

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**

# ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON

Уровень программы: базовый

Возраст детей: 12–14 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 «Основы программирования на языке 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.

«Основы программирования на языке Python» рассматривается как базовый курс, развивающий знания школьников в области владения вычислительной техникой и информационно-коммуникационными технологиями, 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 обучающимися.
- 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 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 12–14 лет, в том числе без предыдущего опыта программирования.

**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 – 40 часов, теоретические – 32 часа.

## СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНЫЙ ПЛАН

№ п/п	Тематический блок	Количество часов			Форма контроля итогов
		теория	пр-ка	всего	
1.	Введение. Что такое язык программирования. Классификация языков программирования. Компиляция, интерпретация. Среда разработки. Техника безопасности	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 числами	4	4	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	4	4	8	программный продукт
6.	Управляющие инструкции – циклы	3	8	11	программный продукт
7.	Строковый тип данных	7	7	14	программный продукт
8.	Списки. Кортежи	5	5	10	программный продукт
9.	Функции. Создание собственных функций	4	5	9	программный продукт
10.	Проекты	0	2	2	Итоговый проект
	<b>ИТОГО</b>	<b>32</b>	<b>40</b>	<b>72</b>	

## УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
<b>1. Введение. Что такое язык программирования. Классификация языков программирования. Компиляция, интерпретация. Среда разработки. Техника безопасности</b>		<b>1</b>	<b>1</b>	<b>2</b>
1	Введение. Что такое язык программирования. Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Области применения. Техника безопасности.	1	0	1
2	<i>Дистрибутивы интерпретатора языка Python. Установка официального дистрибутива интерпретатора языка Python 3.x. Запуск интерактивного режима Python. Работа в официальном IDLE Python.</i>	0	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	0	1
4	Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.	1	0	1
5	<i>Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().</i>	0	1	1
6	<i>Выполнение команд в интерактивном режиме. Создание скриптов. Сохранение скриптов. Программа приветствие.</i>	0	1	1
<b>3. Встроенные типы данных. Переменные. Основные операции с числами</b>		<b>4</b>	<b>4</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	Базовые функции языка Python для работы с числовыми данными. <i>Ввод/вывод числовых данных. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.</i>	1	1	2

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
9	Дополнительные операции с целыми числами (возведение в степень, остаток от деления, целочисленное деление). <i>Обработка цифр числа.</i>	1	1	2
10	Модуль math. Особенности подключения модулей. <i>Решение задач.</i>	1	1	2
<b>4. Ввод/вывод данных в программу</b>		<b>2</b>	<b>2</b>	<b>4</b>
11	Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Основы строкового типа данных. <i>Использование функции input(). Преобразование типов вводимых данных.</i>	1	1	2
12	Функция print() для вывода данных на экран. Необязательные параметры команды print (sep и end). <i>Использование функции print(). Склеивание строк. Формирование строки.</i>	1	1	2
<b>5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор</b>		<b>4</b>	<b>4</b>	<b>8</b>
13	Понятие алгоритма их типы. Описание задачи в виде алгоритма. <i>Создание линейного алгоритма</i>	1	1	2
14	Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. <i>Применение условного оператора if...else для решения прикладных задач.</i>	1	1	2
15	Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций. <i>Использование логических операций. Взаимодействие с числовыми и строковыми типами данных</i>	1	1	2
16	Вложенные условия. Каскадные условия. Ленивая оценка условий. Формулировка условий. <i>Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.</i>	1	1	2
<b>6. Управляющие инструкции – циклы</b>		<b>3</b>	<b>8</b>	<b>11</b>
17	Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций	1	0	1
18	Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. <i>Решение задач с помощью цикла for.</i>	1	2	3

