

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДОМ ДЕТСКОГО ТВОРЧЕСТВА
г. Городовиковск Республика Калмыкия**

УТВЕРЖДАЮ

Директор МБУ ДО ДТТ

01.09.2025

Денисова Б. В.



ПРОГРАММА

«Программирование на Python»

Возраст: от 8 до 17 лет
Срок реализации: 2 года

Составитель: Разумовская Л. Е.
педагог дополнительного
образования

I. Комплекс основных характеристик программы

1. Пояснительная записка

Современное общество переживает активную стадию цифровой трансформации. Все больше сфер жизни людей становятся зависимыми от информационных технологий и электроники.

Одной из составляющих информационной компетентности является владение языком программирования, вследствие чего встает вопрос о выборе языка программирования, который отвечает современным требованиям к написанию программ, служит основой для дальнейшего развития и совершенствования навыков программирования и удобен в освоении подростками.

Язык программирования Python принято считать одним из самых простых в освоении. Обучаться языку возможно имея лишь базовые навыки использования компьютера, установка необходимого ПО не сложнее установки любой другой программы, а синтаксис языка не перегружен и интуитивно понятен. Данные факторы раскрывают Python максимально эффективным инструментом при обучении детей программированию.

При этом Python является современным, востребованным и перспективным языком разработки, позволяющим не только заложить фундамент знаний будущих программистов и инженеров, но получить востребованные и актуальные навыки, необходимые в учебной, проектной и повседневной деятельности.

Направленность программы

Дополнительная общеобразовательная программа «Программирование на Python» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

Актуальность программы обусловлена повышенным спросом на изучение языков программирования детьми, в частности языка Python.

Python изучается в школьном курсе информатики, необходим для решения

олимпиадных и конкурсных заданий, а также сдачи ЕГЭ, имеет прикладной характер и может использоваться для решения повседневных задач. Кроме того, Python является востребованным языком программирования, используемым профессиональными инженерами во многих сферах IT-индустрии, поэтому знание данного языка даже на базовом уровне повышает шансы будущих выпускников на трудоустройство.

Отличительная особенность программы заключается в возможности получения обучающимися универсальных компетенций, необходимых при дальнейшем изучении не только информационных технологий, но и предметов гуманитарного и естественно-научного цикла. Также в результате изучения парадигмы объектно-ориентированного подхода к программированию происходит формирование базовых знаний и умений для работы с большинством популярных языков и необходимых при освоении других IT-направлений.

Программа «Программирование на Python» является практико-ориентированной. Освоение подростками навыков программирования происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать навыками и инструментами разработки продуктов.

Также в программе отдельный модуль выделен на развитие soft-компетенций обучающихся и обучение методикам командного взаимодействия, работы над проектами, поскольку данные навыки приобретают все большее значение в современном обществе, культуре и профессиональной среде.

Адресат общеразвивающей программы

Дополнительная общеразвивающая программа «Программирование на Python» предназначена для детей в возрасте 8–17 лет. Количество обучающихся в группе – 12 человек. Состав групп постоянный.

Режим занятий, периодичность и продолжительность занятий:

длительность одного занятия составляет 2 академических часа с перерывом в 10 минут, периодичность занятий: 1 год обучения – 2 раза в неделю, 2 год – 3 раза в неделю.

Срок освоения общеразвивающей программы: определяется содержанием программы и составляет 2 года. 1 год обучения – 144 часа, 2 год обучения – 216 часов.

Форма обучения: очная

Объём общеразвивающей программы: общее количество учебных часов, запланированных на весь период обучения, необходимых для освоения программы: 360 академических часа.

2. Цель и задачи общеразвивающей программы

2.1 Цели и задачи первого года обучения

Цель программы: формирование у обучающихся базовых навыков прикладной разработки на языке программирования Python для решения практических задач и разработки продуктов.

Для успешной реализации поставленной цели необходимо решить ряд обучающих, развивающих и воспитательных задач:

Задачи:

Обучающие:

- сформировать у обучающихся базовые знания и навыки программирования на Python;
- научить использовать основные конструкции языка Python: переменные, операторы, условные конструкции, циклы, функции, структуры данных (списки, кортежи, словари);
- ознакомить с принципами функционального программирования;
- научить работать с файлами для ввода и вывода данных;
- научить использовать библиотеки и фреймворки Python для решения прикладных задач (в частности, PyGame для разработки игр);
- обучить основам работы с системой контроля версий Git;

Развивающие:

- развить навыки алгоритмического и критического мышления;
- сформировать и развить навыки работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую информацию;
- развить умения планирования, структурирования и разработки проектов, навыков организации и реализации проектной деятельности;
- обучить методикам Agile при проектной работе;
- сформировать трудовые умения и навыки, умение планировать работу, предвидеть результат и достигать его;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой и оборудованием.

Воспитательные:

- способствовать воспитанию корректного поведения в обществе, социальных норм, ролей и понимания форм социального взаимодействия в группах;
- способствовать воспитанию уважительного и продуктивного учебного сотрудничества и совместной деятельности со сверстниками в процессе проектной и учебно-исследовательской деятельности;
- способствовать формированию понимания необходимости организованного и ответственного отношения к учению, труду, другому человеку, его мнению и деятельности.

2.2 Цели и задачи второго года обучения

Цель программы: Углубить знания и навыки программирования на Python, полученные на первом году обучения, и подготовить обучающихся к разработке более сложных проектов, включая веб-приложения и работу с базами данных.

