

Муниципальное бюджетное общеобразовательное учреждение
«Селекционная средняя общеобразовательная школа»
Структурное подразделение
Центр образования цифрового и гуманитарного профилей
«ТОЧКА РОСТА»

Принята на заседании
педагогического совета
от «29» августа 2023
Протокол № 1



**Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Python для начинающих»**

возраст обучающихся: 12-13 лет
срок реализации 1 год

Автор-составитель:
Фриауф Людмила Николаевна,
учитель информатики

с. Селекционное
2023

ОГЛАВЛЕНИЕ

I.	Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы -----	3
	1.1 Пояснительная записка -----	3-4
	1.2 Цель, задачи, ожидаемые результаты -----	5
	1.3 Содержание программы -----	6-8
II.	Комплекс условий реализации программы -----	8
	2.1 Календарный учебный график -----	8
	2.2 Условия реализации программы -----	8-9
	2.3 Формы аттестации -----	9
	2.4 Оценочные материалы -----	9-10
	2.5 Методические материалы -----	10-11
	2.6 Список литературы -----	12
	Приложение -----	13

Раздел 1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1. Пояснительная записка

Нормативные правовые основы разработки ДООП:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ».
- Концепция развития дополнительного образования детей до 2030г. (Распоряжение Правительства РФ от 31.03.2022 № 678-р).
- Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»)
- Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Приказ Главного управления образования и молодежной политики Алтайского края от 19.03.2015 № 535 «Об утверждении методических рекомендаций по разработке дополнительных общеобразовательных (общеразвивающих) программ».
- Об утверждении санитарно-эпидемиологических правил СП 3.1/2.4.3598-20 "Санитарно-эпидемиологические требования к устройству, содержанию и организации работы образовательных организаций и других объектов социальной инфраструктуры для детей и молодежи в условиях распространения новой коронавирусной инфекции (COVID-19)" (с изменениями на 20 июня 2022 года).

По результатам исследования компании Jet Brains (<https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/>) язык Python – самый изучаемый в 2019 году и один из самых востребованных на рынке труда. Python достаточно широко используется при изучении основ алгоритмизации и программирования в школьном курсе информатики.

Актуальность программы

Программа изучения Python в школе откроет ученикам возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решению заданий ЕГЭ, удовлетворит запросы детей и родителей.

Отличительная особенность

Дополнительная общеобразовательная общеразвивающая программа «Python для начинающих» имеет **техническую направленность**. За основу взята рабочая программа «Поколение Python». Авторы курса: Белых Роман, Гув Тимур, Скобина Ольга Благотворительный фонд развития образования “Айкью Опшн” (“Возможность Интеллекта”) Отличительные особенности программы: авторский подход к организации образовательного процесса. Реализация программы предполагает проведение аудиторных занятий с использованием вариативных дистанционных ресурсов:

- онлайн уроков, размещенных на платформе Stepik;
- интерактивных заданий (опросов, викторин, дидактических игр и др.), размещенных на бесплатных специализированных сервисах Kahoot и Learning Apps;
- авторских проектных заданий для создания учащимися игр на языке Python.

Программа особенно будет интересна и полезна тем, кто хочет получить навыки программирования и в дальнейшем планирует связать профессиональную деятельность с ИТ-технологиями.

Педагогическая целесообразность программы заключается в том, что используемые технологии, формы, методы являются наиболее действенными для освоения программного материала учащимися. Данные средства позволяют сохранить высокий творческий подход при обращении к теории и ведут к более глубокому ее усвоению. Принцип доступности, различные формы самостоятельной активности на уроке, групповая взаимопроверка, проектная деятельность позволяют учащимся не только овладеть знаниями, но использовать полученные навыки в смежных предметах.

Адресат программы:

Возраст обучающихся: 12 - 13 лет. Занятия проводятся по группам. Группы учащихся формируются одного возраста. Состав группы обучающихся – постоянный.

Психолого-педагогические особенности возрастной группы учащихся:

Развитие детей 12 – 13 лет осуществляется главным образом на основе ведущей деятельности – учения. В процессе учебной деятельности ребенок не только усваивает знания, умения и навыки, но и учится ставить перед собой учебные задачи, находить способы усвоения и применения знаний, контролировать и оценивать свои действия. Программа опирается на целенаправленное, организованное и планомерное восприятие ребенка. Определяющее значение имеет отношение подростка к наблюдаемому объекту. Дети стремятся вникать в сущность процессов, глубоко осмысливать заключенные в них причины и следствия.

