачи на сравнения с None</i>	1
3. Вложенные списки			8
	7	Двумерные списки. Глубина вложенности.	1
	8	Индексация двумерных списков	1
	9	<i>Создание вложенных списков. Считывание вложенных списков</i>	1
	10 11	<i>Перебор и вывод элементов вложенного списка</i>	2

	12 13 14	<i>Обработка вложенных списков</i>	3
4. Множества			6
	15	Множества в математике: обозначения множеств, конечные и бесконечные множества, равенство множеств, подмножество и надмножество, пустое множество. Диаграммы Эйлера-Венна	1
	16	<i>Числовые множества: натуральных чисел, целых чисел, рациональных чисел, иррациональных чисел, вещественных чисел. Операции над множествами</i>	1
	17	Множества в Python. Создание множества. Встроенная функция set()	1
	18	<i>Перебор элементов множества. Сравнение множеств</i>	1
	19	<i>Методы множеств: добавление элементов, удаление элемента.</i>	1
	20	<i>Методы множеств: операции над множествами. Методы множеств, изменяющие текущие множества. Подмножества и надмножества</i>	1
5. Словари			6
	21	Словари в Python. Создание словаря. Обращение к элементу словаря. Функция dict(). Особенности словарей	1
	22	<i>Вывод словаря. Сравнение словарей</i>	1
	23	<i>Перебор элементов словаря. Методы keys(), values(), items()</i>	1
	24	<i>Распаковка ключей словаря. Сортировка словаря</i>	1
	25	Вложенные словари, генераторы словарей. Итерация по вложенным словарям	1
	26	<i>Методы словарей. Добавление и изменение элементов в словаре: метод get(), метод update(), метод setdefault(). Удаление элементов из словаря: оператор del, метод pop(), метод popitem(), метод clear()</i>	1
6. Парадигмы программирования, введение в функциональное программирование. Функции как объекты			10
	27	Парадигмы программирования: императивное, структурное, объектно-ориентированное, логическое и функциональное программирование	1
	28	<i>Необязательные и именованные аргументы. Позиционные аргументы</i>	1
	29	Функции с переменным количеством аргументов	1

	30	Функции как объекты. Функции в качестве аргументов других функций	1
	31	<i>Функции высшего порядка. Функции <code>map()</code>, <code>filter()</code>, <code>reduce()</code></i>	1
	32	<i>Модуль <code>operator</code></i>	1
	33	Анонимные функции. Определение. Однократное использование функции. Передача анонимных функций в качестве аргументов другим функциям. Возвращение функции в качестве результата другой функции. Ограничения анонимных функций	1
	34	<i>Условный оператор в теле анонимной функции</i>	1
	35	<i>Передача аргументов в анонимную функцию</i>	1
	36	<i>Встроенные функции <code>any()</code>, <code>all()</code>, <code>zip()</code>, <code>enumerate()</code>. Функция <code>all()</code></i>	1
7. Введение в рекурсию			4
	37	Введение в рекурсию. Хвостовая рекурсия	1
	38	<i>Решение задач на рекурсию</i>	1
	39	<i>Решение задач на рекурсию. Мемоизация</i>	1
	40	<i>Рекурсивный обход коллекций. Настройка глубины рекурсии в Python</i>	1
8. Файловый ввод/вывод. Работа с текстовыми файлами			4
	41	Понятие файл. Имена файлов. Местоположение файлов. Относительные и абсолютные пути. Типы файлов	1
	42	<i>Открытие и закрытие файлов. Чтение данных из файла</i>	1
	43	Методы доступа к файлам. Кодировка файлов	1
	44	<i>Позиция (курсор) в файле. Менеджеры контекста. Запись данных в файл</i>	1
9. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python			4
	45	Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы	1
	46	<i>Разработка алгоритма программ с учетом возможных ошибок выполнения. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам</i>	1

	47	<i>Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток</i>	1
	48	<i>Пользовательские исключения. Методики LBYL и EAF. Оператор assert</i>	1
10. Встроенные модули: random, decimal и fractions. Модуль turtle			10
	49	Случайные числа. Псевдослучайные числа	1
	50	<i>Модуль random. Методы shuffle(), choice(), sample(). Модуль string</i>	1
	51	Ещё раз о числовых типах данных. Числа с плавающей точкой float.	1
	52	<i>Модуль decimal. Тип данных Decimal</i>	1
	53	Модуль fractions. Рациональное число. Тип данных Fraction	1
	54	<i>Арифметические операции над Fraction числами. Свойства numerator и denominator. Метод limit_denominator()</i>	1
	55	Черепашья графика. Модуль turtle	1
	56	<i>Рисование отрезков прямой. Поворот черепашки. Установка углового направления черепашки. Изменение внешнего вида черепашки</i>	1
	57	<i>Поднятие и опускание пера. Рисование кругов и точек. Изменение размера пера. Изменение цвета рисунка. Изменение цвета фона. Создание штампа. Возвращение экрана в исходное состояние. Получение текущей позиции черепашки. Управление скоростью анимации черепашки</i>	1
	58	<i>Вывод текста в графическое окно. Заполнение геометрических фигур. Создание нескольких черепашек. Отслеживание нажатия клавиши. Отслеживание нажатия мыши</i>	1
11. Работа с датой и временем. Модули date и time			8
	59	Модуль datetime. Типы данных модуля datetime	1
	60	<i>Тип данных date. Тип данных time</i>	1
	61	<i>Сравнение дат и времени. Функции str() и repr()</i>	1
	62	Форматирование даты и времени	1
	63	<i>Использование локализации. Преобразование строки в дату и время</i>	1
	64	<i>Тип данных timedelta. Атрибуты days, seconds, microseconds. Метод total_seconds(). Сравнение временных интервалов. Операции над временными интервалами timedelta</i>	1

	65	Модуль calendar. Атрибуты day_name, day_abbr, month_name, month_abbr	1
	66	<i>Функции setfirstweekday(), firstweekday(). Функции isleap(), leapdays(). Функции weekday(), monthrange(), monthcalendar(). Функции month(), calendar(), prmonth(), prcal()</i>	1
12. Работа над проектом			6
		<i>Индивидуальное задание. Разработка концепции программы по заданию.</i>	2
		<i>Написание и отладка программы.</i>	2
		<i>Презентация программного продукта.</i>	2
		ИТОГО	72