Для успешной реализации поставленной цели необходимо решить ряд обучающих, развивающих и воспитательных задач:

Обучающие:

- закрепить базовые знания Python: основные конструкции, структуры данных и циклы;
- освоить объектно-ориентированное программирование (ООП) на Python: классы, объекты, принципы ООП (инкапсуляция, наследование, полиморфизм);
- обучить работать с различными инструментами и техниками Python: аргументы функций (args, kwargs), модуль OS, исключения, пакеты и модули, лямбда-функции, функторы, декораторы, регулярные выражения;
- освоить язык SQL и научить работать с базами данных: базовые запросы, фильтрация, агрегация, группировка, подзапросы, команда JOIN, оконные функции;
- обучить создавать дашборды с метриками и графиками;
- обучить разрабатывать веб-приложения и взаимодействовать с API;
- обучить работать с популярными форматами файлов (JSON, XML);
- обучить обрабатывать исключения и создавать собственные исключения;
- обучить работать с командной строкой, создавать скрипты и аргументы, а также планировать периодические задачи.

Развивающие:

- развить навыки абстрактного мышления и проектирования сложных систем;
- развить навыки работы с данными и их анализа;
- развить навыки самостоятельного изучения новых технологий и инструментов;
- развить навыки решения проблем и отладки кода;

- развить навыки работы в команде при реализации проектной деятельности.

Воспитательные:

- воспитать ответственность за качество разрабатываемого кода;
- воспитать культуру использования современных инструментов и технологий разработки;
- воспитать настойчивость и целеустремленность в решении сложных задач;
- воспитать навыки презентации и защиты проектных работ.

3. Содержание общеразвивающей программы

3.1 Учебный (тематический) план первого года обучения

Таблица 1

№ п/п	Название модуля, темы	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Модуль 1. Основы программирования Python		26	10	16	
1.1	Введение в образовательную программу. Инструктаж по ТБ.	2	1	1	Знакомство. Опрос. Инструктаж по ТБ. Тестирование
1.2	Переменные, ввод и вывод данных	2	1	1	Опрос. Практическая работа
1.3	Арифметика. Математические операции. Базовые операторы	4	2	2	Опрос. Практическая работа
1.4	Логика. Операции отношения и логические операции.	4	2	2	Опрос. Практическая работа
1.5	Условные конструкции. Блоки	8	2	6	Опрос. Практическая работа
1.6	Строки	4	2	2	Опрос. Практическая работа
1.7	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 2. Функциональное программирование		54	24	30	
2.1	Цикл for	4	2	2	Опрос. Практическая работа
2.2	Цикл while	4	2	2	Опрос. Практическая работа
2.3	Операторы break и continue	4	2	2	Опрос. Практическая работа
2.4	Итеративные конструкции. Массивы	4	2	2	Опрос. Практическая работа
2.5	Списки	4	2	2	Опрос. Практическая работа
2.6	Решение задач по теме: циклы и линейные структуры данных (массивы и списки)	6	2	4	Решение задач. Тест

2.7	Функции. Кортежи.	6	2	4	Опрос. Практическая работа
2.8	Рекурсия	4	2	2	Опрос. Практическая работа
2.9	Словари	4	2	2	Опрос. Практическая работа
2.10	Создание и запуск скрипта	4	2	2	Опрос. Практическая работа
2.11	Файловый ввод и вывод данных	4	2	2	Опрос. Практическая работа
2.12	Решение задач	4	2	2	Решение задач. Тест
2.13	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 3. Проектная деятельность		20	11	9	
3.1	Интенсив по командообразованию	6	4	2	Решение кейсов.
3.2	Развитие Soft-компетенций, применение Agile-методик: Scrum, Kanban и extreme programming.	10	6	4	Решение кейсов.
3.3	Спринт-интенсив.	2	1	1	Решение кейсов-спринтов
3.4	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 4. Прикладное использование языка программирования Python		26	12	14	
4.1	Введение в репозитории. PyGame. Работа с удаленными репозиториями	4	2	2	Опрос. Практическая работа
4.2	Игровой цикл. События	4	2	2	Опрос. Практическая работа
4.3	Основные команды при одиночной работе с Git	4	2	2	Опрос. Практическая работа
4.4	Классические игры на клеточном поле	4	2	2	Опрос. Практическая работа
4.5	Изображения. Спрайты	4	2	2	Опрос. Практическая работа
4.6	Взаимодействия объектов	4	2	2	Опрос. Практическая работа
4.7	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 5. Подготовка итоговых проектов		18	3	15	
5.1	Работа над итоговыми проектами	8	2	6	Тестирование проектов
5.2	Инструменты и методы эффективной презентации.	6	1	5	Предзащита, презентация проектов
5.3	Итоговое занятие	4	0	4	Защита итоговых проектов
Итого		144	60	84	

Содержание учебного (тематического) плана первого года обучения

Модуль 1. Основы программирования Python.

Тема 1.1. Введение в образовательную программу. Инструктаж по ТБ.

Теория: Введение в образовательную программу, краткий обзор программы.

Инструктаж по технике безопасности.

Практика: Выполнение входной диагностики.

Тема 1.2. Арифметика. Математические операции. Логика. Операции отношения и логические операции.

Теория: Системы счисления. Понятия переменных и константы. Целочисленные типы данных. Как задать значение константы в десятичной, двоичной, восьмеричной, шестнадцатеричной системе счисления. Как указать, что константа относится к типу long. Математические операции с целочисленным типом. Логические операции и операции отношения.

Практика: Написание простейших программ, объявляющих переменные целого типа, присваивающих им значения.

Тема 1.3. Логика. Операции отношения и логические операции.

Теория: Логические операции и операции отношения. Булевый тип. Значения True и False. Операторы сравнения

Практика: Решение задач со значениями True и False

Тема 1.4. Условные конструкции. Блоки

Теория: Область действия блоков. Конструкция if-else. Конструкция switch-case. Мотивировка использования конструкции как упрощение сложных ветвлений.

