

**муниципальное автономное общеобразовательное учреждение
города Ростова-на-Дону «Школа № 60 имени пятого гвардейского Донского казачьего
кавалерийского Краснознаменного Будапештского корпуса»
(МАОУ «Школа № 60»)**

СОГЛАСОВАНО

Протокол заседания
Методического совета
МАОУ «Школа № 60»
от 30.08.2023 № 1
_____ Чубарьян Г.З.

СОГЛАСОВАНО

Заместитель директора
МАОУ «Школа № 60»
_____ Казачкова О.И.
_____ 2023 г.

УТВЕРЖДАЮ

Директор МАОУ «Школа № 60»
_____ Вихтоденко А.В.
Приказ от 30.08.2023 № 350

Директор
Вихтоденко Александр Владимирович
Подписано: 2023-08-30
Программный ключ:
089B82F27E67C7CF5CF45A1C9A10425F



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

«Основы программирования на Python»

Срок реализации: 1 год.

Возраст обучающихся: 11– 15 лет.

г. Ростов-на-Дону
2023 г.

СОДЕРЖАНИЕ

I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3 стр.
II. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	10 стр.
III. СОДЕРЖАНИЕ ПРОГРАММЫ	12 стр.
IV. МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ	16 стр.
V. СПИСОК ЛИТЕРАТУРЫ	19 стр.
VI. ПРИЛОЖЕНИЕ	21 стр.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В связи с растущим интересом к it-технологиям актуально ведение образовательного процесса в этой сфере. Дополнительная общеобразовательная (общеразвивающая) программа «Основы программирования на Python» (далее - программа), является **технической направленности** и предназначена для использования в системе дополнительного образования детей. Ее содержание ориентирована на знакомство с основными понятиями программирования, решение большого количества творческих задач, многие из которых моделируют процессы и явления из повседневной жизни и из таких предметных областей, как информатика, алгебра, геометрия, география, физика и др.

Программа содержит профориентационную работу с учащимися к профессии программиста. Программирование способствует развитию мышления, логики, коммуникативных способностей, развивает навыки взаимодействия, самостоятельности при принятии решений, раскрывает творческий потенциал. В процессе обучения у детей формируются и развиваются универсальные учебные действия, не связанные с конкретной предметной областью (soft-компетенции), развивается учебная мотивация, развивается у учащихся логическое мышление, конструкторские способности в процессе проектной работы, формирование у них знаний, умений и навыков в области программирования начального уровня (hard-компетенции).

Содержание дополнительной общеобразовательной программы «Основы программирования на Python» разработано с учетом возрастных психофизиологических особенностей обучающихся подросткового и юношеского возраста.

В социальном плане подростковая фаза – продолжение первичной социализации. Этот возрастной период отличается проявлением интереса к определенным областям знаний. Растущая заинтересованность обучающихся позволяет успешно осваивать новые научные категории, оперировать информацией, решать задачи с использованием высокоточного инновационного оборудования и специализированных компьютерных программ.

Особенности подросткового возраста были учтены при разработке содержания программы и технологий её реализации.

Актуальность программы

Несмотря на то, что программирование преподается в школах, курс программирования в дополнительном образовании предполагает отойти от обязательной программы школьного курса и предложить обучающимся курс программирования, в котором теоретические знания вкладываются в практические разработки и готовые программные продукты. Таким образом, учащимся предлагается возможность применить полученные знания на практике. В данной программе большое внимание уделяется практической работе на компьютере, самостоятельному написанию кода.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке программирования. В рамках этой программы для обучения был выбран язык

Python. Этот выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, обучающиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, являются фундаментом для дальнейшего совершенствования мастерства программирования, и, кроме этого, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам.

Педагогическая целесообразность в основу ДОП заложены принципы модульности и практической направленности. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности обучающихся.

Вид программы: модифицированная.

Уровень освоения: базовый.

При ее разработке использовались материалы таких дополнительных общеобразовательных программ как, «"Поколение Python": курс для начинающих», автор Гуев Т., «"Поколение Python": курс для продвинутых», автор Гуев Т., «Основы программирования на Python», авторы Ботов М.А., Умнов А.В., Соболева М.В. Паволоцкий А.В., канд. пед. н. 2019 г., (Екатеринбург); «Основы программирования на Python», авторы Тарасюк В.Л. и Хайруллин Д.И.. 2019 г. (Казань); «Основы программирования на Python», автор Александрова Ю.Р., 2019 г. (Вологда).

Отличительная особенность программы состоит в том, что она позволяет привлечь детей среднего школьного возраста к изучению программирования при помощи языка Python, так как он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разных программ, от текстовых процессоров до веб-браузеров;
- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python совсем не сложно;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;

- Python используется как язык программирования крупными корпорациями, такими как Google.

Новизна и практическая значимость данной образовательной программы заключается в создании такой методики изучения учащимися современных технологий программирования, которая даст почву для самообразования и практической, исследовательской, самостоятельной научной деятельности. Обучение нацелено на раннее выявление и становление талантливых детей как через приобретение знаний и умений, так и через развитие творческих навыков посредством участия в творческих конкурсных состязаниях, популяризации науки, научной, изобретательской деятельности.

Обучающиеся по программе: школьники возраста от 13 до 15 лет.

Срок реализации: Общее количество часов – 116 часа в год.

Режим занятий:

Занятия проводятся в соответствии с СанПиН 2.4.3648–20 2 раза в неделю по 2 урока.

Условия набора и комплектования учебных групп: принимаются учащиеся в возрасте от 13 до 15 лет, проявляющие интерес к информационным технологиям. Представленная программа рассчитана на любой социальный статус учащихся, имеющих различные интеллектуальные, технические, творческие способности. Набор в группы осуществляется без специальной подготовки, от учащихся не требуется специальных знаний и умений.

