

**Муниципальное общеобразовательное учреждение  
«Ломоносовская общеобразовательная школа №3»**

**ПРИНЯТА**

на заседании педагогического совета

МОУ «Ломоносовская

школа №3»

Протокол № 1 от 29.08.2023 г.

**УТВЕРЖДАЮ**

Приказ № 285 от 31.08.2023 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
«Программирование на языке Python»**

**Направленность: техническая**

**Возраст обучающихся: 12-18 лет.**

**Срок реализации: 1 год**

**Разработчик:** педагог дополнительного образования Р.Н. Набоких

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеразвивающая программа социально-гуманитарной направленности «Программирование на языке Python» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации».
- Приказом Министерства просвещения Российской Федерации № 629 от 27 июля 2022 года «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- Концепцией развития дополнительного образования детей до 2030 г. (утвержденного распоряжением Правительства Российской Федерации от 31 марта 2022 года № 678-р).
- СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденными постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 (далее – СП 2.4.3648-20).
- Письмом Минобрнауки России № 09-3242 от 18.11.2015 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»).
- Порядком применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ, утвержденного приказом Министерства образования и науки Российской Федерации от 23 августа 2017 г. № 816.
- Письмом Комитета общего и профессионального образования Ленинградской области от 1 апреля 2015 г. № 19-2174/15-0-0 «О методических рекомендациях по разработке и оформлению дополнительных общеразвивающих программ различной направленности».
- Уставом и соответствующими локальными актами МОУ «Ломоносовская школа №3»

**Уровень освоения - стартовый**

### Актуальность.

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

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

**Программа адресована** обучающимся 12–18 лет, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей учащихся (самих учащихся с 14 лет). Группы формируются из школьников разного возраста на добровольной внеконкурсной основе.

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

**Объем программы**– 68 часа, 1 год обучения.

**Форма обучения** – очная, возможно использование дистанционных технологий.

**Формы проведения занятий** – аудиторные, внеаудиторные (самостоятельные). Формами проведения занятий являются: инструктаж, анкетирование, рассказ, беседа, демонстрация, анализ, работа с электронными образовательными ресурсами, самостоятельная работа, презентация работ, защита проекта, конкурс, творческая работа, деловая игра.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

**Язык обучения** – русский.

**Режим занятий:** продолжительность занятий исчисляется в академических часах – 45 минут. Между занятиями установлен 10-минутный перерыв. Недельная нагрузка на группу 2 часа. В условиях дистанционного обучения занятие проводится в течение 30 минут с перерывом на 5 минут (при необходимости на 10 минут) для проведения физкультминутки.

### **Особенности организации образовательного процесса**

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

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

В рамках предлагаемого курса «Программирования на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизирует процесс индивидуально-личностного становления учащихся.

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

### **Задачи программы:**

#### **Обучающие:**

способствовать улучшению усвоения и углублению знаний обучающимися предмета «Информатика» в соответствии с современными требованиями к Государственной итоговой аттестации в формате ОГЭ;

стимулировать познавательную активность и повысить мотивацию учеников к изучению образовательных областей Информатика и информационно-коммуникативные технологии, в том числе в разрезе выбора ими технического профиля обучения;

формировать навыки самостоятельной исследовательской, творческой и проектной деятельности;

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

#### **Развивающие:**

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

#### **Воспитательные:**

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;

- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

### **Условия реализации программы**

#### **Условия набора в коллектив:**

В кружок принимаются все желающие из обучающихся 9-х классов МОУ «Ломоносовская школа №3», имеющих базовые знания по предмету «Информатика и ИКТ» за курс 8 класса. (По итогам года за 8 класс). Результатом освоения программы является зачетная работа по типу ОГЭ.

#### **Условия формирования групп:**

Набор в группы свободный, основной набор – 1 сентября, возможен дополнительный набор в течение года. Наполняемость группы – от 15-25 человек.

#### **Организационно-педагогические условия.**

Кабинет, соответствующий санитарным нормам (СанПиН 2.4.4.1251-03) с индивидуальными рабочими местами для обучающихся и отдельным оборудованным рабочим местом для педагога дополнительного образования. Формирование групп и расписания занятий в соответствии с требованиями Сан ПиН и программой. Персональные компьютеры и программное обеспечение.

#### **Кадровые условия.**

Педагог дополнительного образования, образование высшее.