Программа развивает потребностно-мотивационную сферу подростков, широко используются элементы занимательности учебных занятий.

Совокупный объем: 34 часа

Срок реализации: 1 год

Формы и режим работы:

Курс состоит из 34 занятий длительностью 40 минут.

Форма обучения: очная.

В очных занятиях в группах по 5-10 человек используются задания на онлайн-платформе с автоматизированной проверкой. Задания на платформе доступны ученикам для самостоятельного изучения в любое время. В ходе курса запланирован плавный переход к формату “перевернутого класса”: на первых пяти занятиях учеников знакомят с онлайн-платформой и основными инструментами программирования на Python. Начиная с шестого урока предлагаются различные формы самостоятельной активности на уроке, с групповой взаимопроверкой. После десятого урока иногда ученикам предлагается самостоятельно изучить новую тему, опираясь на конспект онлайн-курса, а роль учителя сдвигается к консультации и модерации. Последние семь уроков отведены на почти самостоятельную работу над проектами.

Формы организации деятельности обучающихся на занятии: индивидуальная, групповая, фронтальная.

1.2. Цель, задачи, ожидаемые результаты

Цель реализации программы – формирование у обучающихся навыков программирования на языке Python и самонаправленного обучения.

Задачи реализации программы:

Обучающие

1. Изучить основы программирования на языке Python;
2. Научиться применять полученные знания для решения практических задач.
3. Повысить уровень самостоятельности в обучении

Ожидаемые результаты

Предметные

По окончании курса учащийся сможет:

- Программировать на языке Python.
- Использовать инструменты разработки среды Wing.
- Самостоятельно реализовывать проекты, связанные с разработкой игр.

Метапредметные

Научатся применять полученные знания для решения практических задач:

- Ставить учебные цели.
- Формулировать достигнутый результат.
- Планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели.
- Определять подходы и методы для достижения поставленной цели.
- Отбирать необходимые средства для достижения поставленной цели.
- Осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.
- Проводить рефлексию своей учебно-познавательной деятельности.

Личностные:

- Повысить уровень самостоятельности в обучении.

Практическим результатом работы служит финальный проект каждого ученика: серия самостоятельно разработанных мини-проектов в консольном режиме языка Python.

1.3. Содержание программы

В связи с проектным подходом каждая из тем, изложенных преподавателем, отрабатывается в уроках платформы Stepik и в практических заданиях. (<https://stepik.org/course/58852/syllabus>)

Модуль 1. Ввод-вывод данных

Тема 1. Знакомство с Python. Команды `input()` и `print()`.

Теория. История языка Python. Сильные и слабые стороны Python. Python 2 VS Python 3. Философия Python. Ввод и вывод данных в языке Python.

Практика. Установка Python на компьютер. Установка Wing IDE на компьютер. Решение задач. Несложные программы, которые умеют что-то выводить на экран (команда `print()`) и считывать информацию с клавиатуры (команда `input()`)

Тема 2. Параметры `sep`, `end`. Переменные. Комментарии. PEP 8.

Теория. Необязательные параметры команды `print()`: `sep`, `end`. Переменные. Стандарт PEP 8. Комментарии.

Практика. Решение задач на изучение необязательных параметров команды `print()` и более детальному разбору понятия переменных.

Тема 3. Работа с целыми числами.

Теория. Целочисленный тип данных. Преобразование строки к целому числу. Операции над целыми числами. Операция возведения в степень. Операция нахождения остатка. Операция целочисленного деления. Обработка цифр числа.

Практика. Решение задач на работу с целыми числами, обработку цифр целого числа. Изучение основных и дополнительных операций, а также операции преобразования строки в число.

Модуль 2. Условный оператор

Тема 4. Условный оператор. Логические операции `and`, `or`, `not`.

Теория. Условный оператор. Отступы. Операторы сравнения. Логическое умножение `and`. Логическое сложение `or`. Логическое отрицание `not`.

Практика. Решение задач с использованием условного оператора и логических операций.

Тема 5. Вложенный и каскадный условный оператор

Теория. Вложенные условия. Каскадные условия.

Практика. Решение задач с использованием каскадного условного оператора и вложенных ветвлений.