Цели программы:

- создание условий для изучения методов программирования на языке Python;
- рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная);
- подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Задачи программы:

Обучающие:

- предоставить обучающимся базовые знания по основам программирования на языке Python;
- дать представление о среде программирования Wing IDE;
- познакомить с современным уровнем развития технических и программных средств в области программирования;
- познакомить с основными понятиями, принципами и инструментариями разработки систем в среде программирования Wing IDE;
- выработать навыки применения средств компьютерных технологий для решения поставленных задач;
- обучить методам научного познания, моделирования, компьютерного эксперимента;
- обучить навыкам моделирования поведения объектов.

Развивающие:

- способствовать развитию творческих способностей учащихся,

познавательных интересов, развитию индивидуальности и самореализации;

- развивать пространственное воображение, внимательность к деталям, ассоциативное и аналитическое мышление;
- развивать познавательные способности обучающегося, память, внимание, пространственное мышление, аккуратность и изобретательность при выполнении учебных проектов;
- формировать творческий подход к поставленной задаче;
- развивать навыки инженерного, системно-комбинаторного мышления и эффективного использования электронного вычислительного оборудования.

Воспитательные:

- развить у обучающихся чувство внутренней инициативы, самостоятельности;
- привить обучающемуся тягу к самосовершенствованию;
- воспитать мотивацию учащихся к изобретательству, созданию собственных проектов, ответственности за результат своей работы на компьютере и за возможные ошибки;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность, критичность и самокритичность мышления.

Мотивирующие:

- мотивировать обучающихся к нестандартному мышлению, изобретательству и инициативности при выполнении задач;
- поощрять у учащихся мотивацию к работе в формате «от идеи до законченного проекта» на всех этапах деятельности;
- поощрять инициативу обучающихся предлагающих нестандартное решение задач и их реализацию;
- поддерживать стремление к самостоятельному повышению уровня навыков программирования, моделирования и визуализации, необходимых для поддержания конкурентоспособности специалиста в современном высокотехнологичном мире.

Планируемые результаты освоения программы

В результате освоения программы, обучающиеся будут *знать*:

- основные классические алгоритмы и способы их реализации;
- синтаксис языка программирования Python;
- простые и сложные структуры данных, а также конструкции для работы с ними;
- некоторые модули (turtle, random, tkinter и др.).
- основные элементы программирования: ввод (вывод) информации в память (из памяти), данные, операции с данными, условное выполнение, циклы, подпрограммы

Личностные результаты:

- сформированность способности к саморазвитию;

- готовность к конструктивному общению и взаимодействию, урегулированию конфликтов в условиях работы в команде при реализации проектов;

- готовность к профессиональному самоопределению.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;

- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;

- умение критически оценивать правильность решения учебно-исследовательской задачи;

- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

- владение основами самоконтроля, способность к принятию решений;

- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);

- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Познавательные универсальные учебные действия:

- разбивать решение задачи на подзадачи, то есть составлять алгоритм;

- объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;

- искать и обрабатывать ошибки в коде;

- писать грамотный, красивый код;

- анализировать как свой, так и чужой код;

- импортировать модули в программу;

- работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);

- грамотно строить коммуникацию, исходя из целей и ситуации;

- работать с вычислительной техникой.

Коммуникативные универсальные учебные действия:

- умение самостоятельно организовывать целенаправленное учебное взаимодействие в группе;

- способность выражать собственное мнение, отстаивать свою точку зрения, приводить аргументы, подтверждая их фактами;

- умение учитывать разные мнения, сравнивать разные точки зрения;

- умение осуществлять взаимный контроль и оказывать в сотрудничестве необходимую взаимопомощь;

- умение принимать критику к своей работе.

Предметные результаты:

- написать программу на языке Python;
- искать информацию в свободных источниках и структурировать ее;
- комбинировать, видоизменять и улучшать идеи;
- грамотно в письменной форме формулировать свои мысли;
- проверять и изменять свойства решение задачи;
- оперировать основными терминами, используемыми при разработке приложений;
- следовать требованиям;
- использовать приобретенные навыки работы для создания собственных и групповых проектов;
- проводить тестирования для выявления ошибок;
- организовывать индивидуальное информационное пространство.
- умение осуществлять поиск ошибок программного кода производить отладку составленных программ;
- самостоятельная разработка и презентация проектов.

Формы обучения и виды занятий

Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием обучающей on-line платформы или интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует

интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/контроля
		Теория	Практика	Всего	
	Входная педагогическая диагностика		2	2	
Модуль 1. Введение. Знакомство с Python		3	4	7	
1	Введение в программирование, общие сведения о Python, знакомство с on-line платформой курса.	1	1	2	Беседа, практические задания
2	Ввод-вывод данных, параметры <code>sep</code> и <code>end</code> , переменные, <code>PEP</code> .	1	1	2	Беседа, практические задания
3	Работа с целыми числами, операции с числами, преобразование типов.	1	1	2	Беседа, практические задания
4	Итоговая работа на ввод-вывод данных.		1	1	Практическое занятие
Модуль 2. Базовые конструкции в Python		9	19	28	
1	Условный оператор.	3	4	7	Беседа, практические задания
2	Итоговая работа на условный оператор.		2	2	Практическое занятие
3	Числовые и строковые типы данных, модуль <code>math</code> .	2	3	5	Беседа, практические задания
4	Циклы.	4	8	12	Беседа, практические задания
5	Итоговая работа на циклы.		2	2	Практическое занятие

Модуль 3. Использование в Python строк, списков и функций		7	21	28	
1	Строковый тип данных.	3	3	6	Беседа, практические задания
2	Итоговая работа на строки.		2	2	Практическое занятие
3	Списки.	1	6	7	Беседа, практические задания
4	Итоговая работа на списки.		2	2	Практическое занятие
5	Функции.	2	3	5	Беседа, практические задания
6	Итоговая работа на функции.		2	2	Практическое занятие
7	Работа над мини-проектом, модуль random.	1	3	4	Беседа, выполнение мини-проекта
Модуль 4. Использование в Python более сложных конструкций		8	41	49	
1	Повторение основных конструкций языка.		8	8	Практические задания
2	Типы данных bool и NoneType.	1		1	Беседа
3	Вложенные списки, матрицы.	2	15	17	Беседа, практические задания
4	Итоговая работа на вложенные списки и матрицы.		3	3	Практическое занятие
5	Кортежи.	1	2	3	Беседа, практические задания
6	Итоговая работа на кортежи.		2	2	Практическое занятие
7	Множества.	2	3	5	Беседа, практические задания
8	Итоговая работа на множества.		2	2	Практическое занятие
9	Словари.	2	4	6	Беседа, практические задания
10	Итоговая работа на словари.		2	2	Практическое занятие
	Итоговая педагогическая диагностика		2	2	Беседа
Итого		31	85	116	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

Входная педагогическая диагностика (2 ч.)

