

**Муниципальное общеобразовательное учреждение  
«Средняя общеобразовательная школа п.Пробуждение  
им. Л. А. Кассиля»  
Энгельсского муниципального района Саратовской области**

Принята на заседании  
педагогического совета  
Протокол № 11  
от «31» мая 2023года

«Утверждаю»  
Руководитель  
МОУ "Средняя  
общеобразовательная  
школа п. Пробуждение  
им. Л. А. Кассиля"  
Кузнецова Ю.Г.  
Приказ №  
от «31» мая 2023года



**Дополнительная общеобразовательная  
общеразвивающая программа**

**«Юные программисты»**

Направленность: техническая  
Срок реализации программы: 4 месяца  
Объем программы: 32 часа  
Возраст детей: 14-18 лет

Ф.И.О. Малышева Ирина Сергеевна  
педагог дополнительного образования

Энгельс 2023

## **Пояснительная записка**

Дополнительная общеразвивающая программа «Юные программисты» разработана в соответствии с Положением дополнительного образования МОУ «СОШ п. Пробуждение им. Л.А. Кассиля» (приказ №398 от 26.09.2019 года).

Программа предложена для реализации со школьниками в возрасте 14-18 лет. Образовательная деятельность организуется по подгруппам, количество детей в подгруппе (12-17).

**Направленность:** техническая

### **Актуальность**

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

### **Отличительные особенности программы обучения**

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

### **Новизна программы**

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

### **Возраст и возрастные особенности обучающихся**

Программа предложена для реализации с школьниками в возрасте 14-18 лет. Программа рассчитана на обучающихся подросткового возраста, имеющих базовые навыки программирования. Дифференцируется содержание образования, обусловленное профессиональной ориентацией и личностными интересами. Расширяется диапазон видов деятельности, увлечений. Усиливается стремление выразить себя.

### **Педагогическая целесообразность**

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию алгоритмического, логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и

сложные программы.

**Адресат программы** – обучающиеся 14-18 лет. Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

**Сроки реализации программы:** 32 часа.

**Принцип набора в объединение** – свободный.

**Цели и задачи**

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

**Задачи программы:**

**Обучающие.**

- обучить языку программирования Python;
- научить создавать прикладное программное обеспечение;
- расширить кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

**Развивающие:**

- Развить память, внимание, познавательную и творческую активность
- Развивать техническое и проектное мышление.
- Развить познавательные и творческие способности обучающихся, прививать активно познавательный подход к жизни
- Развить устойчивый интерес к поисковой творческой деятельности.
- Развивать мотивацию доведения решения задач до реализации в программе.
- Развить умение работать над проектом в команде, эффективно распределять обязанности.
- Развить умение излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений.

**Воспитательные:**

- привить интерес к активному творческому самовыражению, культуре труда;
- воспитать упорство в достижении желаемого результата;
- воспитать чувство взаимопомощи, доверия, коллективизма.

**Планируемые результаты освоения программы:**

**Личностные результаты:**

- сформированность интереса к активному творческому самовыражению, культуре труда;
- сформированность ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- сформированность коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

**Метапредметные результаты:**

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления

осознанного выбора в учебной и познавательной деятельности;

- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов;
- формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

**Предметные результаты:**

- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ;
- владение элементарными навыками формализации прикладной задачи и документирования программ;
- получение опыта использования компьютерных средств представления и анализа данных.

**В результате изучения дисциплины учащиеся должны знать:**

- основные типы алгоритмов;
- иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;
- содержание этапов разработки программы: алгоритмизация-кодирование-отладка-тестирование;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;
- основы разработки простых игр в системе программирования Python.

**Учащиеся должны уметь:**

- записывать алгоритмические структуры на языке программирования Python;
- использовать Python для решения задач из области математики, физики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- использовать основные алгоритмические приемы при решении математических задач; – решать простые, сложные и нестандартные задачи;
- создавать простые игры;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения

**Формы и режим занятий**

Занятия проходят 1 раз в неделю по 2 академических часа с 15 минутным перерывом.

Сроки реализации программы: Программа рассчитана на 32 часа.

**Формы контроля и подведения итогов**

В процессе реализации программы предусмотрены следующие виды контроля:

- **входной контроль проводится** с целью определения уровня знаний учащихся (Приложение № 1);
- **промежуточный контроль** проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;
- **итоговый контроль** – защита проекта.

## Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1.	Введение в Python	5	2	3	Тест
2.	Алгоритмические конструкции	6	3	3	Тест
3.	Строки	5	2	3	Решение задач
4.	Функции	5	1	4	Решение задач
5.	Списки и кортежи	4	2	2	Решение задач
6	Словари и множества	3	1	2	Решение задач
7	Разработка и программирование собственного проекта	4	-	4	Защита проекта
	<b>Всего:</b>	<b>32</b>	<b>11</b>	<b>21</b>	

### Содержание разделов учебного плана

#### 1. Введение в Python

*Основные понятия:* трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

*Теория.* Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы. Ввод и вывод числовой информации. Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

*Практика.* Установка языка программирования Python 3.x и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах. Тренировочное задание на ввод и вывод числовой информации. Решение простых задач в интерактивном режиме

#### 2. Алгоритмические инструкции

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

*Теория.* Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

*Практика.* Решение задач на анализ чисел потока и целочисленной арифметики. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

#### 3. Строки

*Основные понятия:* символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

*Теория.* Понятие «литералы строк». Форматирование строки. Срезы строк. Методы работы со строкой.

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

Преобразование строки. Решение упражнений. Применение методов строки. Решение задач.

#### **4. Функции**

*Основные понятия:* подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

*Теория.* Обзор парадигм программирования. Особенности применения языков программирования. Применение стек и граф вызовов функций. Понятие прямой рекурсии и ее применение. Понятие косвенной рекурсии и ее применение.

*Практика.* Выполнение тренировочных упражнений. Нахождение суммы чисел. Вычисление степени. Числа Фибоначчи. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности

#### **5. Списки и кортежи**

*Основные понятия:* список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

*Теория.* Представление списка и кортежа в памяти компьютера, сходства и различия. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле). Методы работы со списком и кортежем. Методы сортировки списка. Вычисление сложности алгоритма. Многомерные списки (матрицы).

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

#### **6. Словари и множества**

*Основные понятия:* словарь, множество, ключ, кодирование.

*Теория.* Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

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

*Практика.* Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря. Решение задач повышенной трудности.

#### **7. Разработка и программирование собственного проекта**

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

#### **Организационно-педагогические условия реализации программы.**

##### **Учебно-методическое обеспечение программы**

Занятия проводятся в следующих формах: лекции, семинары, практические задания.

Дидактический материал, необходимый для проведения занятий:

- краткие конспекты материалов для лекций;
- распечатки заданий для практикумов;
- презентационные материалы для объяснения;
- карточки с индивидуальными заданиями.

В конце занятия проходит обсуждение результатов и оценка проделанной работы.

##### **Материально-технические условия реализации программы.**

Занятия проводятся в помещении, где есть учебная зона: столы, стулья по количеству участников.

- компьютер для демонстрации презентаций;
- проектор;
- рабочие компьютеры учащихся для работы с доступом в Интернет;
- принтер для распечатки заданий.

**Требования к педагогическому работнику:** высшее педагогическое образование, первая или высшая квалификационная категория.

## Информационное обеспечение программы

Интернет-ресурсы:

1. <http://informatics.mccme.ru/course/view.php?id=156>
2. [https://inf5.ru/podgotovka\\_k\\_olympiad/olym\\_zadachi\\_s\\_resheniyami.htm](https://inf5.ru/podgotovka_k_olympiad/olym_zadachi_s_resheniyami.htm)
3. <http://anng Georg.ru/olimp/materials>
4. <http://kpolyakov.spb.ru/school/probook/python.htm>
5. <http://dist-olimpiada.krasnogorka.edusite.ru/p4aa1.html>

## Список литературы

*Для педагога:*

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
1. 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.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: [openbookproject.net](http://openbookproject.net), свободный.



### Календарный график учебного процесса

№	Дата	Тема занятия	Количество часов	Место проведения	Форма проведения	Форма Аттестации/контроля
1		Вводное занятие. Инструктаж по ТБ. Язык программирования Python. Достоинства и недостатки.	1	Технологический кабинет	Комбинированное	Педагогическое наблюдение, беседа
2		Области применения. Интерактивный режим работы программы.	1	Технологический кабинет	Комбинированное	Педагогическое наблюдение, беседа
3		Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.	1	Технологический кабинет	Практическое	демонстрация результатов
4		Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).	1	Технологический кабинет	Практическое	Практическая работа Тест
5		Решение простых задач в интерактивном режиме.	1	Технологический кабинет	Практическое	Практическая работа
6		Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
7		Цикл while. Теория чисел. Фильтрация потока чисел.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
8		Решение задач на анализ чисел потока и целочисленной арифметики (НОД и НОК)	1	Технологический кабинет	Практическое	Практическая работа
9		Цикл for. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
10		Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Проверка числа на простоту).	1	Технологический кабинет	Практическое	Практическая работа