#### **Материально-технические условия.**

Персональный компьютер с постоянным доступом в Интернет (скорость 50 Мб). Сканер, принтер, мультимедиа проектор, экран, школьная доска.

#### **Методические условия.**

Дидактический материал (раздаточный материал по темам занятий программы, наглядный материал, мультимедийные презентации). Компакт-диски с обучающими и информационными программами по основным темам программы. Видеоуроки. Архив видео и фотоматериалов. Методические разработки занятий, УМК к программе.

### **Планируемые результаты**

#### **Личностные результаты**

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

#### **Планируемые метапредметные результаты** Метапредметные результаты отражают:

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;

#### **Регулятивные универсальные учебные действия:**

- умение применять индуктивные и дедуктивные способы рассуждения, видеть различные стратегии решения задач;
- умение планировать и осуществлять деятельность, направленную на решение задач исследовательского характера;
- умение демонстрировать свое речевое и неречевое поведение в учебных и внеучебных ситуациях;
- способность сознательно организовать и регулировать свою деятельность – учебную, общественную и др., контролировать и корректировать деятельность, давать ей оценку;
- умение генерировать идеи и определять средства для их реализации;

- умение вносить необходимые дополнения и коррективы в план и способ действия в случае расхождения эталона, реального действия и его продукта.

### **Познавательные универсальные учебные действия:**

- владение умениями работать с учебной и внешкольной информацией (анализировать и обобщать факты, составлять план, тезисы, формулировать и обосновывать выводы);  
- способность к решению творческих задач, участие в проектной и учебно-исследовательской деятельности;

- понимание различий между исходными фактами и гипотезами, теоретическими моделями и реальными объектами для их объяснения;

- овладение универсальными учебными действиями на примерах гипотез для объяснения известных фактов и экспериментальной проверки выдвигаемых гипотез, разработки теоретических моделей, процессов или явлений;

- овладение навыками самостоятельного приобретения новых знаний;

- чтение текстов с извлечением необходимой информации, умение анализировать информацию, сопоставлять факты, делать заключения и выводы;

- умение взаимодействовать со сверстниками и взрослыми, работать в группах над задачами исследовательского характера;

- умение контролировать, корректировать и оценивать свои действия и действия партнеров;

- развитие и активное проявление коммуникативной компетенции (речевой, языковой, социокультурной, компенсаторной, учебно-познавательной), включая умение взаимодействовать с окружающими, выполняя разные социальные роли;

- владение навыками организации и участия в коллективной деятельности

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.

- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.

- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.

- Научить организовывать данные для эффективной алгоритмической обработки.

- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.

### **Планируемые предметные результаты**

В результате обучения по программе «Программирование на языке Python» обучающийся научится по темам:

#### **Знакомство с языком Python**

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест № 1. Знакомство с языком Python

*Учащиеся должны знать / понимать:*

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

*Учащиеся должны уметь:*

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

#### **Переменные и выражения**

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест № 2. Выражения и операции.

*Учащиеся должны знать / понимать:*

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

### **Условные предложения**

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы". Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

*Учащиеся должны знать / понимать:*

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

### **Циклы**

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

*Учащиеся должны знать / понимать:*

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

*Учащиеся должны уметь:*

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

### **Функции**

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций Практическая работа 5.4.

Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции" Тест № 5. Функции

*Учащиеся должны знать / понимать:*

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

*Учащиеся должны уметь:*

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

### **Строки - последовательности символов**

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

*Учащиеся должны знать / понимать:*

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

*Учащиеся должны уметь:*

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

### **Сложные типы данных**

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. Списки

*Учащиеся должны знать / понимать:*

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;

- способы описания множества;
- операторы работы с множествами.