Практика: Стартовая педагогическая диагностика. Тест на определение входного уровня знаний обучающихся.

Модуль 1. Введение. Знакомство с Python (Всего 7 ч. Теория – 3 часа, практика – 4 часов).

Тема 1. Введение в программирование, общие сведения о Python, знакомство с on-line платформой курса

Теория: компьютерная программа, язык программирования, программный код, компилируемые и интерпретируемые языки программирования, машинный код,

компилятор. Язык Python, история языка Python, сильные и слабые стороны Python, Python 2 vs Python 3, философия Python, среда разработки Wing IDE. Знакомство с on-line платформой курса – содержание курса, графические обозначения в курсе, как правильно проходить курс, знакомство с встроенной в on-line платформу системой написания программного кода и системой автоматизированной проверки задач.

Практика: установка Python на компьютер, установка Wing IDE на компьютер, регистрация на on-line платформе курса и организация личного кабинета.

Тема 2. Ввод-вывод данных, параметры `sep` и `end`, переменные, PEP

Теория: вывод данных, команда `print()`. Ввод данных, команда `input()`. Необязательные параметры команды `print()` – `sep` и `end`, переменные, стандарт PEP 8, комментарии.

Практика: решение задач.

Тема 3. Работа с целыми числами, операции с числами, преобразование типов

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. Алгоритм получения цифр n -значного числа.

Практика: решение задач.

Тема 4. Итоговая работа на ввод-вывод данных

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Модуль 2. Базовые конструкции в Python (Всего 28 часа. Теория – 9 часов, Практика – 28 часа).

Тема 1. Условный оператор

Теория: условные операторы `if` и `if-else`, отступы, операторы сравнения, цепочки сравнений, транзитивность. Логические операторы: `and`, `or`, `not`. Приоритеты логических операторов. Вложенный оператор. Каскадный условный оператор - `if-elif-else`.

Практика: решение задач.

Тема 2. Итоговая работа на условный оператор

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 3. Числовые и строковые типы данных, модуль `math`

Теория: Целочисленный тип данных: тип данных `int`, преобразование строки в целое число, целочисленные операторы, длинная арифметика, символ разделитель. Числа с плавающей точкой: тип данных `float`, преобразование строки к числу с плавающей точкой, арифметические операторы. Преобразование между `int` и `float`. Встроенные функции: `max()`, `min()`, `abs()`. Строковый тип данных: тип данных `str`, функции `len()` и `str()`, конкатенация строк, умножение строки на число, оператор `in`. Модуль `math`: особенности подключения модулей, список функций модуля `math`, список констант модуля `math`.

Практика: решение задач.

Тема 4. Циклы

Теория: Цикл for: Примеры использования цикла for, переменная цикла, имена переменных цикла, функции range(), отрицательный шаг генерации. Частые сценарии: подсчет количества, вычисление суммы и произведения, обмен значений переменных, сигнальные метки, определение максимума и минимума, расширенные операторы присваивания. Цикл while: цикл for vs цикл while, считывание данных до стоп значения, бесконечный цикл. Использование цикла while для обработки цифр числа. Оператор break, оператор continue, бесконечные циклы, ключевое слово else в циклах. Поиск ошибок и ревью кода. Вложенные циклы, операторы break и continue во вложенных циклах.

Практика: решение задач.

Тема 5. Итоговая работа на циклы

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Модуль 3. Использование в Python строк, списков и функций. (Всего 28 часов. Теория – 7 часов, Практика – 21 часов).

Тема 1. Строковый тип данных

Теория: Строки: индексация строк, итерирование строк. Срезы строк, изменение символов строки. Методы строк. Представление строк в памяти компьютера, таблица символов ASCII, таблица символов Unicode, функция ord(), функция chr().

Практика: решение задач.

Тема 2. Итоговая работа на строки

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 3. Списки

Теория: Введение в списки: создание списков, пустые списки, встроенная функция list(), вывод списков. Основы работы со списками: функции len(), sum(), min(), max(); оператор принадлежности in; индексация и срезы; конкатенация и умножение на число; отличие списков от строк. Методы добавления и удаления элементов. Вывод элементов списка: с помощью цикла for, с помощью распаковки списка. Строковые методы split() и join(). Методы списков: insert(), index(), remove(), pop(), reverse(), count(), clear(), copy(), sort(). Списочные выражения. Сортировка списков: пузырьком, выбором, простыми вставками.

Практика: решение задач.

Тема 4. Итоговая работа на списки

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 5. Функции

Теория: Пользовательские функции: функции без параметров, преимущества использования функций, ключевое слово pass, функции с параметрами, область видимости параметрической переменной, параметры по умолчанию. Локальные и глобальные переменные: локальные переменные, область действия локальной переменной, глобальные переменные, глобальные константы. Функции с возвратом значения.

Практика: решение задач.

Тема 6. Итоговая работа на функции

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 7. Работа над мини-проектом, модуль random

Теория: Модуль random: случайные числа, модуль random, функция shuffle(), функция choice(), функция sample(). Проекты: Угадайка чисел, Магический шар 8, Генератор безопасных паролей, Шифр Цезаря, Калькулятор систем счисления, Угадайка слов.

Практика: разработка проекта.

Модуль 4. Использование в Python более сложных конструкций. (Всего 49 часов. Теория – 8 часов, Практика – 41 часов).

Тема 1. Повторение основных конструкций языка

Практика: решение задач.