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
19	<i>Частые сценарии: подсчет количества, вычисление суммы и произведения, обмен значений переменных, сигнальные метки, нахождение максимума и минимума</i>	0	4	4
20	Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Ключевое слово else в циклах. Вложенные циклы. Бесконечный цикл. Формулировка условия. <i>Решение задач с использованием циклов while.</i>	1	2	3
<b>7. Строковый тип данных</b>		<b>7</b>	<b>7</b>	<b>14</b>
21	Строка как последовательность. Индексация. Функция len(). <i>Перебор элементов строки.</i>	1	1	2
22	Срезы строк. Отрицательные индексы в срезе. Шаг среза. <i>Решение задач на срезы</i>	1	1	2
23	Методы строк. <i>Подсчет. Поиск и замена</i>	1	1	2
24	Методы строк. <i>Сравнение строк.</i>	1	1	2
25	Методы строк. <i>Классификация символов.</i>	1	1	2
26	Форматирование строк. <i>Метод format(). f – строки.</i>	1	1	2
27	Представление строк в памяти компьютера, ASCII и Unicode. <i>Функции ord() и chr(). Решение задач</i>	1	1	2
<b>8. Списки. Кортежи</b>		<b>5</b>	<b>5</b>	<b>10</b>
28	Списки. Перебор элементов списка. Операции со списками. Срезы списков. <i>Методы списков. Отображение, фильтрация и сокращение. Удаление элементов списка. Создание синонимов в списке.</i>	3	3	6
29	Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. <i>Кортежи с переменным числом аргументов. Списки и кортежи. Сравнение кортежей. Последовательности последовательностей.</i>	2	2	4
<b>9. Функции. Создание собственных функций</b>		<b>4</b>	<b>5</b>	<b>9</b>
30	Понятие функция. Функции с параметрами и без параметров. Аргументы функции. <i>Создание функции draw_box()</i>	1	1	2

№ п/п	Тематический блок / Тема	Количество часов		
		теория	прак-а	всего
31	Локальные параметры и переменные. Глобальные переменные. Область видимости переменных. Ключевое слово <code>global</code> . <i>Использование функций в программах.</i>	1	1	2
32	Функции с возвратом значения. Возвращаемое значение. Инструкция <code>return</code> . <i>Решение задач</i>	1	1	2
33	Функции-предикаты. <i>Использование булевых функций для валидации входных данных</i> Функции с возвратом нескольких значений. <i>Решение задач</i>	1	2	3
<b>10. Проекты</b>		<b>0</b>	<b>2</b>	<b>2</b>
34	<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>	0	2	2
<b>ИТОГО</b>		<b>32</b>	<b>40</b>	<b>72</b>

# СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

## **1. Введение. Что такое язык программирования. Классификация языков программирования. Компиляция, интерпретация. Среда разработки. Техника безопасности**

### 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. Техника безопасности.

### 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.

## **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 числами**

### 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. Дополнительные операции с целыми числами (возведение в степень, остаток от деления, целочисленное деление). Модуль math. Особенности подключения модулей.

#### 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. Обработка цифр числа.
6. Решение задач.

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

##### Теория:

Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Основы строкового типа данных. Функция print() для вывода данных на экран. Необязательные параметры команды print (sep и end).

#### Практические занятия:

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

#### **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. Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций. Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий.

#### Практические занятия:

1. Применение условного оператора if...else для решения прикладных задач.
2. Использование логических операций.
3. Взаимодействие с числовыми и строковыми типами данных
4. Вложенные условия.
5. Множественные условия.
6. Ленивая оценка условий.
7. Сложные условия.
8. Формулировка условий.
9. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.

## **6. 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. Ключевое слово else в циклах. Вложенные циклы. Бесконечный цикл. Формулировка условия.

### Практические занятия:

1. Решение задач с помощью цикла for.
2. Бесконечный цикл.
3. Частые сценарии: подсчет количества
4. Вычисление суммы и произведения,
5. Обмен значений переменных, сигнальные метки
6. Нахождение максимума и минимума
7. Решение задач с использованием циклов while.
8. Решение задач различного типа с помощью циклических алгоритмов.

## **7. Строковый тип данных**

### Теория:

Строка как последовательность. Индексация. Функция len(). Срезы строк. Отрицательные индексы в срезе. Шаг среза. Методы строк. Подсчет. Поиск и замена. Методы строк. Сравнение строк. Методы строк. Классификация символов. Форматирование строк. Метод format(). f – строки. Представление строк в памяти компьютера, ASCII и Unicode. Функции ord() и chr().

### Практические занятия:

1. Перебор элементов строки.
2. Решение задач на срезы
3. Подсчет. Поиск и замена
4. Сравнение строк.
5. Классификация символов.
6. Метод format(). f – строки.
7. Функции ord() и chr(). Решение задач

## **8. Списки. Кортежи**

### Теория:

Списки. Перебор элементов списка. Операции со списками. Срезы списков. Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения.

### Практические занятия:

1. Методы списков.
2. Отображение, фильтрация и сокращение.
3. Удаление элементов списка.