*Учащиеся должны уметь:*

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы

элементов списка;

- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

### **Стиль программирования и отладка программ**

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

*Учащиеся должны знать / понимать:*

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

*Учащиеся должны уметь:*

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

### **УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

/п	Наименование тем	Количество часов			Формы аттестации/ контроля
		сего	Теоория	Практика	
•	<b>Раздел 1. Знакомство с языком Python .Инструктаж по технике безопасности в компьютерном классе</b>		2	2	Опрос, решение практических задач
•	<b>Раздел 2. Переменные и выражения</b>		2	6	решение практических задач
	<b>Раздел 3. Условные предложения</b>	4	3	11	Тестирование, Решение практических задач
	<b>Раздел 4. Циклы</b>	4	4	10	Тестирование, решение практических задач, творческая работа
	<b>Раздел 5. Функции</b>	0	1	9	Тестирование, решение практических задач
	<b>Раздел 6. Строки - последовательности символов</b>		1	3	Решение практических задач
	<b>Раздел 7. Сложные типы данных</b>	0	3	7	Тестирование, Решение практических задач
	<b>Раздел 8. Стиль программирования и отладка программ</b>		1	3	Решение практических задач, презентация проекта, рефлексия.
	<b>ВСЕГО</b>	<b>8</b>	<b>7</b>	<b>51</b>	



## СОДЕРЖАНИЕ КУРСА

Раздел 1. Знакомство с языком Python .Инструктаж по технике безопасности в компьютерном классе

Общие сведения о языке

Практическая работа: Установка программы Python

Режимы работы

Практическая работа: Режимы работы с Python

Раздел 2. Переменные и выражения

Переменные

Практическая работа: Работа со справочной системой

Выражения. Практическая работа: Выражения

Практическая работа: Переменные

Практическая работа: Логические выражения

Ввод и вывод

Ввод и вывод

Задачи на элементарные действия с числами

Задачи на элементарные действия с числами

Раздел 3. Условные предложения

Логические выражения и операторы.

Условный оператор

Практическая работа: «Условный оператор»

Практическая работа: «Условный оператор»

Множественное ветвление

Практическая работа: Множественное ветвление

Практическая работа: Множественное ветвление

Реализация ветвления в языке Python.

Практическая работа: «Условные операторы»

Самостоятельная работа по теме «Условные операторы».

Самостоятельная работа по теме «Условные операторы».

Составление программ с ветвлением".

Составление программ с ветвлением".

Раздел 4. Циклы

Оператор цикла с условием

Практическая работа "Числа Фибоначчи"

Оператор цикла for

Практическая работа Решение задачи с циклом for.

Вложенные циклы

Практическая работа: Реализация циклических алгоритмов

Случайные числа

Практическая работа: Случайные числа

Примеры решения задач с циклом

Практическая работа: Решение задач с циклом.

Самостоятельная работа "Составление программ с циклом"

Самостоятельная работа "Составление программ с циклом"

Творческая работа "Циклы"

Творческая работа "Циклы"

Раздел 5. Функции

Создание функций

Практическая работа Создание функций

Локальные переменные

Примеры решения задач с использованием функций

Практическая работа Решение задач с использованием функций

Самостоятельная работа по теме "Функции"

Самостоятельная работа по теме "Функции"

Рекурсивные функции  
Практическая работа «Рекурсивные функции»

Раздел 6. Строки

Строки - последовательности символов

Срезы строк

Примеры решения задач со строками

Практическая работа: Решение задач со строками.

Раздел 7. Сложные типы данных

Списки: примеры решения задач

Срезы списков

Матрицы

Матрицы

Кортежи

Введение в словари

Множества в языке Python

Множества в языке Python

Раздел 8. Стилль программирования и отладка программ

Стилль программирования

Отладка программ

Зачет по курсу «Программирование на языке Python»

Зачет по курсу «Программирование на языке Python»

### Оценочные и методические материалы

#### **Методы, используемые на занятиях:**

**Объяснительно-иллюстративные** (методы обучения, при использовании которых обучающиеся воспринимают и усваивают готовую информацию - рассказ, беседа, объяснение, обсуждение, демонстрация наглядных пособий и иллюстраций, показ документов, фотографий, видеofilьмов).

**Репродуктивные методы обучения** (обучающиеся воспроизводят полученные знания и освоенные способы деятельности).

**Частично-поисковые методы обучения** (участие детей в коллективном поиске, решение поставленной задачи совместно с педагогом, выполнение практических заданий, самостоятельная работа, работа по заданному алгоритму/плану).

**Исследовательские методы обучения** (овладение обучающимися методами научного познания, самостоятельной познавательной и творческой деятельности).

**Рассказ, беседа** позволяют изложить теоретический материал, необходимый для освоения программы. **Демонстрация наглядных, образцов, пособий и иллюстративного и видео материала** способствует лучшему усвоению теории и изучению практических приемов работы. **Выполнение практических заданий** способствует закреплению теоретических основ и позволяет обучающимся отработать практические навыки и приемы работы с документами, получить опыт выступлений и освоения разных социальных ролей в коллективе.