Тема 2. Типы данных bool и NoneType

Теория: тип данных bool: логический тип данных, логические операторы, булевы значения как числа, функции bool(), type(), isinstance(). Тип данных NoneType: пустое значение, литерал None, сравнение None с другими типами данных.

Тема 3. Вложенные списки, матрицы

Теория: Вложенные списки: объявление и индексация; функции len(), max(), min(); списочные методы; создание вложенных списков; считывание вложенных списков; перебор элементов вложенных списков; обработка вложенных списков; вывод вложенных списков. Матрицы: работа с матрицами; квадратные и прямоугольные матрицы; функции ljust() и rjust(); главная и побочная диагонали. Операции над матрицами: сложение матриц; умножение матрицы на число; умножение матриц.

Практика: решение задач.

Тема 4. Итоговая работа на вложенные списки и матрицы

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 5. Кортежи

Теория: Введение в кортежи: тип данных tuple, особенности работы с кортежами. Основы работы с кортежами: функция tuple(), особенности кортежей, методы кортежей, вложенные кортежи, перебор кортежей, сравнение кортежей, сортировка кортежей, преобразование кортежа в список и строку, упаковка кортежей, распаковка кортежей, присваивание кортежей.

Практика: решение задач.

Тема 6. Итоговая работа на кортежи

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 7. Множества

Теория: Множества: множества в математике, числовые множества, операции над множествами, диаграммы Эйлера-Вена. Множества в Python: создание множеств, пустые множества, встроенная функция set(), вывод множеств, функции len(), sum(), min(), max(), оператор принадлежности in, перебор

элементов множеств, форматированный вывод множеств, сравнение множеств. Методы множеств: метод добавления элемента `add()`; методы удаления элементов `remove()`, `discard()`, `pop()`; метод удаления всех элементов `clear()`; методы `union()`, `intersection()`, `difference()`, `symmetric_difference()`; методы `update()`, `intersection_update()`, `difference_update()`, `symmetric_difference_update()`; операторы `&`, `|`, `-`, `^`; методы `issuperset()`, `issubset()`, `isdisjoint()`; сравнение множеств (`<`, `>`, `<=`, `>=`). Генераторы множеств и `frozenset`. *Практика*: решение задач.

Тема 8. Итоговая работа на множества

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Тема 9. Словари

Теория: Введение в словари в Python: новый тип коллекции, отличия словарей от списков, создание словарей, обращение по ключу, встроенная функция `dict()`, создание словарей на основе списков и кортежей, пустой словарь, вывод словаря, особенности словарей. Основы работы со словарями: встроенные функции `len()`, `sum()`, `min()`, `max()`; оператор принадлежности `in`; перебор элементов словарей; распаковка словаря; форматированный вывод словарей; сравнение словарей; методы `keys()`, `values()` и `items()`. Методы словарей: добавление и изменение элементов в словаре; удаление элементов из словаря; методы `get()`, `update()`; методы `pop()`, `popitem()`; методы `clear()`, `copy()`; метод `setdefault()`. Вложенные словари и генераторы словарей.

Практика: решение задач.

Тема 8. Итоговая работа на словари.

Практика: решение задач контрольной работы по пройденной теме. Анализ результатов.

Итоговая педагогическая диагностика.

Практика: Итоговая педагогическая диагностика. Тест на определение уровня освоения знаний обучающихся.

4. МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python» разработана с учетом современных требований к организации образовательного процесса.

Программа основана на постулатах личностно-ориентированного образования, особенностью которого является актуализация гуманитарных, человекообразующих функций обучения: сохранение и восстановление телесного и духовного здоровья, личной свободы, нравственности; гуманный подход направлен не на формирование ребенка, а на обеспечение поддержки в его развитии, создание механизмов саморегуляции, саморазвития, адаптации, самозащиты, самовоспитания личности; акцентирование внимания не на вооружении ребенка знаниями и умениями, а его становлении, обретении им неповторимой индивидуальности, духовности, творческого начала; построение учебного процесса осуществляется на основе образовательных технологий,

основными признаками которых является диалогичность, деятельностный творческий характер, поддержка индивидуального развития ребенка, предоставление ему необходимого пространства свободы, творчества и др.

При проектировании программы учтены следующие педагогические принципы:

- *научности и доступности обучения;*
- *упорядоченности педагогического процесса* (целостности, систематичности, последовательности), что означает достижение единства и взаимосвязи между всеми компонентами педагогического процесса;
- *природосообразности*, т.е. образовательный процесс проектируется с учетом возрастных, гендерных и индивидуальных особенностей обучающихся;
- *гуманизации* - педагогический процесс строится на полном признании гражданских прав ребенка и уважения к нему;
- *демократизации* - предоставление всем участникам образовательного процесса свободы для саморазвития, саморегуляции, самоопределения, самообучения, самовоспитания;
- *культуросообразности* - максимальное использование в образовательном процессе той культурной среды, в которой находится МБУ ДО ДТДМ;
- *принципа единства и непротиворечивости действий учебного заведения и образа жизни учащихся*, что способствует установлению взаимосвязей между сферами жизни обучающихся;
- *принцип дифференциации* реализуется на всех годах обучения посредством индивидуальных дифференцированных заданий и упражнений.

Программа построена так же на принципах *погружения в среду визуального программирования*. Обучающиеся вначале изучают основы алгоритмизации и программирования с помощью специального on-line тренажёра, затем, на практических примерах, разбирая реальные существующие задачи, выполняя на каждом занятии учебные проекты, и, наконец, получают практический опыт командной работы над итоговым проектом-приложением с последующей «публичной» защитой проекта.

В процессе реализации программы используются **педагогические технологии** личностно-ориентированного образования, сотрудничества, индивидуального, дифференцированного, компетентностного подходов, проектно-исследовательская, проблемного обучения, рефлексивная, и др.

В целях обеспечения целостного характера педагогической деятельности при реализации данной образовательной программы, предусмотрены мероприятия психолого-педагогического сопровождения: диагностические исследования совместно с педагогами-психологами МБУ ДО ДТДМ.