Практика: Написание собственного примера на использование операторов ветвления.

Тема 1.5. Строки

Теория: Массив символов и математические операции над ними. Оператор «split». Отличие строк от символов

Практика: Написание собственного примера на использование строкового и символьного типов данных

Тема 1.6. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему основ программирования на Python

Модуль 2. Функциональное программирование

Тема 2.1. Цикл for.

Теория: Итеративные конструкции. Применение цикла for для задач с заведомо известным количеством итераций.

Практика: Написание примеров с использованием цикла for

Тема 2.2. Цикл for.

Теория: Цикл с предусловием while. Цикл с постусловием do-while. Их отличие от цикла for

Практика: Написание примеров с использованием циклов while и do-while

Тема 2.3. Операторы break и continue

Теория: Безусловные операторы перехода break, continue. Способы выхода из цикла. Различие в использовании безусловных операторов

Практика: Написание примеров с использованием прерывания циклом с помощью операторов break и continue

Тема 2.4. Итеративные конструкции. Массивы.

Теория: Изучение различных способов повторения действий в коде. Хранение и обработка коллекции однотипных данных. Создание, доступ к элементам по индексу, изменение и основные операции над такими структурами.

Практика: Решение задач, требующих применения изученных итеративных конструкций для автоматизации повторяющихся действий и работы с элементами массивов.

Тема 2.5. Списки

Теория: Структура данных списки. Их отличие от массивов. Создание и доступ к элементам. Функции списков.

Практика: Решение задач, требующих применения функций списков

Тема 2.6. Решение задач по теме циклы и линейные структуры данных (массивы и списки)

Теория: Повторение изученного материала

Практика: Решение задач, Тестирование

Тема 2.7. Функции. Кортежи.

Теория: Изучение понятий функций, основных видов функций, способов применения, стандартного набора функций. Понятие кортежей, их отличие от списков.

Практика: Решение задач с использованием пользовательских функций.

Тема 2.8. Рекурсия

Теория: Определение рекурсии как способа описания или решения задачи через ее же собственное определение. Необходимость наличия базового случая (условия останова), при котором рекурсивный вызов не происходит, чтобы избежать бесконечного цикла. Преимущества и недостатки рекурсии.

Практика: Решение задач, иллюстрирующих применение рекурсии

Тема 2.9. Словари

Теория: Определение словаря, как коллекции элементов. Требования к ключам и ограничения на значение. Методы словарей. Итерация по словарям

Практика: Решение задач направленных на реализацию алгоритмов, использующих словари для хранения и обработки данных

Тема 2.10. Создание и запуск скрипта

Теория: Определение скрипта. Структура Python-скрипта. Аргументы командной строки.

Практика: Написание команд и программы, которая будет их выполнять

Тема 2.11. Файловый ввод и вывод данных

Теория: Понятие файлов. Режимы открытия файлов. Чтение и запись в файлах. Позиционирование в файле.

Практика: Решение задач, включающих работу с чтением и записью данных файла.

Тема 2.12. Решение задач

Теория: Обобщение материала модуля для написания более сложных программ

Практика: Решение задач.

Тема 2.13. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему функциональное программирование на Python

Модуль 3. Проектная деятельность

Тема 3.1. Интенсив по командообразованию

Теория: Роль команды при создании проекта. Распределение ролей в команде. Характеры личности участников команды, работа с людьми различных взглядов и темпераментов. Реактивный и проактивный подходы. Понятие лидерства.

Практика: Командообразование, выбор нескольких тем проекта для спринта, распределение ролей, решение кейсов на представление проектов.

Тема 3.2. Развитие Soft-компетенций, применение Agile-методик: Scrum, Kanban и extreme programming.

Теория: Роль soft-компетенций в учебной, проектной и повседневной деятельности. Описание Методик scrum, kanban и extreme programming.

Практика: Решение кейсов для реализации выбранных тем с применением методик scrum, kanban, extreme programming.

Тема 3.3. Спринт-интенсив.

Теория: Понятия спринтов, роль многозадачности и вариативность ролей в команде.

Практика: Решение кейсов-спринтов по практическим и актуальным темам, используя ранее изученные методики.

Тема 3.4. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему проектной деятельности

Модуль 4. Прикладное использование языка программирования Python

Тема 4.1. Введение в репозиторий. PyGame. Работа с удалёнными репозиториями.

Теория: Разновидности и предназначение систем контроля версий. Репозитории для хранения и обмена данными. Использование среды разработки PyCharm с удалёнными репозиториями.

Практика: Решение задач с удалёнными репозиториями.

Тема 4.2. Игровой цикл. События.

Теория: Создание игрового цикла. Понятие событий в программировании.

Обработка событий стандартных устройств ввода.

Практика: Решение задач, связанных с обработкой событий

Тема 4.3. Основные команды при одиночной работе с Git

Теория: Система контроля версий Git. Использование git при разработке приложения в одиночку - графический и консольный интерфейс, основные команды

Практика: Решение задач с использованием Git.

Тема 4.4. Классические игры на клеточном поле

Теория: Реализация клеточного поля в PyGame. Способы хранения глобальных настроек и обмена данными между игровыми агентами. Обзор классических игр на клеточном поле и вариантов их реализации при помощи PyGame.

Практика: Решение задач по реализации игры на клеточном поле.

Тема 4.5. Изображения. Спрайты

Теория: Графические возможности PyGame, чтение, изменение и встраивание изображений. Техника работы со спрайтами.

Практика: Решение задач по работе с изображениями.