Модуль 3. Типы данных

Тема 6. Типы данных `int`, `float`, `str`. Встроенные функции `min()`, `max()`, `abs()`. Оператор `in`.

Теория. Целочисленный тип данных `int`. Числа с плавающей точкой `float`. Встроенные функции `max()`, `min()`, `abs()`. Строковый тип данных `str`. Функции `len()` и `str()`. Конкатенация строк. Умножение строки на число. Оператор `in`.

Практика. Решение задач с числовыми типами данных. Особенности работы с целыми числами, работа с числами с плавающей точкой. Встроенные функции для работы с числами `max`, `min`, `abs`.

Решение задач со строковым типом данных, использование встроенных функций `len()`, `str()`, а также операторов `+`, `*`, `in`.

Модуль 4. Циклы `for` и `while`

Тема 7. Цикл *for*. Функция *range()*

Теория. Цикл *for*. Понятие переменной цикла. Функция *range()* с одним параметром. Функция *range()* с двумя параметрами. Функция *range()* с тремя параметрами

Практика. Решение задач на использование цикла и функции *range()*, а также двух дополнительных перегрузок функции *range()*, позволяющих настраивать элементы последовательности.

Тема 8. Частые сценарии при написании циклов. Расширенные операторы присваивания

Теория. Частые сценарии при написании циклов. Расширенные операторы присваивания.

Практика. Решение задач: подсчет количества, вычисление суммы и произведения, обмен значений переменных, сигнальные метки, определение максимума и минимума.

Тема 9. Цикл с предусловием *while*

Теория. Цикл *while*, считывание данных до стоп значения, бесконечный цикл, использование цикла *while* для обработки цифр числа.

Практика. Решение задач на использование цикла с предусловием. Изучение процедуры обработки цифр натурального числа.

Тема 10. Операторы *break*, *continue*, *else*

Теория. Оператор *break*, оператор *continue*. Бесконечные циклы. Ключевое слово *else* в циклах.

Практика. Работа над ревью кода. Отработка применения оператора *break* в циклах.

Тема 11. Вложенные циклы

Теория. Вложенные, находящиеся внутри других циклов, циклы. Использование вложенных циклов при решении уравнений.

Практика. Отработка применения вложенных циклов.

Модуль 5. Строковый тип данных

Тема 12. Строковый тип данных: индексация и срезы

Теория. Строковый тип данных. Основные операции над строками, работа с отдельными символами, итерирование символов строки. Срезы строк. Изменение символов строки.

Практика. Решение задач по теме.

Тема 13. Методы строк

Теория. Строковый тип данных, основные методы конвертации регистра, методы поиска и замены строк внутри других строк, методы классификации символов. Представление строк в памяти компьютера. Таблицы символов ASCII и Unicode. Функция *ord()*. Функция *chr()*

Практика. Решение задач на использование методов строк.

Модуль 6. Списки

Тема 14. Введение в списки

Теория. Создание списков. Пустые списки. Встроенная функция *list()*. Вывод списков.

Практика. Решение задач на использование списков.

Тема 15. Основы работы со списками. Методы списков

Теория. Метод добавления элемента *append()*. Метод расширения списка *extend()*. Оператор *del*.

Практика. Решение задач на отработку методов работы со списками.

Тема 16. Вывод элементов списка. Строковые методы *split()* и *join()*

Теория. Вывод элементов списка. Строковые методы *split()* и *join()*.

Практика. Решение задач на работу со списками.

Тема 17. Методы списков. Списочные выражения

Теория. Методы *insert()*, *index()*, *remove()*, *pop()*, *reverse()*, *count()*, *clear()*, *copy()*, *sort()*.

Списочные выражения.

Практика. Решение задач на отработку методов списков и списочных выражений.

Модуль 7. Функции

Тема 18. Функции

Теория. Функции без параметров. Преимущества использования функций. Ключевое слово *pass*. Функции с параметрами. Область видимости параметрической переменной. Параметры по умолчанию.

Практика. Решение задач на создание пользовательских функций без параметров и функций с параметрами.

Тема 19. Локальные и глобальные переменные. Функции, возвращающие значения

Теория. Локальные переменные. Область действия локальной переменной. Глобальные переменные. Глобальные константы. Функции с возвратом числового значения.

Практика. Решение задач по теме «Локальные и глобальные переменные».