Таким образом, основными компонентами образовательного процесса при реализации данной образовательной программы являются:

- практический компонент, направленный на формирование у учащихся практических знаний и умений;
- компонент самостоятельности, пронизывающий все сферы и виды деятельности, способствующий воспитанию самостоятельной личности;

- творческий компонент, направленный на развитие у учащихся творческих компетенций и творческого подхода к своей деятельности;
- исследовательский, направленный на формирование научного мировоззрения, исследовательского отношения к окружающему миру, развивающий способности к анализу, альтернативному мышлению в выборе различных способов деятельности.

Педагогические методы и приемы, применяемые при освоении данной программы, способствуют комплексному решению педагогических задач по созданию творческой, развивающей среды для обучающихся.

Метод проектов обеспечивает вариативность учебного процесса с учетом уровня подготовки, интересов обучающихся и предполагает решение проблемы, предусматривающей, с одной стороны, использование разнообразных методов, средств обучения, а с другой - интегрирование знаний, умений из различных областей науки, техники, технологии, творческих областей. Разработка технологической карты, составление технического паспорта, сборка и построение алгоритма поведения модели позволяет учащимся самостоятельно освоить целый набор знаний из разных областей, в том числе робототехники, электроники, механики, программирования, что способствует повышению интереса к быстроразвивающейся науке робототехнике.

Каждый проект осуществляется под руководством педагога, который оказывает помощь в определении темы и разработке структуры проекта, дает рекомендации по подготовке, выбору средств проектирования, обсуждает этапы его реализации. Роль педагога сводится к оказанию методической помощи, а каждый обучающийся учится работать самостоятельно, получать новые знания и использовать уже имеющиеся, творчески подходить к выполнению заданий и представлять свои работы.

Итоговые работы представляются на выставках технического творчества, что дает возможность учащимся оценить значимость своей деятельности, услышать и проанализировать отзывы со стороны сверстников и взрослых.

Мониторинг образовательных результатов осуществляется посредством педагогического наблюдения в процессе выполнения обучающимися практических заданий. (Приложение 1).

Диагностики личностного развития и уровня развития творческих способностей обучающихся проводится педагогом-психологом в начале и в конце освоения программы.

Предусматриваются различные формы подведения итогов реализации образовательной программы:

- выставка,
- соревнование,
- внутригрупповой конкурс,
- участие в олимпиадах, соревнованиях,
- учебно-исследовательских конференциях,
- презентация проектов обучающихся.

Формы работы с родителями: консультации по вопросам развития и обучения ребенка, родительские собрания, детско-родительские занятия творческого характера.

Форма занятий	Приёмы и методы организации образовательного процесса	Дидактический материал. Электронные источники	Техническое оснащение и расходный материал	Форма подведения итогов
Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	1. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com); 2. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru); 3. Официальная документация языка Python (http://docs.python.org).	- Ноутбуки с мышкой и доступом к сети Интернет. Презентационное оборудование. - Принтер. - Маркерная доска.	Итоговая работа.

5. СПИСОК ЛИТЕРАТУРЫ.

Нормативно-правовые документы:

1. Государственная программа Российской Федерации «Развитие образования» на 2019-2025 г, утвержденная Постановлением Правительства Российской Федерации от 26 декабря 2017 года № 1642 (ред. от 15.03.2021).
2. Приказ Министерства просвещения Российской Федерации №467 от 03.09.2019 г. «Об утверждении Целевой модели развития региональных систем дополнительного образования».
3. Распоряжение Министерства просвещения Российской Федерации №Р-126 от 21.06.2021 г. «Об утверждении ведомственной целевой программы «Развитие дополнительного образования детей, выявление и поддержка лиц, проявивших выдающиеся способности».
4. Государственная программа Ростовской области «Развитие образования», утверждена постановлением Правительства Ростовской области от 17.10.2018 № 646 (с изменениями на 28 декабря 2020 года).
5. Конвенция о правах ребенка (принята резолюцией 44/25 Генеральной Ассамблеи от 20 ноября 1989 г.) — URL: http://www.un.org/ru/documents/decl_conv/conventions/childcon.shtml.
6. Концепция развития дополнительного образования детей до 2030 года, утвержденная распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.
7. Национальный проект «Образование», утвержденный на заседании президиума Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам (протокол от 24 декабря 2018 г. № 16).

8. Постановление Правительства Российской Федерации от 31 октября 2018 г. № 1288 (ред. от 10.07.2020, № 1019) «Об организации проектной деятельности в Правительстве Российской Федерации».
9. Приказ Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196 (ред. от 30.09.2020 г.) «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
10. Приказ Министерства просвещения Российской Федерации от 30 сентября 2020 г. № 533 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденный приказом Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196».
11. Приказ Министерства труда и социальной защиты Российской Федерации от 22 сентября 2021 г. N 652н н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».
12. Распоряжение Правительства Российской Федерации от 23.01.2021г. № 122-р «Об утверждении Плана основных мероприятий, проводимых в рамках Десятилетия детства, на период до 2027 года.
13. СанПиН 2.4.3648–20 «Санитарно-эпидемиологические требования к организациям воспитания, обучения, отдыха и оздоровления детей и молодежи», Утверждены постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 (зарегистрировано Минюстом России 18.12.2020, регистрационный № 61573).
14. Стратегическая инициатива «Новая модель системы дополнительного образования», одобренная Президентом Российской Федерации 27 мая 2015 г
15. Стратегия развития воспитания в Российской Федерации на период до 2025 года, утвержденная Распоряжением Правительства Российской Федерации от 29 мая 2015 г. № 996-р.
16. Указ Президента Российской Федерации от 29 мая 2017 г. № 240 «Об объявлении в Российской Федерации Десятилетия детства».
17. Указ Президента Российской Федерации от 21 июля 2020 г. № 474 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2030 года».
18. Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 24.03.2021) «Об образовании в Российской Федерации».
19. Федеральный проект «Успех каждого ребенка», утвержденный президиумом Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам (протокол от 3 сентября 2018 года № 10).