Тема 4.6. Взаимодействия объектов

Теория: Обсуждение основных видов взаимодействия между игроками и другими сущностями игры, а также способов программной реализации такого рода взаимодействий.

Практика: Решение задач по реакции на взаимодействия между спрайтами.

Тема 4.7. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему прикладное использование языка программирования Python.

Модуль 5. Подготовка итоговых проектов.

Тема 5.1. Работа над итоговыми проектами.

Теория: Концепция проекта, понятия целеполагания, задачи, проблемы, актуальности.

Практика: Разработка итоговых проектов, тестирование, устранение багов,

отладка.

Тема 5.2. Инструменты и методы эффективной презентации.

Теория: Обзор инструментов для создания эффективной презентации и методов подачи информации, взаимодействия с публикой.

Практика: Разработка презентации, доклада, предзащита (пробное выступление)

Тема 5.3. Итоговое занятие.

Практика: Представление проектов, выступление перед комиссией, защита проектов.

3.2 Учебный (тематический) план второго года обучения

Таблица 2

№ п/п	Название модуля, темы	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Модуль 1. Повторение базовых знаний Python		26	11	15	
1.1	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ	2	1	1	Опрос. Инструктаж по ТБ. Тестирование
1.2	Повторение материала первого года обучения. Решение задач на основные конструкции, структуры данных и циклы.	12	6	6	Опрос. Практическая работа
1.3	Повторение материала первого года обучения. Создание простой игры PyGame	8	4	4	Опрос. Практическая работа
1.4	Тестирование и решение задач по материалу первого года обучение	4	0	4	Тестирование, решение задач
Модуль 2. Объектно-ориентированное программирование		66	25	41	
2.1	Классы и объекты	14	4	10	Опрос. Практическая работа
2.2	Принципы объектно-ориентированного программирования	12	4	8	Опрос. Практическая работа
2.3	Решение задач на основе принципов ООП	4	1	3	Решение задач
2.4	Работа с args, kwargs	4	2	2	Опрос. Практическая работа
2.5	Работа с модулем OS	4	2	2	Опрос. Практическая работа
2.6	Работа с исключениями	4	2	2	Опрос. Практическая работа
2.7	Использование пакетов и модулей	4	2	2	Опрос. Практическая работа
2.8	Lambda-функции	4	2	2	Опрос. Практическая работа

2.9	Функторы и декораторы	8	4	4	Опрос. Практическая работа
2.10	Регулярные выражения	4	2	2	Опрос. Практическая работа
2.11	Контрольное тестирование по модулю	4	0	4	Тестирование
Модуль 3. Базы данных SQL		54	22	32	
3.1	Базовые запросы SQL	4	2	2	Опрос. Практическая работа
3.2	Фильтрация данных	4	2	2	Опрос. Практическая работа
3.3	Агрегация данных	6	2	4	Опрос. Практическая работа
3.4	Группировка данных	6	2	4	Опрос. Практическая работа
3.5	Подзапросы	8	4	4	Опрос. Практическая работа
3.6	Команда JOIN	8	4	4	Опрос. Практическая работа
3.7	Оконные функции	4	2	2	Опрос. Практическая работа
3.8	Построение дашбордов. Метрики и графики	4	2	2	Опрос. Практическая работа
3.9	Программирование на Python с подключением запросов SQL	8	2	6	Опрос. Практическая работа. Решение задач
3.10	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 4. Проекты WEB и API		38	16	22	
4.1	WEB. Работа с популярными форматами файлов (json, xml)	8	4	4	Опрос. Практическая работа
4.2	WEB. Знакомство с API	8	4	4	Опрос. Практическая работа
4.3	WEB. Понятие исключения, обработка исключений. Собственные исключения.	10	4	6	Опрос. Практическая работа
4.4	WEB. Работа с командной строкой (скрипты, аргументы). Периодические задачи (модуль shedule)	10	4	6	Опрос. Практическая работа
4.5	Контрольное тестирование по модулю	2	0	2	Тестирование
Модуль 5. Проектная деятельность		32	8	24	
5.1	Проблематизация	4	1	3	Опрос
5.2	Гипотеза	4	1	3	Анализ работы
5.3	Цели и задачи	4	1	3	Анализ работы
5.4	Диаграмма Ганте	4	1	3	Анализ работы
5.5	Работа над проектом	8	3	5	Анализ работы
5.6	Работа с презентацией	4	1	3	Предзащита, презентация проектов
5.7	Итоговая защита проекта	4	0	4	Защита итоговых проектов
Итого		216	82	134	

Содержание учебного (тематического) плана второго года обучения

Модуль 1. Повторение базовых знаний Python

Тема 1.1. Вводное занятие. Введение в программу 2-го года обучения.

Инструктаж по ТБ.

Теория: Общая характеристика программы. Цели и задачи. Тематические модули. ПО необходимое для 2-го года обучения.

Практика: Установка ПО

Тема 1.2. Повторение материала первого года обучения. Решение задач на основные конструкции, структуры данных и циклы

Теория: Закрепление и восстановление знаний, полученных на первом году обучения, по ключевым аспектам языка Python.

Практика: Решение задач, затрагивающих материал первого года обучения, связанного с конструкциями, структурами данных и циклами.

Тема 1.3. Повторение материала первого года обучения. Создание простой игры PyGame

Теория: Закрепление и восстановление знаний, полученных на первом году обучения, по созданию игр на Python

Практика: Создание простой игры PyGame

Тема 1.4. Тестирование и решение задач по материалу первого года обучения

Практика: Выполнение электронного теста, который охватывает материал первого года обучения

Модуль 2. Объектно-ориентированное программирование

Тема 2.1. Классы и объекты.

Теория: Понятия «класс», «объект». Описание протокола, полей, методов класса. Понятие статичные методы. Конструкторы классов.