II. Комплекс условий реализации программы

2.1. Календарный учебный график

Календарный учебный график составлен на основе Годового календарного графика учреждения и является документом, регламентирующим организацию образовательной деятельности в учреждении.

Начало учебного года 01.09.2023 г. Окончание учебного года 23.05.2024 г. Продолжительность учебного года 34 недели.

Год обучения (уровень)	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год обучения	01.09.2023 г.	23.05.2024 г.	34	68	68 часов, 2 часа в неделю	2 раз в нед. по 2 часа

2.2. Условия реализации программы

Для успешной реализации программы необходимы:

Материально - техническое обеспечение

- учебный кабинет;
 - 10 рабочих мест: стол, стул, розетка, компьютеры на каждое рабочее место;
 - проектор, аудио-колонки;
 - Интернет-соединение, скорость загрузки не менее 2 Мбит/сек;
 - магнитно-маркерная доска или флипчарт;
 - общие условия в соответствии с СанПиНом 2.4.4.3172-14

Требования к ПО:

- Операционная система Windows 7 или моложе / MacOS / Unix-based системы с поддержкой протокола HTML5;
- Приложения Google Chrome, Gimp, Python;
- интерактивная оболочка (бесплатная IDE Wing101 или аналог)

Информационно-методическое обеспечение

В помощь педагогу разработана серия дидактических материалов, которая включает:

- планы-конспекты занятий;
- презентации;
- дополнительные материалы.

2.3. Формы аттестации

Для отслеживания результативности образовательной деятельности по программе проводятся: текущий, промежуточный и итоговый контроль.

Текущий контроль проводится на каждом занятии в форме опроса (теста) и заданий на онлайн-тренажере. Необходимый минимум выполнения заданий модуля 70%.

Промежуточный контроль проводится после каждого модуля программы в виде проверочной работы на онлайн-платформе.

Итоговый контроль - оценка уровня и качества освоения учащимися дополнительной общеобразовательной общеразвивающей программы по завершению всего периода обучения по программе.

Формы проверки результатов:

- наблюдение за обучающимися в процессе работы;
- игры;
- индивидуальные и коллективные творческие проекты;
- беседы с обучающимися и их родителями.

Формы отслеживания и фиксации образовательных результатов:

- журнал посещаемости;
- материал тестирования;
- демонстрация созданных проектов.

Итоговая аттестация обучающихся проводится по результатам подготовки и защиты проекта.

2.4. Оценочные материалы

Результативность обучения обеспечивается применением различных форм, методов и приемов, которые тесно связаны между собой и дополняют друг друга. Большая часть занятий отводится практической работе. Содержание и объем материала, подлежащего проверке, определяется Программой. При проверке усвоения материала необходимо выявлять полноту, прочность усвоения обучающимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Текущий контроль проводится на каждом занятии в форме опроса (теста) и заданий на онлайн-тренажере. Необходимый минимум выполнения заданий модуля 70%.

Промежуточный контроль проводится после каждого модуля программы в виде проверочной работы на онлайн-платформе.

Итоговый контроль (зачетное занятие), на котором обучающиеся представляют свои работы и обсуждают их, проходит в конце учебного года в форме защиты проектов.

Формы проведения аттестации:

- тестирование;
- самостоятельная работа;
- защита проектов.

Предметные

- Отслеживается динамика успешности выполнения практических работ и творческих заданий.

Метапредметные

Отслеживается:

- полнота и степень успешности выполнения практических задач;
- уровень развития психофизиологических качеств учащихся: память, внимание, способность логически мыслить, анализировать, концентрировать внимание на главном.

Личностные

- Отслеживается динамика уровня воспитанности, умения совместно выполнять поставленные задачи в атмосфере доброжелательности и комфорта.

2.5. Методические материалы

Данная программа предполагает смешанный формат обучения. Сочетание групповой работы с учителем в классе и индивидуальной работы в личном кабинете на онлайн-платформе, позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Для предъявления учебной информации используются следующие методы:

- наглядные;
- словесные;
- практические.

Для стимулирования учебно-познавательной деятельности применяются методы:

- игровые;
- проблемных ситуаций.

В образовательном процессе педагогом используются следующие образовательные технологии:

- технология смешанного обучения «перевернутый класс»
- информационно-коммуникативные технологии;
- проектный метод обучения;
- здоровьесберегающие технологии.