Основная литература:

- К.Ю. Поляков, Е.А. Еремин. Информатика. Углубленный уровень. Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2014
- М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2011
- Задачи по программированию. Под ред. С.М. Окулова. М.: БИНОМ, Лаборатория знаний, 2006
- С.М. Окулов. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2012

Дополнительная литература:

- М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2011
- Информатика и ИКТ. Задачник – практикум в двух частях. Под ред. И.Г. Семакина и Е.К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014

Электронные ресурсы:

- Сайт stepik.org – «"Поколение Python": курс для начинающих»
- Сайт stepik.org – «"Поколение Python": курс для продвинутых»
- Сайт <https://itproger.com/test/python> - «Тест на знание основ Python»
- Сайт <https://proglib.io/tests/test-na-znanie-yazyka-python> - «Тест на знание языка Python»
- Материалы и презентации к урокам в LMS Яндекс. Лицея.
- Сайт pythonworld.ru – «Python 3 для начинающих»
- Сайт pythontutor.ru – «Питонтьютор»
- <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh5OpdwBI> – лекции А.В. Умнова, прочитанные в Школе Анализа Данных Яндекса

6. ПРИЛОЖЕНИЕ.

Диагностика эффективности образовательного процесса.

Входной контроль - имеет диагностические задачи и осуществляется в начале учебного года. Цель предварительной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью. Входной контроль может проводиться в следующих формах: творческие работы, самостоятельные работы, вопросники, тестирование и пр.

Тест для стартовой диагностики

Вопрос 1

Сколько библиотек можно импортировать в один проект?

- 1) Не более 3
- 2) Не более 5
- 3) Неограниченное количество
- 4) Не более 23
- 5) Не более 10

Вопрос 2

Что покажет этот код?

```
for j in 'Hi! I\'m mister Robert':  
    if j == '\\':
```

```
        print("Найдено")
        break
```

else:

```
        print ("Готово")
```

- 1) "Найдено" и "Готово"
- 2) "Готово"
- 3) "Найдено"
- 4) Ошибку в коде

Вопрос 3

Какая библиотека отвечает за время?

- 1) clock
- 2) localtime
- 3) time
- 4) Time

Вопрос 4

Что покажет этот код?

```
for i in range(5):
```

```
    if i % 2 == 0:
```

```
        continue
```

```
    print(i)
```

- 1) Ошибку, так как i не присвоена
- 2) Числа: 1 и 3
- 3) Числа: 1, 3 и 5
- 4) Числа: 0, 2 и 4
- 5) Ошибку из-за неверного вывода

Вопрос 5

Что будет результатом этого кода?

```
x = 23
```

```
num = 0 if x > 10 else 11
```

```
print(num)
```

- 1) 23
- 2) 10
- 3) 11
- 4) Ошибка
- 5) 0

Вопрос 6

Где правильно создана переменная?

(спрашивается про вариант ответа, который не выдаст ошибку при запуске проекта)

- 1) `int num = 2`
- 2) Нет подходящего варианта
- 3) `var num = 2`
- 4) `$num = 2`
- 5) `num = float(2)`

Вопрос 7

Как получить данные от пользователя?

- 1) Использовать метод `get()`
- 2) Использовать метод `cin()`
- 3) Использовать метод `read()`
- 4) Использовать метод `readLine()`
- 5) Использовать метод `input()`

Вопрос 8

Что будет показано в результате?

```
name = "John"
print('Hi, %s' % name)
```

- 1) "Hi, name"
- 2) Ошибка
- 3) "Hi, John"
- 4) "Hi, "

Вопрос 9

Какая функция выводит что-либо в консоль?

- 1) `print()`;
- 2) `log()`;
- 3) `out()`;
- 4) `write()`;

Вопрос 10

Какие ошибки допущены в коде ниже?

