

Муниципальное общеобразовательное учреждение
«Алгатуйская средняя общеобразовательная школа»

Рассмотрена
На заседании методического
Совета
Протокол № 1 от 30.08.23



Утверждена
Директор МОУ «Алгатуйская СОШ»
Е.И. Гапеевцева
Приказ № 95 от 31.08.2023г

Дополнительная общеобразовательная (общеразвивающая)
программа технической направленности
**Основы программирования на языке Python, на примере
разработки беспилотного летательного аппарата**

Срок реализации – 1 год
Возраст обучающихся -12-18 лет
Автор-составитель – Гапеевцев И.А.,
педагог ДОП образования

Алгатуй, 2023 г.

Пояснительная записка

Дополнительная общеразвивающая программа (ДОП) «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» разработана с учётом рабочей программы воспитания на основе:

Закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями);

Распоряжение Правительства Российской Федерации от 04.09.2014 № 1726-р «Концепция развития дополнительного образования детей»;

Стратегия развития воспитания в Российской Федерации до 2025г

Приказ Министерства просвещения РФ от 9 ноября 2018 г. N 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (с изменениями от 30.09.2020).

Приказ Министерства просвещения РФ от 3 сентября 2019 г. № 467 «Об утверждении Целевой модели развития систем дополнительного образования детей».

Постановлениями Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»».

Направленность (профиль) программы информационно-техническая.

Актуальность программы: в настоящее время процесс информатизации проявляется во всех сферах человеческой деятельности. Использование современных информационных технологий является необходимым условием успешного развития как отдельных отраслей, так и государства в целом. Создание, внедрение, эксплуатация, а также совершенствование информационных технологий невозможно без участия квалифицированных и увлечённых специалистов, в связи с этим внедрение курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» в учебный процесс актуально.

Отличительные особенности программы: Данная Программа учебного курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» отличается тем что направлена на подготовку творческой, технически грамотной, гармонично развитой личности, обладающей логическим мышлением, способной анализировать и решать задачи в команде в области информационных и аэротехнологий, решать ситуационные кейсовые задания, основанные на групповых проектах.

Занятия по данному курсу рассчитаны на общенаучную подготовку обучающихся, развитие их мышления, логики, математических способностей, исследовательских навыков.

Учебный курс «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» направлен на изучение

основ программирования на языке Python и программирование автономных квадрокоптеров.

В рамках курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» обучающиеся смогут познакомиться с физическими, техническими и математическими понятиями. Приобретённые знания будут применимы в творческих проектах.

Учебный курс «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» представляет собой самостоятельный модуль и содержит необходимые темы из курса информатики и физики.

Адресат программы: учащиеся 5 – 11 классов.

Объем и срок освоения программы: 1 год, 306 часов.

Форма обучения: очная, групповая и индивидуальная, теоретические и мотивационные занятия могут проводиться в дистанционном формате в режиме видеоконференции, обучающего ролика, презентации. Учащиеся зачисляются в группы на добровольной основе по личному выбору, подачи заявления, путём регистрации на выбранную программу в системе «Навигатор дополнительного образования Иркутской области» За учащихся до 15 лет заявление на программу подаёт родитель (законный представитель). В течение года обучающийся может менять группы по своему усмотрению и согласованию с руководителем программы.

Условия организации занятий для учащихся с ОВЗ: Для учащихся с ДЦП Предусмотрено применение специальных методов обучения и воспитания. Здание школы оборудовано пандусом, предусмотрены сокращение времени занятий для учащихся с ОВЗ, увеличение количества динамических перемен, увеличение срока освоение программы.

Режим занятий Занятия проводятся в специализированных классах в разновозрастных группах по **8-15** (Требование СанПиН) человек. Зачисление в группы проходит по желанию обучающегося в течение всего года. Занятия проходят на базе структурного подразделения МОУ «Алгатуйская СОШ» Центр образования цифрового и гуманитарного профилей «Точка роста». 3 группы по 2 занятия в неделю каждая. Продолжительность одного занятия - 2 часа. Соревнования, конкурсы и спортивные мероприятия согласованы с календарно-тематическим планом воспитательной работы и проходят во второй половине дня с учётом нормативов для внеурочной деятельности.

Виды занятий: основная форма организации занятий – учебное занятие, проводится с использованием фронтального, группового, поточного и других методов, с учётом возрастных и физических особенностей обучающихся. Программой предусмотрены инструктажи, практические занятия в специально оборудованном кабинете. Кроме того, предусмотрены теоретические занятия, мотивационные практики, общеразвивающие занятия, проекты.