4. Создание синонимов в списке.
5. Кортежи с переменным числом аргументов.
6. Списки и кортежи.
7. Сравнение кортежей.
8. Последовательности последовательностей.

## **9. Функции. Создание собственных функций**

### Теория:

Понятие функция. Функции с параметрами и без параметров. Аргументы функции. Локальные параметры и переменные. Глобальные переменные. Область видимости переменных. Ключевое слово `global`. Функции с возвратом значения. Возвращаемое значение. Инструкция `return`. Функции-предикаты. Функции с возвратом нескольких значений.

### Практические занятия:

1. Создание функции `drow_box()`.
2. Использование функций в программах.
3. Решение задач.
4. Использование булевых функций для валидации входных данных.

## **10. Проекты**

### 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), 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/l/python3>

Курс «Поколение Python: курс для начинающих»: <https://stepik.org/course/58852/>

## 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 текстовым 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 «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.	
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).

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
		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. Уважительное и корректное отношение к одноклассникам. Навык формулирования ответов и вопросов к одноклассникам и преподавателям.	Наблюдение. Беседа и тестирование на выявление коммуникативных склонностей.
Создание условий для формирования культуры общения между учащимися	Сформированность коммуникативной культуры обучающихся	Наличие коммуникативной культуры обучающихся. Отсутствие конфликтов в группах.	Наблюдение на уровень воспитанности.
Создания условий для возникновения потребности к самостоятельному изучению материала	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.	Самоанализ Беседа Проверка работ
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. Оценка перспективы применения полученных знаний в дальнейшей учебной и профессиональной деятельности.	Беседы на выявление мотивов обучающихся к занятию по данному профилю, беседы на выявление профессиональных намерений. Статистический учет профессионального выбора обучающихся.
Развивающие			
Развитие логического мышления	Применение логики при решении поставленных задач	Способность использовать логику при решении конкретных задач Выбор оптимального метода решения задачи.	Наблюдение. Анализ продуктов деятельности детей.
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. Использование оригинальных решений для рутинных задач, комбинирование условий.	Анализ и оценивание детских творческих достижений.

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
<p>Развитие интереса обучающихся к выбранному ими профилю деятельности</p>	<p>Развитие интереса обучающихся к программированию</p>	<p>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.</p>	<p>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.</p>
		<p>Применение знаний полученных в процессе обучения в решении смежных задач в процессе обучения и для бытового применения.</p>	
<p>Развитие трудолюбия и целеустремленности</p>	<p>Развитие трудолюбия и целеустремленности</p>	<p>Целеустремленность, трудолюбие, проявляемые при выполнении индивидуальных заданий.</p> <p>Учащийся не бросает попытки решить сложные задачи.</p>	<p>Наблюдение.</p>

**Личная карта обучающегося 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 КОНТРОЛЬНЫХ 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 3.12.

- 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.
- 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 «Основы программирования на языке Python» предназначен для начального знакомства с основными алгоритмами и с программированием на языке Python в интегрированных средах разработки (IDE) Python. 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](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. Програмуємо 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> _____					
<b>Продолжительность реализации программы</b> _____					
<b>Количество часов</b> _____					
<b>Расписание занятий</b> _____					
<b>Праздничные дни:</b> 1,2,3,4,5,6,7,8 января, 23 февраля, 8 марта, 1 мая, 9 мая, 4 ноября, 31 декабря					
<b>Промежуточный контроль:</b> декабрь <b>Итоговый контроль:</b> май					
№ п/п	Дата	Тематический блок / Тема	Количество часов		
			теория	прак-а	всего
<b>1. Введение. Что такое язык программирования. Классификация языков программирования. Компиляция, интерпретация. Среда разработки. Техника безопасности</b>			<b>1</b>	<b>1</b>	<b>2</b>
1		Введение. Что такое язык программирования. Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Области применения. Техника безопасности.	1	0	1
2		<i>Дистрибутивы интерпретатора языка Python. Установка официального дистрибутива интерпретатора языка Python 3.x. Запуск интерактивного режима Python. Работа в официальном IDLE Python.</i>	0	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	0	1
4		Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.	1	0	1
5		<i>Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().</i>	0	1	1
6		<i>Выполнение команд в интерактивном режиме. Создание скриптов. Сохранение скриптов. Программа приветствие.</i>	0	1	1