Практика: Написание примеров классов и соответствующих им примеры объектов, полей и методов.

Тема 2.2. Принципы объектно-ориентированного программирования

Теория: Изучение базовых принципов объектно-ориентированного

программирования: инкапсуляция, полиморфизм и наследование. Конструкция переопределения методов. Понятие интерфейса и наследование интерфейса

Практика: Решение задач, затрагивающих принципы ООП.

Тема 2.3. Решение задач на основе принципов ООП

Теория: Применение принципов ООП на примере практических задач.

Практика: Решение практических задач на основе принципов ООП.

Тема 2.4. Работа с args, kwargs

Теория: Позиционные и именованные аргументы. Оператор «звездочка».

Комбинирование позиционных и именованных аргументов

Практика: Создание функций, принимающих произвольное количество позиционных и именованных аргументов. Написание конструкций, комбинирующих args и kwargs для создания гибких и универсальных функций.

Тема 2.5. Работа с модулем OS

Теория: Модуль OS. Функции модуля. Директория. Вычисление размера. Обработка путей.

Практика: Написание команд с использованием модуля OS.

Тема 2.6. Работа с исключениями

Теория: Механизм исключений. Обработка исключений. Встроенные исключения.

Практика: Создание пользовательских исключений и их обработка

Тема 2.7. Использование пакетов и модулей

Теория: Классификация модулей. Относительный и абсолютный импорт. Виртуальное окружение. Модуль venv.

Практика: Решение задач, связанных с импортом и использованием стандартных модулей Python. Решение задач с использованием модуля venv. Создание собственных модулей и пакетов.

Тема 2.8. Lambda-функции

Теория: Анонимная функция. Объявление lambda-функции. Значение аргументов по умолчанию. Синтаксис lambda-функции. Совместное использование.

Практика: Решение ранее освоенных задач с использованием lambda-

функции

Тема 2.9. Функторы и декораторы

Теория: Создание функтора. Функтора, как замена замыкания и частичного применения функции. Функции высшего класса. Определение декораторов. Декораторы с аргументом. Объекты-декораторы.

Практика: Решение задач с использованием функторов и декораторов.

Тема 2.10. Регулярные выражения

Теория: Синтаксис RegEx. Область использования регулярных выражения. Модуль re и его методы.

Практика: Решение задач на поиск определенных паттернов в тексте, замену текста, соответствующего определенному паттерну, валидацию данных

Тема 2.11. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему объектно-ориентированное программирование.

Модуль 3. Базы данных SQL

Тема 3.1. Базовые запросы SQL

Теория: Введение в SQL. Основы реляционных баз данных. Команды SELECT, FROM, WHERE, INSERT, UPDATE, DELETE. Операторы IN, BETWEEN, LIKE. NULL-значения.

Практика: Написание простых запросов командами SELECT, FROM, WHERE, INSERT, UPDATE, DELETE и операторами IN, BETWEEN, LIKE

Тема 3.2. Фильтрация данных

Теория: Логические операторы OR, AND, NOT. Операторы сравнения. Операторы IN, BETWEEN, LIKE в задача фильтрации. Применение фильтров к различным типам данных

Практика: Написание запросов, эффективно извлекающих нужные данные из больших таблиц. Комбинирование нескольких условий фильтрации с использованием логических операторов.

Тема 3.3. Агрегация данных

Теория: Изучение агрегативных функций COUNT(), SUM(), AVG(), MIN(),

MAX()).

Практика: Написание запросов с выводом суммарных значений (сумма, среднее, минимум, максимум, количество) для всей таблицы.

Тема 3.4. Группировка данных

Теория: Группировка посредством GROUP BY и HAVING. Группировка по нескольким столбцам. Использование DISTINCT

Практика: Написание запросов с группировкой данных по нескольким столбцам для получения более детальной информации. Комбинирование GROUP BY с WHERE, HAVING и ORDER BY для сложной фильтрации и сортировки результатов.

Тема 3.5. Подзапросы

Теория: Определение подзапроса. Подзапросы в SELECT, FROM, WHERE. Подзапросы, возвращающие одно значение, список значений и таблицу. Коррелированные подзапросы. Операторы EXISTS и NOT EXISTS.

Практика: Написание запросов с фильтрацией данных с использованием подзапросов в WHERE; вычислением значений для столбцов с использованием подзапросов в SELECT; созданием временных таблиц с использованием подзапросов в FROM. Использование коррелированных подзапросов для сложных задач. Проверка существования данных с использованием EXISTS и NOT EXISTS.

Тема 3.6. Команда JOIN

Теория: Применение JOIN. Типы JOIN: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN. Самообъединение (Self Join)

Практика: Написание запросов на объединение таблиц с использованием INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN

Тема 3.7. Оконные функции

Теория: Определение оконных функций. Синтаксис оконных функций. Основные оконные функции и фреймы.

Практика: Написание запросов, выводящих значение скользящей суммы, среднего, минимума и максимума с использованием оконных функций. Ранжирование строк в пределах каждого раздела. Получение предыдущего и следующего значения в столбце с использованием LAG() и LEAD(). Определение

первого и последнего значения в окне с использованием FIRST_VALUE() и LAST_VALUE().

Тема 3.8. Построение дашбордов. Метрики и графики.

Теория: Определение дашборда. Принципы проектирование дашбордов. Метрики (KPIs). Визуализация данных. Инструменты для построения дашбордов.