Цель программы: освоение Hard- и Soft-компетенций обучающимися в области программирования и аэротехнологий через использование кейс-технологий.

Данная программа решает следующие задачи:

Обучающие:

- изучить базовые понятия: алгоритм, блок-схема, переменная, цикл, условия, вычисляемая функция;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять навыки программирования на конкретной учебной ситуации (программирование беспилотных летательных аппаратов на учебную задачу);
- развить навык пилотирования беспилотных летательных аппаратов (БПЛА) на практике;
- привить навыки проектной деятельности.

Развивающие:

- способствовать расширению словарного запаса;
- способствовать развитию памяти, внимания, технического мышления, изобретательности;
- способствовать развитию алгоритмического мышления;
- способствовать формированию интереса к техническим знаниям;
- способствовать формированию умения практического применения полученных знаний;
- сформировать умение формулировать, аргументировать и отстаивать своё мнение;
- сформировать умение выступать публично с докладами, презентациями и т. п.

Воспитательные:

- воспитывать аккуратность и дисциплинированность при выполнении работы;
- способствовать формированию положительной мотивации к трудовой деятельности;
- способствовать формированию опыта совместного и индивидуального творчества при выполнении командных заданий;
- воспитывать трудолюбие, уважение к труду;
- формировать чувство коллективизма и взаимопомощи;
- воспитывать чувство патриотизма, гражданственности, гордости за достижения отечественной науки и техники.

Прогнозируемые результаты и способы их проверки

Личностные результаты:

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
- развитие самостоятельности суждений, независимости и нестандартности мышления;

- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

Метапредметные результаты:

Регулятивные универсальные учебные действия:

- умение принимать и сохранять учебную задачу;
- умение планировать последовательность шагов алгоритма для достижения цели;
- умение ставить цель (создание творческой работы), планировать достижение этой цели;
- умение осуществлять итоговый и пошаговый контроль по результату;
- способность адекватно воспринимать оценку наставника и других обучающихся;
- умение различать способ и результат действия;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок;
- умение в сотрудничестве ставить новые учебные задачи;
- способность проявлять познавательную инициативу в учебном сотрудничестве;
- умение осваивать способы решения проблем творческого характера в жизненных ситуациях;
- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Познавательные универсальные учебные действия:

- умение осуществлять поиск информации в индивидуальных информационных архивах обучающегося, информационной среде образовательного учреждения, федеральных хранилищах информационных образовательных ресурсов;
- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- умение ориентироваться в разнообразии способов решения задач;
- умение осуществлять анализ объектов с выделением существенных и несущественных признаков;
- умение проводить сравнение, классификацию по заданным критериям;
- умение строить логические рассуждения в форме связи простых суждений об объекте;
- умение устанавливать аналогии, причинно-следственные связи;
- умение моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- умение синтезировать, составлять целое из частей, в том числе самостоятельно достраивать с восполнением недостающих компонентов.

Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;

- умение выслушивать собеседника и вести диалог;
- способность признавать возможность существования различных точек зрения и права каждого иметь свою;
- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия;
- умение осуществлять постановку вопросов: инициативное сотрудничество в поиске и сборе информации;
- умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;
- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владение монологической и диалогической формами речи.
-

Предметные результаты

В результате освоения программы, обучающиеся должны

знать:

- основные алгоритмические конструкции;
- принципы построения блок-схем;
- принципы структурного программирования на языке Python;
- что такое БПЛА и их предназначение.

уметь:

- составлять алгоритмы для решения прикладных задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- применять библиотеку Tkinter;
- отлаживать и тестировать программы, написанные на языке Python;
- настраивать БПЛА;
- представлять свой проект.

владеть:

- основной терминологией в области алгоритмизации и программирования;
- основными навыками программирования на языке Python;
- знаниями по устройству и применению беспилотников.