11		Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Нахождение максимума и минимума).	1	Технологический кабинет	Практическое	Практическая
12		Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
13		Форматирование строки.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
14		Преобразование строки. Решение упражнений.	1	Технологический кабинет	Практическое	Практическая работа
15		Методы работы со строкой. Решение задач.	1	Технологический кабинет	Практическое	Практическая работа
16		Применение методов строки. Решение задач.	1	Технологический кабинет	Практическое	Практическая работа
17		Применение стек и граф вызовов функций.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
18		Понятие прямой рекурсии и ее применение.	1	Технологический кабинет	Практическое	Практическая работа
19		Выполнение тренировочных упражнений.	1	Технологический кабинет	Практическое	Практическая работа
20		Понятие косвенной рекурсии и ее применение.	1	Технологический кабинет	Практическое	Практическая работа
21		Выполнение тренировочных упражнений.	1	Технологический кабинет	Практическое	Практическая работа
22		Представление списка и кортежа в памяти компьютера, сходства и различия.	1	Технологический кабинет	Теоретическое	Опрос
23		Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
24		Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.	1	Технологический кабинет	Практическое	Практическая работа
25		Вычисление сложности алгоритма. Многомерные списки.	1	Технологический кабинет	Практическое	Практическая работа
26		Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа

		Словари со смешанными значениями. Кодирование и декодирование текста.				
27		Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.	1	Технологический кабинет	Теоретическое	Педагогическое наблюдение, беседа
28		Решение задач повышенной трудности.	1	Технологический кабинет	Практическое	Практическая работа
29		Выбор вида и темы проекта. Составление технического задания.	1	Технологический кабинет	Практическое	Практическая работа
30		Программирование.	1	Технологический кабинет	Практическое	Практическая работа
31		Работа над созданием проекта Разработка технической документации и презентации проекта.	1	Технологический кабинет	Практическое	Практическая работа
32		Защита проекта	1	Технологический кабинет	Защита проекта	Опрос, демонстрация результатов
		Итого:	32			

**Входящий контроль**

Низкий уровень: 0-70% выполненных заданий;

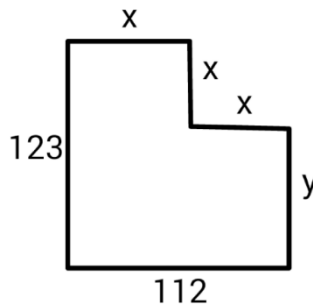
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

**Для учащихся – 14-17 лет**

<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.

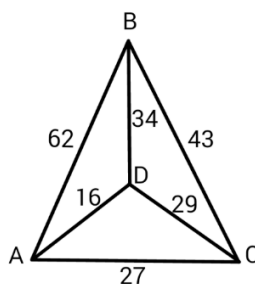
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

**Примечания**

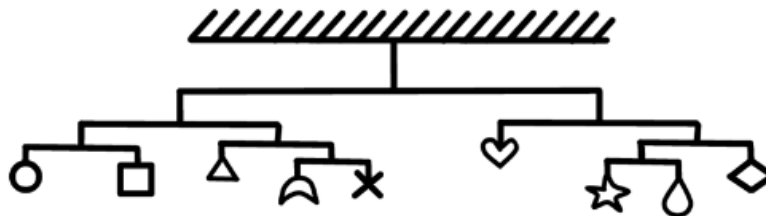
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

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.**


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?

**Итоговый контроль  
Диагностическая карта**

№/ п	Имя	Входящая Диагностика	Промежуточ- ная диагностика	Итоговая диагностика
1				
2				
3+				
	Итого в %	Н С В		Н С В

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

**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 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.

**вопросы для промежуточного контроля по усвоению материала**  
(Промежуточный контроль проводится в последнюю неделю декабря)

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

11111·1111111

— **произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)**

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

<p>Составьте и запишите выражение для вычисления:  <b><math>2014.0^{14}</math>(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  <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.  <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> i = 0 while i &lt;= 10:     i = i + 1     if i &gt; 7:         i = i + 2 </pre>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы?</p> <pre> i = 0 while i &lt;= 10:     i = i + 1     if i &gt; 7:         i = i + 2 </pre>
<p>Сколько всего знаков * будет выведено после исполнения фрагмента программы:</p> <pre> i = 0 while i &lt; 5:     print('*')     if i % 2 == 0:         print('**')     if i &gt; 2:         print('***')     i = i + 1 </pre>
<p>Определите, какое значение будет иметь переменная <i>i</i> после выполнения следующего фрагмента программы:</p> <pre> i = 0 s = 0 </pre>

```
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 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.

Напишите программу, которая выводит 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 примере (здесь  $n=5$ )