

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БУРЯТИЯ
КОМИТЕТ ПО ОБРАЗОВАНИЮ АДМИНИСТРАЦИИ Г. УЛАН-УДЭ
МАУ ДО «ДЕТСКИЙ ОЗДОРОВИТЕЛЬНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР
«РОДНИК» Г. УЛАН-УДЭ

Принята на заседании
педагогического совета
от «15» сентября 2023 г.
протокол № 1



Дополнительная общеобразовательная программа
технической направленности
«Язык программирования Python»

Возраст учащихся: 9-16 лет
Срок реализации: 1 год

Автор - составитель:
Якимова Анна Сергеевна,
педагог дополнительного образования

г. Улан-Удэ, 2023

ОГЛАВЛЕНИЕ

1. Комплекс основных характеристик дополнительной общеразвивающей программы	
1.1. Пояснительная записка	3
1.2. Цель, задачи, ожидаемые результаты.....	6
1.3. Содержание программы	7
2. Комплекс организационно педагогических условий	
2.1. Календарный учебный график	13
2.2. Условия реализации программы	19
2.3. Формы аттестации	20
2.4. Оценочные материалы	22
2.5. Методические материалы.....	22
2.6. Список литературы	24
Приложения	

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы (общий)

1.1. Пояснительная записка

Нормативные правовые основы разработки ДООП:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ».
- Концепция развития дополнительного образования детей (Распоряжение Правительства РФ от 04.09.2014 г. № 1726-р).
- Постановление Главного государственного санитарного врача РФ от 04.07.2014 № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству».
- Приказ Министерства просвещения России от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»).
- Устав МАУ ДО «ДООЦ «Родник»
- Положение о разработке, структуре и порядке утверждения дополнительной общеразвивающей программы в МАУ ДО «ДООЦ «Родник»

Актуальность:

Python — самый быстрорастущий язык программирования за последние несколько лет. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование, он используется для машинного обучения, аналитике, разработке игр, в науке о данных и в других сферах. С Python работают такие гиганты, как Google, YouTube, Facebook, Pinterest, Reddit, Netflix и Spotify.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Данная дополнительная общеобразовательная программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Данная программа относится к углубленному уровню, так как направлена на овладение знаниями и развитие навыков, она дает возможность ребенку

погрузиться в атмосферу дополнительного образования, попробовать себя в новом лично значимом виде деятельности. Родителям дает возможность разобраться с логикой дополнительного образования и наметить линию индивидуального развития своего ребенка.

Обучение включает в себя следующие основные предметы (разделы):

Раздел 1. Введение в Python

Раздел 2. Типы данных и операции

Раздел 3. Инструкции и синтаксис

Раздел 4. Функции и модули программирования

Раздел 5. Сложные типы данных

Раздел 6. Работа с датой и временем

Раздел 7. Файлы

Раздел 8. Исключение

Раздел 9. Объектно-ориентированное программирование в языке Python

Раздел 10. Разработка графических интерфейсов в программе на языке Python

Раздел 11. Написание и отладка программ. Самостоятельная работа

Вид программы:

Модифицированная программа – это программа, в основу которой, положена примерная (типовая) программа либо программа, разработанная другим автором, но измененная с учетом особенностей образовательной организации, возраста и уровня подготовки детей, режима и временных параметров осуществления деятельности, нестандартности индивидуальных результатов.

Направленность программы: техническая

Адресат программы:

Возраст детей, для которых разработана программа от 9 до 16 лет. Дети данного возраста способны на хорошем уровне выполнять практические предлагаемые задания по программированию. Период развития детей от 9 до 16 лет отличается большой активностью как психоэмоциональной, так и физической и глубокой перестройкой организма. Характерные черты подростка - стремление ко всему новому, необычному, интерес к технике и к технической новинке, стремление к активной деятельности. Все эти особенности используются в подборе материала для практических работ.

Срок и объем освоения программы:

На 1 год, 144 академических часов, из них:

- «Вводный модуль» 144 академических часа;

Форма обучения: очная, в случае дистанционного обучения – очная с

использованием ДОТ.

Особенности организации образовательной деятельности: группы разновозрастные, с учетом возрастных и психологических особенностей содержание программы разделено на две возрастные группы 9-12 лет и 13-16 лет.

Режим занятий:

Язык программирования Python	Вводный модуль
В разделе «Язык программирования Python» рассматриваются основы языка программирования Python, создание простых и сложных приложения, а также игр, используя возможности среды программирования Pycharm 2022	4 часа в неделю; 144 часа в год.