Содержание тем программы

№	Темы занятий	Содержание занятий
1.	Введение в программирование	

1.1	Вводное занятие. Введение в предмет, техника безопасности.	Теория: введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ.
1.2	Основы языка Python. Примеры на языке Python с разбором конструкций: ввод и вывод данных, условия, циклы, ветвления, массивы.	Теория: история языка Python, сфера применения языка, различие в версиях, особенности синтаксиса. Объявление и использование переменных в Python. Использование строк, массивов, кортежей и словарей в Python. Использование условий, циклов и ветвлений в Python. Практика: запуск интерпретатора. Различия интерпретатора и компилятора. Написание простейших демонстрационных программ. Мини-программы внутри программы. Выражения в вызовах функций. Имена переменных. Упражнения по написанию программ с использованием переменных, условий и циклов. Генерация случайных чисел. Группировка циклов в блоки. Операции сравнения.
1.3	Кейс 1. «Угадай число». Примеры на языке Python с по угадыванию чисел, метод дихотомии.	Теория: алгоритмы поиска числа в массиве. Варианты сортировок. Поиск дихотомией. Работа с переменными, работа с функциями. Практика: упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов.
1.4	Кейс 1. «Угадай число». Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы.	Теория: создание удобной и понятной презентации. Практика: подготовка презентации для защиты. Подготовка речи для защиты.

2.	Программирование GUI на Python.	
2.1	<p>Кейс 2. «Спаси остров» Работа на языке Python со словарями и списками, множественное присваивание, добавление элементов в список и их удаление.</p>	<p>Теория: знакомство с кейсом, представление поставленной проблемы.</p> <p>Доступ к элементам по индексам. Получение слова из словаря. Отображение игрового поля игрока. Получение предположений игрока. Проверка допустимости предположений игрока.</p> <p>Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения. Создание прототипа программы. Отработка методик.</p>
2.2	<p>Планирование дизайна и механики игры. Создание главного меню игры, подсчёта очков.</p>	<p>Теория: понятие «механика игры», ограничения, правила.</p> <p>Практика: упражнения. Проверка наличия буквы в секретном слове. Проверка — не победил ли игрок. Обработка ошибочных предположений. Проверка — не проиграл ли игрок. Завершение или перезагрузка игры. Создание главного меню игры, реализация подсчёта очков.</p>
2.3	<p>Визуализация программы в виде блок-схемы.</p>	<p>Теория: проектирование проекта с помощью блок-схем.</p> <p>Практика: создание блок-схем. Ветвление в блок-схемах. Заканчиваем или начинаем игру с начала. Следующая попытка. Обратная связь с игроком.</p>
2.4	<p>Тестирование написанной программы и доработка.</p>	<p>Практика: тестирование созданной игры-программы, доработка и расширение возможностей.</p>

2.5	Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы.	Практика: подготовка презентации и речи для защиты. Презентация созданной программы.
2.6	Кейс «Калькулятор» Оформление проектной идеи. Формирование программы работ.	Теория: знакомство с кейсом, представление поставленной проблемы. Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения.
2.7	Программа для работы калькулятора	Практика: написание программы для будущего калькулятора.
2.8	Создание внешнего вида калькулятора.	Практика: создание внешнего вида калькулятора.
2.9	Тестирование написанной программы и доработка.	Практика: тестирование созданной программы, доработка и расширение возможностей.
2.11	Подготовка к публичному выступлению для защиты результатов.	Практика: подготовка презентации и речи для защиты.
2.12	Демонстрация результатов работы.	Практика: презентация созданной программы.
3.	Основы программирования автономных квадрокоптеров	
3.1	Техника безопасности при полётах. Проведение полётов в ручном режиме.	Теория: знакомство с кейсом, представление поставленной проблемы, правила техники безопасности. Практика: полёты на квадрокоптерах в ручном

		режиме.
3.2	Программирование взлёта и посадки беспилотного летательного аппарата.	Теория: основы программирования квадрокоптеров. Практика: тестирование в режимах взлёта и посадки.
3.3	Выполнение команд «разворот», «изменение высоты», «изменение позиции».	Теория: теоретические основы выполнения разворота, изменения высоты и позиции на квадрокоптерах. Практика: тестирование в режимах разворота, изменения высоты и позиции.
3.4	Выполнение группового полёта вручную.	Практика: выполнение группового полёта на квадрокоптере в ручном режиме.
4.	Программирование комплексных операций автономных квадрокоптеров	
4.1	Выполнение позиционирования по меткам.	Теория: основы позиционирования квадрокоптеров. Практика: тестирование режима позиционирования по меткам.
4.2	Программирование группового полёта.	Теория: основы группового полёта квадрокоптеров. Практика: программирование роя квадрокоптеров для группового полёта.

4.3	Программирование роевого взаимодействия.	Теория: основы программирования роя квадрокоптеров. Практика: программирование квадрокоптеров.
-----	--	---

Кейс 1. «Угадай число»

При решении данного кейса обучающиеся осваивают основы программирования на языке Python посредством создания игры, в которой пользователь угадывает число, заданное компьютером.