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
print(factorial(5))
```

- 1) Функция не может вызывать сама себя
- 2) В коде нет никаких ошибок
- 3) Необходимо указать тип возвращаемого значения
- 4) Функция всегда будет возвращать 1

Ответы:

- | | |
|-------|--------|
| 1) 3; | 6) 5; |
| 2) 3; | 7) 5; |
| 3) 3; | 8) 3; |
| 4) 2; | 9) 1; |
| 5) 5; | 10) 2; |

Промежуточная аттестация проводится на основании диагностики теоретических знаний и практических умений и навыков по итогам освоения модуля. Промежуточная аттестация проводится в следующих формах: защита творческих или исследовательских работ и проектов, конференции, выставочный просмотр, викторины, олимпиада, конкурс, соревнование, турнир и пр.

Итоговая аттестация проводится по окончании обучения по программе.

Возможные уровни теоретической подготовки обучающихся:

Высокий уровень – учащийся освоил практически весь объем знаний (80-100%), предусмотренных программой за конкретный период; специальные термины употребляет осознанно и в полном соответствии с их содержанием.

Средний уровень – у учащегося объем освоенных знаний составляет 50-79%; сочетает специальную терминологию с бытовой.

Низкий уровень – учащийся овладел менее чем 50% объема знаний, предусмотренных программой; учащийся, как правило, избегает употреблять специальные термины.

Возможные уровни практической подготовки обучающихся:

Высокий уровень – учащийся овладел 80-100% умениями и навыками, предусмотренными программой за конкретный период; работает с оборудованием самостоятельно, не испытывает особых трудностей; выполняет практические задания с элементами творчества.

Средний уровень – у учащегося объем усвоенных умений и навыков составляет 50-79%; работает с оборудованием с помощью педагога; в основном выполняет задания на основе образца.

Низкий уровень – учащийся овладел менее чем 50% умений и навыков, предусмотренных программой; испытывает затруднения при работе с оборудованием; обучающийся в состоянии выполнять лишь простейшие практические задания педагога.

Достигнутые обучающимися знания, умения и навыки заносятся в сводную таблицу результатов обучения.

В целях определения уровня усвоения программы учащимися осуществляются диагностические срезы:

□ входная диагностика на основе анализа выбранной обучающимися роли в диагностической игре и степени их участия в реализации отдельных ее этапов, где выясняется начальный уровень знаний, умений и навыков учащихся, а так же выявляются их творческие способности.

□ промежуточная диагностика позволяет выявить достигнутый на данном этапе уровень знаний, умений и навыков учащихся, в соответствии с реализованной проектной деятельностью. Предлагаются выполнение практических заданий, контрольные тесты.

□ итоговая диагностика проводится в конце учебного курса (выставка и защита творческих проектов) и предполагает комплексную проверку образовательных результатов по всем ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы учащимися.

Результаты контроля фиксируются в диагностической карте.

Бланк наблюдения за обучающимися

Группа _____

№	ФИО	ПОКАЗАТЕЛИ
---	-----	------------

п/п		Внимателен в течение занятия	Использует базовую систему понятий	Проявляет инициативу, интерес в течение занятия	Идет на деловое сотрудничество	Проявляет самостоятельность при изучении материала	РЕЗУЛЬТАТ
1							
2							
3							
4							
5							
6							

...

За каждое согласие с утверждением 1 – балл.

Итоговая аттестация

Система контроля знаний и умений учащихся представляется в виде учёта результатов по итогам выполнения заданий отдельных кейсов и посредством наблюдения, отслеживания динамики развития учащегося.

Итоговая аттестация учащихся осуществляется по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Баллы, набранные учащимся	Уровень освоения
0–50 баллов	Низкий
50–75 баллов	Средний
75–100 баллов	Высокий

Фонд оценочных материалов

Распределение баллов и критерии оценивания

№ п/п	Название модуля, кейса	Количество баллов	
		min	max
1.	Модуль 1. Введение. Знакомство с Python		
1.1.	Введение в программирование, общие сведения о		

	Python, знакомство с on-line платформой курса		
	Посещение занятий	1	1
	Проектная деятельность	1	1
1.2.	Ввод-вывод данных, параметры <code>sep</code> и <code>end</code> , переменные, <code>PEP</code>		
	Посещение занятий	1	2
	Проектная деятельность	1	2
1.3.	Работа с целыми числами, операции с числами, преобразование типов		
	Посещение занятий	1	2
	Проектная деятельность	1	2
1.4.	Итоговая работа на ввод-вывод данных		
	Посещение занятий	1	2
	Проектная деятельность	1	2
2.	Модуль 2. Базовые конструкции в Python		
2.1.	Условный оператор		
	Посещение занятий	1	2
	Проектная деятельность	1	2
2.2.	Итоговая работа на условный оператор		
	Посещение занятий	1	2
	Проектная деятельность	1	2
2.3.	Числовые и строковые типы данных, модуль <code>math</code>		
	Посещение занятий	1	2
	Проектная деятельность	1	2
2.4.	Циклы		
	Посещение занятий	1	2
	Проектная деятельность	1	2
2.5.	Итоговая работа на циклы		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.	Модуль 3. Использование в Python строк, списков и функций		
3.1.	Строковый тип данных		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.2.	Итоговая работа на строки		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.3.	Списки		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.4.	Итоговая работа на списки		
	Посещение занятий	1	2

	Проектная деятельность	1	2
3.5.	Функции		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.6.	Итоговая работа на функции		
	Посещение занятий	1	2
	Проектная деятельность	1	2
3.7.	Работа над мини-проектом, модуль random		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.	Модуль 4. Использование в Python более сложных конструкций		
4.1.	Повторение основных конструкций языка		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.2.	Типы данных bool и NoneType		
	Посещение занятий	1	1
	Проектная деятельность	1	1
4.3.	Вложенные списки, матрицы		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.4.	Итоговая работа на вложенные списки и матрицы		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.5.	Кортежи		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.6.	Итоговая работа на кортежи		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.7.	Множества		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.8.	Итоговая работа на множества		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.9.	Словари		
	Посещение занятий	1	2
	Проектная деятельность	1	2
4.10.	Итоговая работа на словари		
	Посещение занятий	1	2
	Проектная деятельность	1	2
	Итого:		

Тест для итоговой диагностики

Вопрос 1

Имеется кортеж вида `T = (4, 2, 3)`. Какая из операций приведёт к тому, что имя `T` будет ссылаться на кортеж `(1, 2, 3)`?

- 1) `T[0] = 1`
- 2) `T = (1) + T[1:]`
- 3) `T = (1,) + T[1:]`
- 4) `T.startswith(1)`

Вопрос 2

Для чего в Python используется встроенная функция `enumerate()`?

- 1) Для определения количества элементов последовательности.
- 2) Для одновременного итерирования по самим элементам и их индексам.
- 3) Для сортировки элементов по значениям `id`.

Вопрос 3

Что выведет интерпретатор для следующей программы?

```
def get_name_and_decades(name, age):  
    print(f"My name is {name} and I'm {age / 10:.5f} decades old.")  
get_name_and_decades("Leo", 31)
```

- 1) My name is Leo and I'm 31.00000 decades old.
- 2) My name is Leo and I'm 3.1 decades old.
- 3) Исключение: перед строкой стоит лишняя буква `f`.
- 4) My name is Leo and I'm 3.10000 decades old.
- 5) My name is {name} and I'm {age / 10:.5f} decades old.

Вопрос 4

Необходимо собрать и вывести все уникальные слова из строки рекламного текста. Какой из перечисленных типов данных Python подходит лучше всего?

- 1) кортеж (`tuple`)
- 2) список (`list`)
- 3) множество (`set`)
- 4) словарь (`dict`)

Вопрос 5

Учёт зверей в зоопарке ведётся с помощью приведённого ниже списка словарей. Какая из строчек кода выведет структуру, отсортированную в порядке увеличения возрастов животных?

```
animals = [  
    {'type': 'penguin', 'name': 'Stephanie', 'age': 8},  
    {'type': 'elephant', 'name': 'Devon', 'age': 3},  
    {'type': 'puma', 'name': 'Moe', 'age': 5},  
]
```

- 1) `sorted(animals, key='age')`
- 2) Ни один вариант не является верным, два словаря нельзя сравнивать друг с другом.

3) sorted(animals, key=lambda animal: animal['age'])

4) sorted(animals)

Вопрос 6

Какой результат выведет следующий код?