1.2. Цель, задачи, ожидаемые результаты

Цель:

Основной целью данного учебного курса является ознакомление обучающегося с объектно-ориентированным языком программирования Python, с возможностями, синтаксисом языка, технологией и методами программирования в среде Python, обучение практическим навыкам программирования на языке Python для решения типовых задач математики и информатики, а также при разработке простейших игр. Программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Задачи:

Обучающие:

- познакомить с базовыми понятиями основ программирования
- познакомить с особенностями синтаксиса языка программирования Python
- познакомить с принципами объектно-ориентированного программирования
- формировать навык тестирования и оптимизации созданных проектов.
- формирование навыков работы в системе программирования Python;
- формирование навыков грамотной разработки программ;
- формирование практических навыков разработки игр.
- овладение базовыми понятиями теории алгоритмов при решении математических задач;

Развивающие:

- формирование навыков алгоритмического и логического мышления;
- формировать навык публичного выступления и презентации
- формировать навык планирования деятельности и основ тайм-менеджмента
- формировать и расширять словарный запас на английском языке
- формировать функциональную грамотность
- совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.
- развитие интереса учащихся к изучению программирования;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники;

Воспитательные:

- воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект
- формировать умение давать качественную обратную связь и реагировать на нее.
- формировать информационную культуру

Ожидаемые результаты:

	Вводный модуль
Знать	<ul style="list-style-type: none">- этапы решения задачи на компьютере;- типы данных;- базовые конструкции изучаемых языков программирования;- принципы структурного и модульного программирования;- принципы объектно-ориентированного программирования;- основы создания приложения в Python
Уметь	<ul style="list-style-type: none">- работать в среде программирования;- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;- настраивать рабочую среду Pycharm- записывать алгоритмические структуры на языке программирования Python;- использовать Python для решения задач из области математики, физики;- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;- использовать основные алгоритмические приемы при решении математических задач;- решать простые, сложные и нестандартные задачи;- создавать простые игры;- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.
Владеть	<ul style="list-style-type: none">- навыки владения системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;

1.3. Содержание программы
Учебный план
Первый год обучения. Вводный модуль

Таблица 1.3.1

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1	Раздел 1. Введение в Python	4	2	2	
1.1	Инструкция и структура программы	2	2	0	Опрос
1.2	Установка Python. Ввод и вывод данных	2	0	2	Опрос Практическое задание
2	Раздел 2. Типы данных и операции	12	4	8	
2.1.	Переменные. Операторы	6	2	4	Опрос Практическое задание
2.2.	Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	6	2	4	Опрос Практическое задание
3	Раздел 3. Инструкции и синтаксис	12	4	8	
3.1.	Условный оператор if, Альтернативный оператор elif, else	6	2	4	Опрос Практическое задание
3.2.	Цикл while. Цикл for. Обработка включений. Функция random. Случайные числа	6	2	4	Опрос Практическое задание

4	Раздел 4. Функции и модули в программировании	12	4	8	
4.1.	Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат	6	2	4	Тест по теме Практическое задание
4.2.	Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк	6	2	4	Опрос Практическое задание
5	Раздел 5. Сложные типы данных	24	8	16	
5.1.	Списки. Срезы списков. Решение задач со списками	6	2	4	Опрос Практическое задание
5.2.	Матрицы	6	2	4	Опрос Практическое задание
5.3.	Словари	6	2	4	Опрос Практическое задание
5.4.	Множества в языке Python	6	2	4	Опрос Практическое задание
6	Раздел 6. Работа с датой и временем	12	6	6	

6.1.	Получение текущей даты и времен	4	2	2	Опрос Практическое задание
6.2.	Форматирование даты и времени	4	2	2	Опрос Практическое задание
6.3.	Модули datetime и calendar	4	2	2	Опрос Практическое задание
7	Раздел 7. Файлы	8	4	4	
7.1.	Методы и функции для работы с файлами	4	2	2	Опрос Практическое задание
7.2.	Перенаправление ввод/вывод	4	2	2	Опрос Практическое задание
8	Раздел 8. Исключение	8	4	4	
8.1.	Обработка исключений	4	2	2	Опрос Практическое задание
8.2.	Получение информации об исключении. Создание новых исключений	4	2	2	Опрос Практическое задание
9	Раздел 9. Объектно-ориентированное программирование в языке Python	32	16	16	
9.1.	Базовые принципы объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм	4	2	2	Опрос Практическое задание
9.2.	Классы в языке Python. Инструкция class	4	2	2	Опрос Практическое задание