Технология смешанного обучения «перевернутый класс» – это современная технология осуществления процесса обучения, при котором учащиеся с помощью цифровых средств и интернет-ресурсов прослушивают и просматривают видео-уроки, изучают дополнительные источники информации во внеурочное время, затем совместно обсуждают новые понятия и различные идеи, а педагог помогает применять полученные знания на практике.

Проектный метод обучения - выполнение индивидуального или группового творческого проекта по какой – либо теме.

В данном методе обучающиеся самостоятельно и охотно приобретают недостающие знания

из разных источников; учатся пользоваться приобретенными знаниями для решения познавательных и практических задач; приобретают коммуникативные умения, работая в различных группах; развивают исследовательские умения (умения выявления проблем, сбора информации, наблюдения, проведения эксперимента, анализа, построения гипотез, общения); развивают системное мышление.

Для развития личностных качеств (силы воли, аккуратности, ответственности), повышения мотивации к саморазвитию, развития навыков исследовательской работы обучающимся целесообразно участвовать в конкурсах проектной направленности.

Информационно-коммуникативные технологии. Применение ИКТ на всех этапах занятия (при объяснении нового материала для наглядности, на этапе закрепления синтаксиса языка программирования, при обобщении и повторении, при контроле, при защите проектов) позволяет оптимизировать образовательный процесс, эффективно использовать время, активизировать познавательную деятельность учащихся, повысить мотивацию к изучению языка программирования, а так же создает дополнительные условия для формирования и развития коммуникативных умений и навыков учащихся.

Здоровьесберегающая технология особенно важна при долгой работе за компьютером при программировании.

2.6. Список литературы

Литература для педагога

1. Васильев, А. Н. Python на примерах [Текст]: практ. курс / А. Н. Васильев - Наука и Техника, 2019 - 432 с.
2. Прохоренок, Н. А. Python 3: самое необходимое [Текст]: практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
3. Гэддис, Т. Начинаем программировать на Python [Текст]: учебник / Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
4. Седжвик, Р. Программирование на языке Python / Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. Харрисон, М. Как устроен Python. [Текст]: практ. курс / М. Харрисон - Питер, 2002 - 272 с.

Литература для обучающихся: Не предусмотрена

Интернет-ресурсы:

1. Курс [Поколение Python: курс для начинающих](https://stepik.org/course/58852/syllabus) на платформе Stepik. (<https://stepik.org/course/58852/syllabus>)
2. Документация Python 3 <https://docs.python.org/3/>
3. Документация по языку Python3 <https://docs-python.ru/>

Приложение 1

Здоровьесберегающие технологии

На занятиях нужно систематически проводить гимнастику для глаз, физкультминутки, выполнять упражнения для пальцев, рук, плеч. Преподаватель должен постоянно напоминать обучающимся о правильной посадке, следить за положением рук, спины, ног.

Примерный комплекс упражнений для глаз:

1. Закрывать глаза, сильно напрягая глазные мышцы, на счет 1 - 4, затем открыть глаза, расслабить мышцы глаз, посмотреть вдаль на счет 1-6. Повторить 4-5 раз.

2. Посмотреть на переносицу и задержать взгляд на счет 1 - 4. До усталости глаза не доводить. Затем посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз.

3. Не поворачивая головы, посмотреть направо и зафиксировать взгляд на счет 1 - 4, затем посмотреть вдаль прямо на счет 1 - 6. Аналогичным образом проводятся упражнения с фиксацией взгляда влево, вверх и вниз. Повторить 3 - 4 раза.

4. Перевести взгляд быстро по диагонали: направо вверх - налево вниз, потом прямо вдаль на счет 1 - 6; затем налево вверх - направо вниз и прямо вдаль на счет 1-6. Повторить 4 -5 раз.

Создание комфортного психологического климата, учет индивидуальных особенностей обучающихся, их темперамента, умение работать с различными группами обучающихся, использовать дифференцированный подход в обучении.

Необходимо создавать благоприятный эмоциональный климат на занятиях. Чувство успеха при выполнении заданий, очень положительно влияет на здоровье обучающихся. Проявление доверительного подхода к обучающимся, наличие на занятиях эмоциональной разрядки, создание ситуации успеха – оптимальные приемы создания положительных эмоций на занятии.