#### **Педагогические технологии:**

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

#### Формы контроля

**Входной контроль** осуществляется при комплектовании группы в начале учебного года. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

*Форма контроля:* тестирование.

**Текущий контроль** осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

**Промежуточный контроль** осуществляется в декабре и в мае, 2 раза в год. Форма контроля: тест, решение практических задач средствами языка программирования.

**Итоговый контроль** осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

Вид контроля	Формы контроля	Срок контроля
Входной	Тест	сентябрь
Текущий	решение практических задач средствами языка программирования	В течение года
Промежуточный	Тест	декабрь
Итоговый	Защита творческого проекта	май

Работа состоит из 15 заданий: базового уровня сложности 10, повышенного — 3, высокого — 2. Заданий с кратким ответом — 12, с развернутым ответом — 3. Работа рассчитана на 150 минут.

Обозначение уровня сложности задания: Б — базовый, П — повышенный, В — высокий.

Проверяемые элементы содержания и виды деятельности	Уровень сложности задания	Максимальный балл за выполнение задания	При мерное время выполнения задания (мин.)
<b>Задание 1.</b> Оценивать объём памяти, необходимый для хранения текстовых данных	Б	1	3
<b>Задание 2.</b> Уметь декодировать кодовую последовательность	Б	1	4





<b>20.</b>	Умение написать короткий алгоритм в среде формального исполнителя (вариант задания 20.1) или на языке программирования (вариант задания 20.2)																		
------------	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Варианты заданий**

Низкий уровень: 0-70% выполненных заданий;

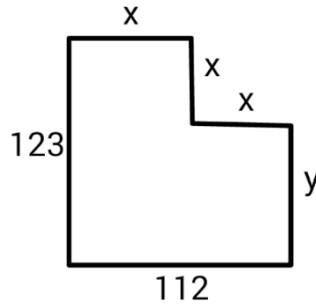
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

**Для учащихся – 14-17 лет**

<b>Фамилия, Имя</b>	
---------------------	--

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




2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок.

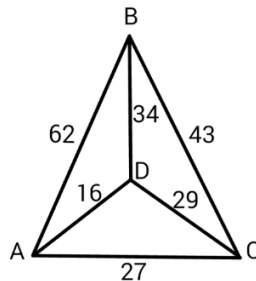
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

**Примечания**

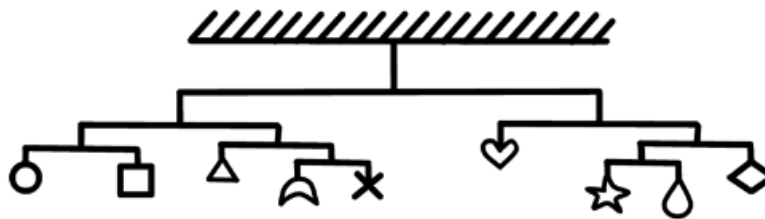
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

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

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




5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?




6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

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

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

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

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

**Значит, некоторые клёны быстро желтеют.**

<input type="checkbox"/>	а
<input type="checkbox"/>	ет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Для учащихся 11-13 лет

Фамилия, Имя.	
---------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?
3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?
4. Как с помощью только одной палочки образовать на столе треугольник?
5. В каком месяце болтливая девочка говорит меньше всего?
6. Что становится на треть больше, если его поставить вверх ногами?
7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?
8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.  
**Вопрос:** Сколько кусков пиццы получилось у Юры.
9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.  
**Вопрос:** Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

### Приложение 3

#### Итоговый контроль Диагностическая карта

/п	Имя	Входящая Диагностика	Промеж- уточ-ная диагностика	Итоговая диагностика
+				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

#### Низкий уровень

Учащиеся должны знать / понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах,
- понятие программы Python; общую структуру программы;
- основные типы данных;



- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов с условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

*Учащиеся должны уметь:*

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;
- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

### **Средний уровень**

*Учащиеся должны знать / понимать:*

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;

- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

*Учащиеся должны уметь:*

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;

- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

### **Высокий уровень**

*Учащиеся должны знать / понимать:*

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;

- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;
- приемы успешной презентации проекта.