Практика: Разработка дашбордов для различных сценариев с использованием SQL

Тема 3.9. Программирование на Python с подключением запросов SQL

Теория: Библиотеки Python для работы с базами данных. Установление соединения с базой данных. Выполнение SQL-запросов из Python. Обработка исключений

Практика: Решение задач, включающих подключение к различным базам данных, выполнение SELECT-запросов и обработку полученных данных, использование параметризованных запросов. Создание простых программ для автоматизации работы с базами данных.

Тема 3.10. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему базы данных SQL

Модуль 4. Проекты WEB и API

Тема 4.1. WEB. Работа с популярными форматами файлов (json, xml)

Теория: Структура форматов json и csv. Python-модули для работы с ними. Основные сферы применения, практика применения в WEB.

Практика: Решение web-задач с форматами json и csv

Тема 4.2. WEB. Знакомство с API

Теория: Программный интерфейс приложения в разработке для основных существующих платформ. Особенности API в WEB-разработке.

Практика: Решение задач с API в WEB-разработке.

Тема 4.3. WEB. Понятие исключения, обработка исключений. Собственные исключения.

Теория: Программные ошибки, понятие исключительной ситуации. Механизм

исключений в Python, различные практики его использования.

Практика: Решение задач с созданием собственных исключений.

Тема 4.4. WEB. Работа с командной строкой (скрипты, аргументы).

Периодические задачи (модуль *schedule*)

Теория: Интерфейс командной строки основных операционных систем. Его применение в backend-разработке и администрировании.

Практика: Решение задач в командной строке

Тема 4.5. Контрольное тестирование по модулю

Практика: Выполнение электронного теста, который охватывает тему проекты WEB и API

Модуль 5. Проектная деятельность

Тема 5.1. Проблематизация

Теория: Выявление и рассмотрение проблем.

Практика: Практическое задание по выявлению проблемы.

Тема 5.2. Гипотеза

Теория: Построение гипотезы решения проблемы

Практика: Практическое задание по построению гипотезы

Тема 5.3. Цели и задачи

Теория: Написание цели и задачи по S.M.A.R.T. технологии.

Практика: Практическое задание по написанию цели.

Тема 5.4. Диаграмма Ганте

Теория: Построение диаграммы Ганте

Практика: Практическое задание по составлению диаграммы Ганте

Тема 5.5. Работа над проектом

Теория: Консультация по работе с проектом

Практика: Работа над проектом

Тема 5.6. Работа с презентацией

Теория: Оформление презентаций, анимация слайдов, добавление PNG файлов.

Практика: Создание презентации к проекту

Тема 5.7. Итоговая защита

Практика: Защита проекта.

4. Планируемые результаты

4.1 Планируемые результаты первого года обучения

Предметные результаты:

Модуль 1: Основы программирования Python

– Знать: Основные типы данных (целые, вещественные, строки, булевы), операторы (арифметические, логические, отношения), синтаксис условных конструкций (if, elif, else).

– Уметь: Объявлять переменные, выполнять базовые арифметические и логические операции, писать программы с использованием условных конструкций, правильно обрабатывать ввод и вывод данных, работать со строками.

– Владеть: Базовыми навыками программирования на Python, способностью создавать простые программы для решения несложных задач.

Модуль 2: Функциональное программирование

– Знать: Синтаксис циклов for и while, операторы break и continue, структуру массивов и списков, особенности функций, кортежей и словарей, принцип рекурсии, методы файлового ввода/вывода.

– Уметь: Использовать циклы для обработки данных, работать с массивами и списками, создавать и вызывать функции, применять рекурсию для решения задач, создавать и использовать словари, читать и записывать данные в файлы.

– Владеть: Навыками работы с циклами и структурами данных, умением создавать и использовать функции, навыками работы с файлами.

Модуль 4: Прикладное использование языка программирования Python

– Знать: Основы работы с репозиториями (Git), игровой цикл в PyGame, основные команды Git для одиночной работы, принципы создания классических игр на клеточном поле, способы работы с изображениями и спрайтами, принципы взаимодействия объектов в PyGame.

– Уметь: Работать с удаленными репозиториями, создавать простые игры на PyGame, использовать основные команды Git для одиночной работы, создавать спрайты и организовывать взаимодействие между ними.

– Владеть: Навыками работы с PyGame и Git, способностью разрабатывать простые игры и работать с системой контроля версий.

Личностные результаты:

- Целостное мировоззрение: Формирование представления о Python как об инструменте решения широкого круга задач.
- Развитие познавательных интересов: Проявление интереса к изучению новых возможностей Python и его применению в различных областях.
- Готовность к саморазвитию и самообразованию: Формирование потребности в постоянном совершенствовании навыков программирования.
- Ответственное отношение к учебе: Активное участие в занятиях, выполнение заданий в срок и с высоким качеством.
- Способность к самооценке: Адекватная оценка собственных знаний и умений в области программирования.
- Готовность к командной работе: (Модуль 3) Проявление уважения к мнению других членов команды, умение эффективно взаимодействовать при решении общих задач.
- Умение управлять своей учебной деятельностью: (Модуль 3) Планировать свою работу, ставить цели и достигать их.
- Трудолюбие и целеустремленность: (Модуль 5) Проявление настойчивости и упорства при работе над итоговым проектом.
- Осознанное отношение к выбору будущей профессии: Рассмотрение возможности использования навыков программирования в профессиональной деятельности.

Метапредметные результаты:

- Поиск и выделение необходимой информации: Умение находить и анализировать информацию из различных источников.
- Структурирование знаний: Умение систематизировать и обобщать полученную информацию.
- Моделирование: Создание моделей объектов и процессов с использованием средств программирования.
- Установление причинно-следственных связей: Анализ результатов работы программы и выявление причин ошибок.
- Формулирование гипотез и их проверка: Предложение и проверка

различных вариантов решения задачи.