И последним, но немаловажным компонентом здоровьесберегающих технологий является формирование, пропаганда здорового образа жизни. Если показывать обучающимся, как соотносится изучаемый материал с повседневной жизнью, приучать их постоянно заботиться о своем здоровье, систематически закреплять их на занятиях, это должно стать для них совершенно естественным и при работе в домашних условиях.

Усталости и перегрузок на занятиях будет меньше, если будет чередоваться и вид деятельности обучающихся, и способы преподнесения информации (зрительная, слуховая) с физкультурными паузами. А создание благоприятной атмосферы в начале и в конце занятия, через улыбку, будет способствовать хорошему настроению обучающихся, как на самом занятии, так и после него.

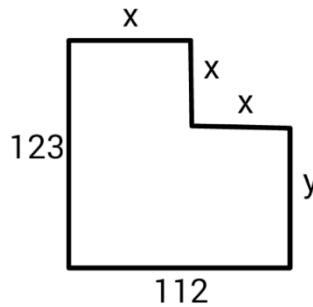
Приложение 2
Входящий контроль

Низкий уровень: 0-70% выполненных заданий;
 Средний уровень: 70-85% правильно выполненных заданий;
 Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 15-18 лет

Фамилия, Имя	
---------------------	--

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.

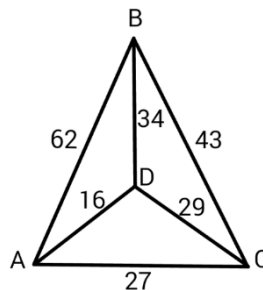
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

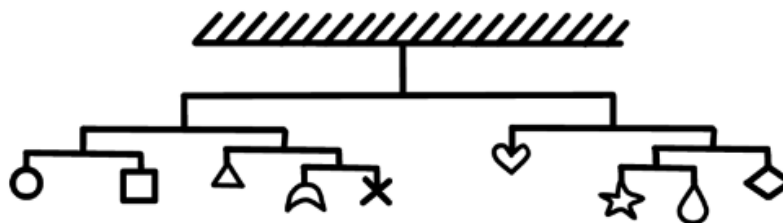
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

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.

т

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?

Приложение 3 Итоговый контроль

Диагностическая карта

№/п	Имя	Входящая Диагностика	Промежуточная 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.

<p>Вопросы для промежуточного контроля по усвоению материала <i>(Промежуточный контроль проводится в последнюю неделю декабря)</i></p>
<p>Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:</p> <p style="text-align: center;">11111·11111111</p> <p>— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)</p>
<p>Запишите число 1.2345e3 в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления:</p> <p style="text-align: center;">2014.0^14(Возвестив 14 степень)</p> <p>Обратите внимание на запись числа: это вещественное число.</p>
<p>Приведите к целому типу число 2.99</p>
<p>Расставьте скобки в выражении</p> <p style="text-align: center;">a and b or not a and not 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>a = True b = False a and b or not a and not b</p>
<p>Отметьте выражения, значения которых равны True:</p> <ul style="list-style-type: none"> • "239" < "30" and 239 < 30 • "239" < "30" and 239 > 30 • "239" > "30" and 239 < 30 • "239" > "30" and 239 > 30
<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 <= 10: i = i + 1 if i > 7: i = i + 2 </pre>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы?</p> <pre> i = 0 while i <= 10: i = i + 1 if i > 7: i = i + 2 </pre>
<p>Сколько всего знаков * будет выведено после исполнения фрагмента программы:</p> <pre> i = 0 while i < 5: </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 π 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.

Введите 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 $n=5$**)

КОНТРОЛЬ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Объекты контроля	Класс		
Запланировано уроков			
1 четверть			
2 четверть			
3 четверть			
4 четверть			
год			
Проведено уроков			
1 четверть			
2 четверть			
3 четверть			
4 четверть			
год			
Запланировано тем (разделов)			
1 четверть			
2 четверть			
3 четверть			
4 четверть			
год			
Проведено тем (разделов)			
1 четверть			
2 четверть			
3 четверть			
4 четверть			
год			
Запланированы (даты)	контрольные	лабораторные	практические
1 четверть			
2 четверть			
3 четверть			
4 четверть			
Проведены (даты)			
1 четверть			
2 четверть			
3 четверть			
4 четверть			
Причина невыполнения			