*Учащиеся должны уметь:*

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;

- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

<b>Вопросы для промежуточного контроля по усвоению материала</b> <i>(Промежуточный контроль проводится в последнюю неделю декабря)</i>
<p>Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:</p> <p style="text-align: center;">11111·1111111</p> <p>— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)</p>
<p>Запишите число <b>1.2345e3</b> в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления:</p> <p style="text-align: center;"><b>2014.0^14(Возвестив 14 степень)</b></p> <p>Обратите внимание на запись числа: это вещественное число.</p>
<p><b>Приведите к целому типу число 2.99</b></p>
<p>Расставьте скобки в выражении</p> <p style="text-align: center;"><b>a and b or not a and not b</b></p> <p>в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).</p>
<p>Найдите результат выражения для заданных значений <i>a</i> и <i>b</i> Учитывайте регистр символов при ответе.</p> <p><b>a = True</b> <b>b = False</b> <b>a and b or not a and not b</b></p>
<p>Отметьте выражения, значения которых равны True:</p> <ul style="list-style-type: none"> <li>• "239" &lt; "30" and 239 &lt; 30</li> <li>• "239" &lt; "30" and 239 &gt; 30</li> <li>• "239" &gt; "30" and 239 &lt; 30</li> <li>• "239" &gt; "30" and 239 &gt; 30</li> </ul>
<p>Укажите результат выражения: "123" + "42"</p>
<p>Какое значение будет у переменной <i>i</i> после выполнения фрагмента программы?</p> <pre><b>i = 0</b> <b>while i &lt;= 10:</b>   <b>i = i + 1</b>   <b>if i &gt; 7:</b>     <b>i = i + 2</b></pre>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы?</p> <pre><b>i = 0</b> <b>while i &lt;= 10:</b>   <b>i = i + 1</b>   <b>if i &gt; 7:</b>     <b>i = i + 2</b></pre>
<p>Сколько всего знаков * будет выведено после исполнения фрагмента программы:</p> <pre><b>i = 0</b> <b>while i &lt; 5:</b></pre>

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

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

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

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

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

### Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, \*, mod, pow, div, где  
mod — это взятие остатка от деления,  
pow — возведение в степень,  
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты. Для числа  $\pi$  в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где  $a$ ,  $b$  и  $c$  — длины сторон треугольника

прямоугольник

a

b

где  $a$  и  $b$  — длины сторон прямоугольника

круг

r

где  $r$  — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример:123321)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число  $n$  — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если  $n = 7$ , то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером  $n \times n$ , заполненную числами от 1 до  $n^2$  по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в

### Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).
- Проверка достигаемых учащимися результатов производится в следующих формах:
- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка педагогом деятельности учащихся;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

## КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

### Календарный учебный график

Работа по программе осуществляется с 01 сентября 2023г. по 24 мая 2024 г.

### Условия реализации программы

#### **Материально-техническое обеспечение:**

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

#### **Информационное обеспечение:**

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

#### **Аппаратное обеспечение:**

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

#### **Программное обеспечение:**

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



## Список литературы

### Нормативно-правовые акты и документы

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

### Литература для педагога

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

### Литература для родителей и учащихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
  2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
  3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
  4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
- Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: [openbookproject.net](http://openbookproject.net), свободный.

## КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН

2023-2024 учебный год

Один год обучения по программе «Программирование на языке Python»

Педагог д/о Набоких Роман Николаевич

Характеристика группы: \_\_\_\_\_

№	План	Факт	Раздел	Тема	Содержание	Всего	Теория	Практика
1.	7.09		Знакомство с языком Python . Инструктаж по технике безопасности и в	Общие сведения о языке	Познакомиться с основным содержанием курса. Наметьте	1	1	
2.	7.09			Практическая работа: Установка программы Python		1		
3.	14.09			Режимы работы		1	1	
4.	14.09			Практическая работа: Режимы работы с Python		1		