Программа затрагивает много ключевых моментов программирования: конвертирование типов данных, запись и чтение файлов, использование алгоритма деления отрезка пополам, обработка полученных данных и представление их в виде графиков.

Кейс 2. «Спаси остров»

Кейс позволяет обучающимся поработать на языке Python со словарями и списками; изучить, как делать множественное присваивание, добавление элементов в список и их удаление, создать уникальный дизайн будущей игры.

Кейс 3. «Калькулятор»

При решении данного кейса учащиеся создают первое простое приложение калькулятор: выполняют программную часть на языке программирования Python и создают интерфейс для пользователя при помощи библиотеки Tkinter.

Кейс 4. Программирование автономных квадрокоптеров

Роевое взаимодействие роботов является актуальной задачей в современной робототехнике. Квадрокоптеры можно считать летающей робототехникой. Шоу квадрокоптеров, выполнение задания боевыми беспилотными летательными аппаратами - такие задачи решаются с помощью применения алгоритмов роевого взаимодействия.

Данный кейс посвящен созданию шоу коптеров из 3х бпла выполняющих полет в автономном режиме. Обучающиеся получают первые навыки программирования технической системы на языке Python. Познакомятся с алгоритмами позиционирования устройств на улице и в помещении, а также узнают о принципах работы оптического распознавания объектов.

Группа	Количество учащихся в группе	Количество занятий	
		в неделю	в год
1	8-15 чел	1 занятие по 3 ч	34 занятий 102 ч
2	8-15 чел	1 занятие по 3 ч	34 занятий 102 ч
3	8-15 чел	1 занятие по 3 ч	34 занятий 102 ч
Итого:	3 группы	Общее количество часов в год 306 ч	

Тематическое планирование

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Блок 1. Введение в программирование				
1.1	Введение в образовательную программу, техника безопасности	1	1		Тестирование
1.2	Основы языка Python. Примеры на языке Python с разбором конструкций: ввод и вывод данных, условия, циклы, ветвления, массивы	18	9	9	Тестирование
1.3	Кейс 1. «Угадай число». Примеры на языке Python с поугадыванию чисел, метод дихотомии.	18	8	10	
1.4	Кейс 1. «Угадай число». Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы	18	8	10	Демонстрация решений кейса

2.	Блок 2. Tkinter. Программирование GUI на Python.				
2.1	Кейс 2. «Спаси остров» Работа на языке Python со словарями и списками, множественное присваивание, добавление элементов в список и их удаление	18	9	9	
2.2	Планирование дизайна и механики игры. Создание главного меню игры, подсчёта очков	18	8	10	
2.3	Визуализация программы в виде блок-схемы	18		18	
2.4	Тестирование написанной программы и доработка. Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы	18	9	9	Демонстрация решений кейса
2.5	Кейс «Калькулятор». Постановка проблемы, генерация путей решения	18	8	10	
2.6	Создание простейшего калькулятора с помощью библиотеки Tkinter	18		18	
2.7	Тестирование написанной программы и доработка	18		18	
2.8	Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы	18	10	9	Демонстрация решений кейса
3.	Блок 3. Основы программирования автономных квадрокоптеров				

3.1	Техника безопасности при полётах. Проведение полётов в ручном режиме	1	1		
3.2	Программирование взлёта и посадки беспилотного летательного аппарата	18	7	11	
3.3	Выполнение команд «разворот», «изменение высоты», «изменение позиции»	18	7	11	
3.4	Выполнение группового полёта вручную	18		18	
4.	Блок 4. Программирование комплексных операций автономных квадрокоптеров				
4.1	Выполнение позиционирования по меткам	17	7	10	
4.2	Программирование группового полёта	17	7	10	
4.3	Программирование роевого взаимодействия	18	7	11	
	Итого:	306 ч.	106 ч.	200 ч.	

Условия реализации программы

Календарный график

год	Дата начала занятий	Дата окончания занятий	Количество недель	Количество учебных дней	Режим занятий
2021-2022	1.09.2021	31.05.2022	34	170	1 группа вторник - 3 ч 2 группа среда – 3 ч 3 группа пятница- 3 ч

Материально-технические условия реализации программы

Аппаратное и техническое обеспечение:

- Рабочее место обучающегося:

ноутбук: производительность процессора (по тесту PassMark - CPU BenchMark<http://www.cpubenchmark.net/>): не менее 2000 единиц; объем оперативной памяти: не менее 4 Гб; объем накопителя SSD/eMMC: не менее 128 Гб (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками).