```
def f(a, *pargs, **kargs): print(a, pargs, kargs)
```

```
f(1, 2, 3, x=4, y=5)
```

1) 1, 2, 3, {'x': 4, 'y': 5}

2) 1 (2, 3) {'x': 4, 'y': 5}

3) Будет вызвано исключение, после двоеточия обязательно нужно перейти на новую строку.

4) 1, 2, 3, 'x=4', 'y=5'

5) 1, 2, 3, 4, 5

Вопрос 7

Как можно более кратко представить следующую запись?

```
if X:
```

```
    A = Y
```

```
else:
```

```
    A = Z
```

1) A = Y if Z else Y

2) A = Y if X else Z

3) A = X if Z else Y

4) A = X if Y else Z

Вопрос 8

*Какая из перечисленных инструкций выполнится быстрее всего, если $n = 10^{**}6$?*

1) a = list(i for i in range(n))

2) a = [i for i in range(n)]

3) a = (i for i in range(n))

4) a = {i for i in range(n)}

Вопрос 9

Что выведет на экран следующий код?

```
a, *b, c = [1, 2]
```

```
print(a, b, c)
```

1) [1] [] [2]

2) Будет вызвано исключение: элементов в списке меньше, чем переменных.

3) 1 0 2

4) 1 [] 2

Вопрос 10

Что выведет следующий код?

```
try:
```

```
    raise IndexError
```

```
except IndexError:
```

```
    print('Получено исключение.')
```

```
else:
```

```
    print('Но в этом нет ничего страшного.')
```

- 1) IndexError
- 2) Получено исключение.
- 3) None
- 4) TypeError
- 5) Получено исключение. Но в этом нет ничего страшного.

Ответы:

- 1) 3;
- 2) 2;
- 3) 4;
- 4) 3;
- 5) 3;
- 6) 2;
- 7) 2;
- 8) 3;
- 9) 4;
- 10) 2.

Диагностика личностных и метапредметных образовательных результатов

Ожидаемый результат	Параметры	Критерии	Методы отслеживания
Умение школьников принимать неочевидные решения, видеть нестандартные ходы как в учебной деятельности, так и в повседневной жизни.	Изобретение школьниками способов решения проблем, по новизне превосходящих авторские (общепринятые)	Оригинальность таких решений	Анализ разрозненной информации
Значительное опережение сверстников в областях знаний, связанных с информатикой и WWW.	Наличие обращений за помощью по предмету	Количество обращений	Наблюдения, беседа
Умение эффективно работать над поставленной проблемой в коллективе.	Соотношение коллективного и индивидуальных результатов	Наличие и адекватность распределения ролей в коллективе в ходе совместного решения проблем. Сравнение коллективного и личных результатов	Наблюдение Беседа Эксперимент

Изменения круга общения ребенка		Рост количества друзей среди членов объединения	Социометрия Анкетирование Наблюдение Эксперимент
Устойчивый интерес к предмету и к внепрограммному материалу	Место учебного предмета в жизни ребенка	Приведение самостоятельных примеров на уроках	Беседа с родителями Наблюдение
	Обращение к педагогу по вопросам содержания, непосредственно не связанным с изучаемым материалом	Количество обращений. Характер вопросов и сообщений, глубина заинтересованности	Статистика (беседы при личной встрече, по телефону, e-mail)
Способность самостоятельно изучать материал	Наличие умения самостоятельно изучать трудные или значительные по объему темы	Степень самостоятельности (участие педагога) Качество усвоения	Самоанализ Беседа Проверка работ
Умение планировать свою деятельность	Развитие навыков планирования	Количество усвоенных компонент (построение сложных планов, учет взаимосвязей при «распараллеливании работы»)	Наблюдение Эксперимент Беседа с родителями
Способность к самоконтролю	Умение контролировать ход выполнения работ, требующих длительного времени	Эффективность и результативность контроля	Наблюдение Эксперимент Беседа с родителями
Умение составлять запросы	Успешность результатов	Уровень конкретики. Красота идей	Беседа
Наличие определенной культуры при решении задач	Умение понятно излагать свои мысли как устно, так и письменно	Отсутствие неверно понятых рассуждений сверстниками и взрослыми	Наблюдение Сравнение результатов на соревнованиях до и после апелляции с последующим выяснение причины в беседе с ребенком
Успешное выступление перед сверстниками	Рост успехов школьников (каждого в отдельности) и статистика по	Сравнение уровня соревнований, набранных баллов, дипломов, мест	Анализ результатов соревнований

	учебной группе		
--	----------------	--	--

Результаты контроля освоения программы

Уровни освоения программы	Параметры	Показатели
Высокий уровень (80-100%)	Теоретические знания.	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, самостоятельно ориентируется в содержании материала по темам. Учащийся заинтересован, проявляет устойчивое внимание к выполнению заданий.
Практические умения и навыки.		Способен применять практические умения и навыки во время выполнения самостоятельных заданий. Правильно и по назначению применяет инструменты. Работу аккуратно доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
Средний уровень (50-79%)	Теоретические знания.	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Учащийся заинтересован, но не всегда проявляет устойчивое внимание к выполнению задания.
Практические умения и навыки.		Владеет базовыми навыками и умениями, но не всегда может выполнить самостоятельное задание, затрудняется и просит помощи педагога. В работе допускает небрежность, делает ошибки, но может устранить их после наводящих вопросов или самостоятельно. Оценить результаты своей деятельности может с подсказкой педагога.

Набранные баллы учащимся	Уровень освоения
0–49 баллов	низкий
50–79 баллов	средний
80–100 баллов	высокий

Рефлексия индивидуальной образовательной деятельности:

1. Полученные результаты _____ соответствуют (указывается в какой степени) поставленным целям
2. Мне удалось _____
3. Я создал (достиг, участвовал и т.п.) _____
4. Я научился _____

5. Самооценка результатов на основании критериев

№	Полученные образовательные продукты	Критерии оценки		
		1 критерий	2 критерий	3 критерий
		Новизна и актуальность	Практическая значимость	Культура оформления материалов
1				
2				

В дальнейшем мне бы хотелось изучить (научиться, освоить)