- Решение проблем: Анализ проблемной ситуации, поиск путей решения, реализация решения с помощью программного кода.

- Смысловое чтение: Извлечение необходимой информации из текста программного кода, инструкций и документации.

4.2 Планируемые результаты второго года обучения

Предметные результаты:

Модуль 1. Повторение базовых знаний Python

– Знать: Основные конструкции языка Python, типы данных, операторы, условные конструкции, циклы, структуры данных (списки, кортежи, словари), основы PyGame.

– Уметь: Решать задачи, используя базовые конструкции Python, создавать простые программы, работать со структурами данных, создавать простые игры в PyGame.

– Владеть: Базовыми навыками программирования на Python, навыками решения задач, достаточными для перехода к более сложным темам.

Модуль 2. Объектно-ориентированное программирование

– Знать: Основные принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция), понятие классов и объектов, работу с args и kwargs, работу с модулем os, основы работы с исключениями, способы создания и использования пакетов и модулей, лямбда-функции, функторы и декораторы, синтаксис регулярных выражений.

– Уметь: Создавать классы и объекты, применять принципы ООП при разработке программ, использовать args и kwargs в функциях, работать с файловой системой с помощью модуля os, обрабатывать исключения, импортировать и использовать пакеты и модули, создавать и использовать лямбда-функции, создавать и использовать функторы и декораторы, создавать и использовать регулярные выражения.

– Владеть: Навыками объектно-ориентированного программирования, умением применять ООП-принципы, навыками работы с файловой системой, исключениями и регулярными выражениями, умением организовывать код в модули и пакеты.

Модуль 3. Базы данных SQL

– Знать: Основы SQL, базовые запросы, операторы фильтрации, агрегатные функции, группировку, подзапросы, команды JOIN, оконные функции, основы построения дашбордов, способы подключения Python к базам

данных.

- Уметь: Писать SQL-запросы для извлечения, фильтрации, агрегации и группировки данных, использовать подзапросы и команды JOIN, применять оконные функции, строить простые дашборды, подключаться к базам данных из Python и выполнять SQL-запросы.

- Владеть: Навыками написания SQL-запросов, умением анализировать данные с помощью SQL, навыками работы с базами данных из Python, умением создавать визуализации данных.

Модуль 4. Проекты WEB и API

- Знать: Основы работы с форматами данных JSON и XML, основы работы с API, принципы обработки исключений в WEB-приложениях, способы работы с командной строкой, основы планирования задач (модуль schedule).

- Уметь: Работать с форматами данных JSON и XML, взаимодействовать с API, обрабатывать исключения в WEB-приложениях, писать скрипты командной строки, планировать выполнение задач.

- Владеть: Базовыми навыками работы с WEB, API и командной строкой, умением обрабатывать данные в формате JSON и XML, навыками планирования задач.

Личностные результаты:

- Систематизация знаний: Умение систематизировать и обобщать ранее полученные знания по программированию.

- Критическое мышление: Умение анализировать информацию, оценивать ее достоверность и делать обоснованные выводы.

- Развитие самостоятельности: Способность к самостоятельному изучению нового материала и решению поставленных задач.

- Усидчивость и настойчивость: Проявление терпения и целеустремленности при решении сложных задач.

- Ответственность: Выполнение задач в срок и с высоким качеством.

- Умение работать в команде: (При работе над проектами в Модуле 5) Участие в командной работе, умение слушать и учитывать мнения других, принимать общие решения.

- Развитие коммуникативных навыков: (При работе над проектами в Модуле 5) Умение представлять результаты своей работы, объяснять сложные концепции простым языком.

- Профессиональная ориентация: Осознание возможности применения полученных знаний и навыков в будущей профессиональной деятельности.

- Уверенность в себе: Уверенность в своих силах и способностях решать сложные задачи.

Метапредметные результаты:

- Поиск и анализ информации: Умение находить, анализировать и использовать информацию из различных источников (документация, учебники, интернет).

- Системное мышление: Понимание взаимосвязей между различными компонентами программного кода и системами.

- Моделирование и проектирование: Разработка моделей программных систем и процессов.

- Решение проблем: Анализ проблемных ситуаций, поиск решений, реализация решений с использованием программирования.

- Построение логических цепочек: Выстраивание логических цепочек при решении задач, умение находить ошибки и исправлять их.

- Креативность: Применение творческого подхода к решению задач, поиск нестандартных решений.

5. Условия реализации общеразвивающей программы

Материально-техническое обеспечение

Требования к помещению:

— помещение для занятий, отвечающие требованиям СП 2.4.3648-20 для учреждений дополнительного образования;

— качественное освещение;

— столы, стулья по количеству обучающихся и рабочим местом для педагога.

Оборудование:

— ноутбук преподавателя;

— проекционное оборудование – 2 шт;

— маркерная доска – 1 шт;

— ноутбук с предустановленной ОС и манипулятором типа
мышь.

Расходные материалы:

— whiteboard маркеры;

— бумага писчая;

— шариковые ручки;

— permanent маркеры.

Информационное обеспечение

Программное обеспечение: Python, Jupyter Notebook в составе дистрибутива Anaconda, среда разработки PyCharm, пакет приложений LibreOffice, Astra Linux, Mozilla Firefox.

6. Формы аттестации и оценочные материалы

Система контроля знаний и умений обучающихся представляется в виде учёта индивидуального результата по итогам выполнения практических заданий и посредством наблюдения, отслеживания динамики развития обучающегося, по каждому контрольному мероприятию и подведения в итоге суммарного балла для каждого обучающегося.

Вводная диагностика определения уровня умений, навыков, развития детей и их творческих способностей проводится в начале обучения.