<b>3. Встроенные типы данных. Переменные. Основные операции с числами</b>			<b>4</b>	<b>4</b>	<b>8</b>
7 8		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
9 10		Базовые функции языка Python для работы с числовыми данными. <i>Ввод/вывод числовых данных. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.</i>	1	1	2
11 12		Дополнительные операции с целыми числами (возведение в степень, остаток от деления, целочисленное деление). <i>Обработка цифр числа.</i>	1	1	2
13 14		Модуль math. Особенности подключения модулей. <i>Решение задач.</i>	1	1	2
<b>4. Ввод/вывод данных в программу</b>			<b>2</b>	<b>2</b>	<b>4</b>
15 16		Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Основы строкового типа данных. <i>Использование функции input(). Преобразование типов вводимых данных.</i>	1	1	2
17 18		Функция print() для вывода данных на экран. Необязательные параметры команды print (sep и end). <i>Использование функции print(). Склеивание строк. Формирование строки.</i>	1	1	2
<b>5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор</b>			<b>4</b>	<b>4</b>	<b>8</b>
19 20		Понятие алгоритма их типы. Описание задачи в виде алгоритма. <i>Создание линейного алгоритма</i>	1	1	2
21 22		Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. <i>Применение условного оператора if...else для решения прикладных задач.</i>	1	1	2

23 24	Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций. <i>Использование логических операций. Взаимодействие с числовыми и строковыми типами данных</i>	1	1	2
25 26	Вложенные условия. Каскадные условия. Ленивая оценка условий. Формулировка условий. <i>Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.</i>	1	1	2
<b>6. Управляющие инструкции – циклы</b>		<b>3</b>	<b>8</b>	<b>11</b>
27	Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций	1	0	1
28 29 30	Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. <i>Решение задач с помощью цикла for.</i>	1	2	3
31 32 33 34	<i>Частые сценарии: подсчет количества, вычисление суммы и произведения, обмен значений переменных, сигнальные метки, нахождение максимума и минимума</i>	0	4	4
35 36 37	Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Ключевое слово else в циклах. Вложенные циклы. Бесконечный цикл. Формулировка условия. <i>Решение задач с использованием циклов while.</i>	1	2	3
<b>7. Строковый тип данных</b>		<b>7</b>	<b>7</b>	<b>14</b>
38 39	Строка как последовательность. Индексация. Функция len(). <i>Перебор элементов строки.</i>	1	1	2
40 41	Срезы строк. Отрицательные индексы в срезе. Шаг среза. <i>Решение задач на срезы</i>	1	1	2
42 43	Методы строк. <i>Подсчет. Поиск и замена</i>	1	1	2
44 45	Методы строк. <i>Сравнение строк.</i>	1	1	2
46 47	Методы строк. <i>Классификация символов.</i>	1	1	2
48 49	Форматирование строк. <i>Метод format(). f – строки.</i>	1	1	2

50 51		Представление строк в памяти компьютера, ASCII и Unicode. <i>Функции ord() и chr(). Решение задач</i>	1	1	2
<b>8. Списки. Кортежи</b>			<b>5</b>	<b>5</b>	<b>10</b>
52 53 54 55 56 57		Списки. Перебор элементов списка. Операции со списками. Срезы списков. <i>Методы списков. Отображение, фильтрация и сокращение. Удаление элементов списка. Создание синонимов в списке.</i>	3	3	6
58 59 60 61		Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. <i>Кортежи с переменным числом аргументов. Списки и кортежи. Сравнение кортежей. Последовательности последовательностей.</i>	2	2	4
<b>9. Функции. Создание собственных функций</b>			<b>4</b>	<b>5</b>	<b>9</b>
62 63		Понятие функция. Функции с параметрами и без параметров. Аргументы функции. <i>Создание функции draw_box()</i>	1	1	2
64 65		Локальные параметры и переменные. Глобальные переменные. Область видимости переменных. Ключевое слово global. <i>Использование функций в программах.</i>	1	1	2
66 67		Функции с возвратом значения. Возвращаемое значение. Инструкция return. <i>Решение задач</i>	1	1	2
68 69 70		Функции-предикаты. <i>Использование булевых функций для валидации входных данных</i> Функции с возвратом нескольких значений. <i>Решение задач</i>	1	2	3
<b>10. Проекты</b>			<b>0</b>	<b>2</b>	<b>2</b>
71 72		<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>	0	2	2
<b>ИТОГО</b>			<b>32</b>	<b>40</b>	<b>72</b>