			компьютерном классе		перспективы совершенствования умений и навыков в процессе учебной деятельности. Определить основные требования к результатам обучения и критерии успешной работы учащихся			
5.	21.09		Переменные и выражения	Переменные	Понимать общую структуру программы. Знать типы данных, операторы. Составлять простые программы.	1	1	
6.	21.09	Практическая работа: Работа со справочной системой		1				
7.	28.09	Практическая работа: Переменные		1				
8.	28.09	Выражения. Практическая работа: Выражения		1				
9.	5.10	Ввод и вывод		1		1		
10.	5.10	Ввод и вывод		1				
11.	12.10	Задачи на элементарные действия с числами		1				
12.	12.10	Задачи на элементарные действия с числами		1				
13.	19.10		Условные предложения	Логические выражения и операторы.	Узнать назначение условного оператора и логической данных. Использовать условный оператор. Создавать сложные условия с помощью логических операторов	1	1	
14.	19.10	Практическая работа: Логические выражения		1				
15.	26.10	Условный оператор		1		1		
16.	26.10	Практическая работа: «Условный оператор»		1				
17.	9.11	Практическая работа: «Условный оператор»		1				
18.	9.11	Множественное ветвление		1		1		
19.	16.11	Практическая работа: Множественное ветвление		1				
20.	16.11	Практическая работа: Множественное ветвление		1				
21.	23.11	Реализация ветвления в языке Python.		1				
22.	23.11	Практическая работа: «Условные операторы»		1				
23.	30.11	Самостоятельная работа по		1				

				теме «Условные операторы».	в.			
24.	30.11			Самостоятельная работа по теме «Условные операторы».		1		
25.	7.12			"Составление программ с ветвлением".		1		
26.	7.12			"Составление программ с ветвлением".		1		
27.	14.12			Оператор цикла с условием	Узнать циклы с условием и их виды. Правила записи циклов условием. Использовать циклов различных типов. определять вид цикла, наиболее удобный для решения поставленной задачи. Определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.	1	1	
28.	14.12			Практическая работа "Числа Фибоначчи"		1		
29.	21.12			Оператор цикла for		1	1	
30.	21.12			Практическая работа Решение задачи с циклом for.		1		
31.	28.12			Вложенные циклы		1	1	
32.	28.12			Практическая работа: Реализация циклических алгоритмов		1		
33.	11.01			Случайные числа		1	1	
34.	11.01			Практическая работа: Случайные числа		1		
35.	18.01			Примеры решения задач с циклом		1		
36.	18.01			Практическая работа: Решение задач с циклом.		1		
37.	25.01		Циклы	Самостоятельная работа "Составление программ с циклом"		1		
38.	25.01			Самостоятельная работа "Составление программ с циклом"		1		
39.	01.02			Творческая работа "Циклы"		1		
40.	01.02			Творческая работа "Циклы"		1		
41.	8.02					Создание функций	1	1
42.	8.02			Практическая работа Создание функций		1		
43.	15.02		Функции	Локальные переменные	1			
44.	15.02			Практическая работа Локальные переменные	1			
45.	22.02			Примеры решения задач с использованием функций	1			

46.	22.02			Практическая работа Решение задач с использованием функций	ирования. Подпрограмм.	1		
47.	29.02			Самостоятельная работа по теме "Функции"	Создание и	1		
48.	29.02			Самостоятельная работа по теме "Функции"	использование	1		
49.	7.03			Рекурсивные функции	функции;	1		
50.	7.03			Практическая работа Рекурсивные функции	использование механизм параметров для передачи значений	1		
51.	14.03		Строки - последовательности символов	Строки. Практическая работа: Строки	Описывание строк. Нахождение длины строк.	1		
52.	14.03	Срезы строк		1				
53.	21.03	Примеры решения задач со строками		1		1		
54.	21.03	Практическая работа: Решение задач со строками.		1				
55.	4.04		Сложные типы данных	Списки. Практическая работа: Списки	Описывать сложные типы данных. Описывание списков. Вывод элементов списка.	1		
56.	4.04	Срезы списков		1		1		
57.	11.04	Списки: примеры решения задач		1				
58.	11.04	Практическая работа. Решение задач со списками		1				
59.	18.04	Матрицы		1		1		
60.	18.04	Матрицы		1				
61.	25.04	Кортежи		1				
62.	25.04	Введение в словари		1		1		
63.	2.05	Множества в языке Python		1				
64.	2.05	Множества в языке Python		1				
65.	16.05		Стиль	Стиль программирования	Основные правила при написании и выполнении тестирования и отладку программ	1		
66.	16.05	Отладка программ		1				
67.	23.05	Зачет по курсу «Программирование на языке Python»		1		1		
68.	23.05	Зачет по курсу «Программирование на языке Python»		1				
						68	17	1