- рабочее место преподавателя:

ноутбук: процессор Intel Core i5-4590/AMD FX 8350 аналогичная или более новая модель, графический процессор NVIDIA GeForce GTX 970, AMD Radeon R9 290 аналогичная или более новая модель, объем оперативной памяти: не менее 4 Гб, видеовыход HDMI 1.4, DisplayPort 1.2 или более новая модель (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками);

- компьютеры должны быть подключены к единой сети Wi-Fi с доступом в интернет;
- презентационное оборудование (проектор с экраном) с возможностью подключения к компьютеру — 1 комплект;
- флипчарт с комплектом листов/маркерная доска, соответствующий набор письменных принадлежностей — 1 шт.;
- квадрокоптер DJI Ryze tello — не менее 3 шт.;
- поле меток;
- Wi-Fi роутер.

Программное обеспечение:

- компилятор Python 3.5;
- веб-браузер;
- пакет офисного ПО;
- текстовый редактор.

Кадровые условия реализации программы

Комплектование образовательной организации педагогическими, руководящими и иными работниками, соответствующими квалификационным характеристикам по соответствующей должности.

Формы аттестации (контроля)

Входящий 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 конце I полугодия учебного года.

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.

Оценочные материалы

Входящий контроль: Тестирование

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

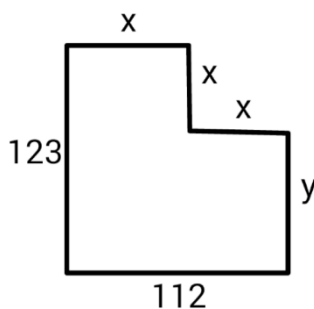
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 14-17 лет

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.

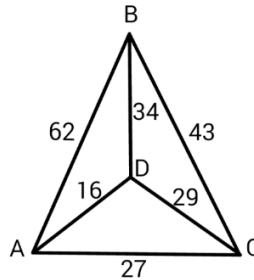
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

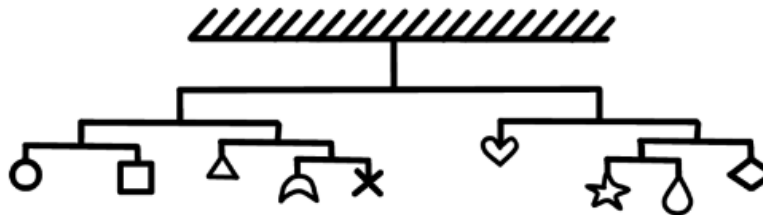
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

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.

Да

Нет

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?

Промежуточный контроль: Тестирование, решение задач

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:

11111·11111111

— произведение чисел 11111 (5 единиц) и 11111111 (7 единиц)

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

Составьте и запишите выражение для вычисления:

2014.0^14(Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений *a* и *b*

Учитывайте регистр символов при ответе.

a = True

b = False

a and b or not a and not b

Отметьте выражения, значения которых равны True:

- "239" < "30" and 239 < 30
- "239" < "30" and 239 > 30
- "239" > "30" and 239 < 30
- "239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной *i* после выполнения фрагмента программы?

```
i = 0  
while i <= 10:  
    i = i + 1  
    if i > 7:  
        i = i + 2
```

Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0  
while i <= 10:  
    i = i + 1  
    if i > 7:  
        i = i + 2
```

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

```
i = 0  
while i < 5:  
    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 ... (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 примере (**здесь $n=5$**)

Итоговый контроль: учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.

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.

Методические материалы

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2021 года.

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

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.

Учебно-методическое обеспечение курса.

Литература.

1. Бреннан, К. Креативное программирование / К. Бреннан, К. Болкх, М. Чунг. — Гарвардская Высшая школа образования, 2017.
2. Лутц, М. Программирование на Python. Т.1 / М. Лутц. — М.: Символ, 2016. — 992с.
3. Лутц, М. Программирование на Python. Т. 2 / М. Лутц. — М.: Символ, 2016. — 992с.
4. Понфиленок, О.В. Клевер. Конструирование и программирование квадрокоптеров / О.В. Понфиленок, А.И. Шлыков, А.А. Коригодский. — Москва, 2016.
5. Бриггс, Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс. — МИФ. Детство, 2018. — 320с.
6. <https://github.com/dji-sdk/Tello-Python>.
7. <https://dl-cdn.ryzero.com/downloads/tello/0222/Tello+Scratch+Readme.pdf>.