Текущий контроль осуществляется регулярно во время занятий. Контроль теоретических знаний осуществляется с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций, практических работ. В практической деятельности результативность оценивается качеством выполнения работ учащихся, где анализируются положительные и отрицательные стороны работ, корректируются недостатки.

Система промежуточной и итоговой аттестации знаний и умений обучающихся представляется в виде учёта индивидуального результата по каждому контрольному мероприятию и подведения в итоге суммарного балла для каждого обучающегося.

Промежуточная аттестация реализуется посредством оценки решения задач и тестирования. Максимальное количество баллов, которое возможно получить по результатам промежуточной аттестации – 50 баллов.

Итоговая аттестация обучающихся реализуется посредством оценки решения задач и тестирования. Максимальное количество баллов, которое возможно получить по результатам итоговой аттестации – 25 баллов.

Защита итогового проекта осуществляется путем выступления-презентации обучающимся или командой обучающихся. Презентация должна включать в себя тему проекта, его цели и задачи, результаты, средства, которыми были достигнуты полученные результаты. Презентация может быть выполнена любым удобным наглядным показательным способом (видеоролик, презентация и т. п.). Бланк оценки итоговых проектов представлен в. Максимальное количество баллов

за выполнение итогового проекта – 25 баллов.

Сумма баллов результатов промежуточной аттестации, итоговой аттестации и защиты итогового проекта переводится в один из уровней освоения образовательной программы согласно таблице 3:

Уровень освоения программы по окончании обучения

Таблица 3

Баллы, набранные учащимся.	Уровень освоения
0-39	Низкий
40-79	Средний
80-100	Высокий

Формы проведения итогов по общеразвивающей программы соответствуют целям и задачам ДООП.

7. Методические материалы

Образовательный процесс осуществляется в очной форме с применением электронного обучения и дистанционных образовательных технологий.

В образовательном процессе используются следующие *методы*:

- объяснительно-иллюстративный;
- метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
- проектно-исследовательский;
- наглядный (демонстрация плакатов, схем, таблиц, диаграмм);
- использование технических средств; просмотр видеоматериалов);
- практический (практические задания; анализ и решение проблемных ситуаций и т. д.).

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания модуля, степени сложности материала, типа учебного занятия. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Формы организации образовательного процесса: групповая.

Формы проведения занятия:

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения детьми образовательной программы, в соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля: беседа, лекция, кейс, практическое занятие, защита проектов, тестирование.

Педагогические технологии: группового обучения; коллективного взаимообучения; дифференцированного обучения; разноуровневого обучения;

проблемного обучения; развивающего обучения; игровой деятельности; коммуникативная технология обучения; коллективной творческой деятельности; решения изобретательских задач; здоровьесберегающая технология.

Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- через включение в занятие динамических пауз, периодической смены деятельности обучающихся;
- через контроль педагога за соблюдением обучающимися правил работы за ПК;
- через создание благоприятного психологического климата в учебной группе в целом.

Дидактические материалы:

Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, учебная литература.

Список литературы

Нормативные документы:

1. Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в РФ»;
2. Федеральный закон от 29.12.2012 N 273-ФЗ «Об образовании в Российской Федерации»;
3. Распоряжение Правительства РФ от 31.03.2022 №678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года»;
4. Распоряжение Правительства РФ от 29.05.2015 N 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»;
5. Приказ Министерства Просвещения России от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
6. Приказ Министерства Просвещения Российской Федерации от 03.09.2019 г. №467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
7. Приказ Министерства труда и социальной защиты Российской Федерации от 05.05.2018 г. № 298н «Об утверждении профессионального стандарта «Педагогдополнительного образования детей и взрослых»;
8. Постановление Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 «Об утверждении СанПиН 2.4.3648–20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыхаи оздоровления детей и молодежи»;
9. Письмо Министерства образования и науки РФ от 18.11.2015г. № 09–3242. «О направлении Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
10. Распоряжение Правительства Свердловской области № 646-РП от 26.10.2018 «О создании в Свердловской области целевой модели развития

региональной системы дополнительного образования детей;

11. Положение о дополнительных общеобразовательных общеразвивающих программах ГАНОУ СО «Дворец молодежи», утвержденное приказом от 14.05.2020 №269-д.

Список литературы, использованной при написании программы:

1. Архитектура компьютера, Таненбаум Эндрю, Остин Тодд – СПб.:Прогресс книга, 2022 – 816 с.;
2. Гид по Computer Science для каждого программиста, Вильям Спрингер –СПб.: Питер, 2020 – 193 с.;
3. Информатика, Тимофеева Е.В. М.: Эксмо, 2021 – 176 с.;
4. Python, например, Никола Лейси, – СПб.: Питер, 2021 – 192 с.;
5. Ли Воган. «Непрактичный» Python занимательные проекты для тех, кто хочет поумнеть. – СПб.: БХВ-Петербург, 2021. – 457 с..

Электронные ресурсы:

1. Питонтьютор. Бесплатный курс по программированию с нуля. // [Электронный ресурс] URL: <https://pythontutor.ru/> (дата обращения: 14.04.2021);
2. Code Basics: обучение базовым аспектам языков программирования от образовательной платформы Hexlet. // [Электронный ресурс] URL: <https://ru.code-basics.com/> (дата обращения: 20.04.2021);

Литература, рекомендованная обучающимся:

1. Классические задачи Computer Science на языке Python, Дэвид Копец –СПб.: Питер, 2022 – 224 с.;
2. Современные операционные системы, Таненбаум Эндрю, Бос Херберт –СПб.: Питер, 2022 – 1120 с.;
3. Python Быстрый старт, Джейми Чан, 352 стр. 2021 г. – СПб.: Питер, 2022 – 224 с