9.3	Создание экземпляра класса	4	2	2	Опрос
9.4.	Атрибуты класса и экземпляра класса. Закрытые атрибуты	4	2	2	Опрос Практическое задание
9.5.	Методы класса. Конструктор класса <code>_init_()</code>	4	2	2	Опрос Практическое задание
9.6	Использование ссылки на экземпляр класса. Статистические методы. Закрытые методы. Специальные методы	4	2	2	Опрос Практическое задание
9.7.	Наследование. Простое наследование. Множественное наследование	4	2	2	Опрос Практическое задание
9.8.	Абстрактные методы	4	2	2	Опрос Практическое задание
10	Раздел 10. Разработка графических интерфейсов в программе на языке Python	20	8	12	
10.1	Событийно-ориентированное программирование. Событие. Обработчик события. Цикл обработки событий	4	2	2	Опрос Практическое задание
10.2.	Инструменты для создания графических интерфейсов	4	2	2	Опрос Практическое задание

	пользователя. Создание базового окна. Отслеживание событий				
10.3.	Элементы графического интерфейса. Создание и конфигурирование виджета. Менеджер размещения	6	2	4	Опрос Практическое задание
10.4.	Использование элементов Button. Надпись, поле ввода, флажок, переключатель, другие виджеты	6	2	4	Опрос Практическое задание
	итого	144	60	84	

СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА

Раздел 1. Введение в Python

1.1. Инструкция и структура программы.

Теория. Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы алгоритмизации. Достоинства языка.

Практика. Тест по теме

1.2. Установка Python. Ввод и вывод данных.

Теория. Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы алгоритмизации.

Практика. Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

Раздел 2. Типы данных и операции

2.1. Переменные. Операторы

Теория. Переменные. Оператор присваивания. Имена переменных и

ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран.

Практика. Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.

2.2. Типы данных. Базовые функции для работы с различными типами данных.

Теория. Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)

Практика. Решение математических задач с использованием функции import math.

Раздел 3. Инструкции и синтаксис

3.1. Условные операторы if, elif, else.

Теория. Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python

Практика. Практическое закрепление знаний по условным операторам. Создание программ.

3.2. Циклы while, for. Обработка исключений. Случайные числа.

Теория. Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практика. Числа Фибоначчи. Решение задач с циклом for. Создание игры угадай число. Повторение пройденного.

Раздел 4. Функции и модули в программировании

4.1. Встроенные и пользовательские функции.

Теория. Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Ананимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

Практика. Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

4.2. Строки – последовательности символов.

Теория. Составной тип данных – строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практика. Написание программ. Работа со строками.

Раздел 5. Сложные типы данных

5.1. Списки. Срезы списков.

Теория. Списки. Типы список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функции range. Списки: примеры решения задач.

Практика. Написание программ. Работа со списками.

5.2. Матрицы

Теория. Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.

Практика. Создание программ. Практическое закрепление полученных знаний.

5.3.Кортежи

Теория. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практика. Создание программ. Практическое закрепление полученных знаний.

5.4.Словари

Теория. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Практика. Создание программ. Практическое закрепление полученных знаний.

5.5.Множества в языках Python

Теория. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами:

объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практика. Создание программ. Практическое закрепление полученных знаний.

Раздел 6. Работа с датой и временем

Теория. Значение даты и времени в программировании. Методы работы с датой

Практика. Прокомментировать результаты выполнения представленных методов.

6.1. Получение текущей даты и времен

Теория. Использование метода `today()`. Использование метода `now()`

Практика. Выведите текущее время, дату, а также дату и время одновременно. Используя данные, прокомментируйте результаты выполнения представленных методов.

6.2. Форматирование даты и времени

Теория. Значения форматирования даты. Форматы вывода даты и времени. Метод `strftime()`

Практика. Используя пример, извлеките текущие дату и время и представьте их в формате чч-мм-сс-дд-мм-гггг

6.3. Модули `datetime` и `calendar`

Теория. Основные модули для работы со временем и датой

Практика. Используя предыдущий пример и метод `datetime()`, вывести текущую дату и время, которые указаны на часах. Отформатировать данные пользуясь методом `format()` и `strftime()`.

Раздел 7. Файлы

7.1. Методы и функции для работы с файлами

Теория. Методы для работы с текстовыми файлами. Запись, редактирование, удаление данных. Чтение информации из файла. Функции для работы с файлами. Режимы открытия файлов.

Практика. Составить программу, которая формирует список (квадратную матрицу произвольного порядка).

7.2. Перенаправление ввод/вывод

Теория. Функция `print()`, модуль `sys`. Функции `dump()`, `load()`

Практика. Создайте программу, которая сериализует три записи, включающие ФИО, дату рождения, направление подготовки, курс, баллы по ЕГЭ, в файл, а затем выводит их в обычный файл на экран.

Раздел 8. Исключение.

8.1. Обработка исключений. Деление на ноль и строку, отсутствие переменной. Сообщения в ответ на ошибки. Основные исключения

Практика. Создать программу, которая вызывает исключение Index Error и Key Error.

8.2. Получение информации об исключении. Создание новых исключений.

Теория. Создание класса исключений.

Практика. Ввести данные о студенте и предусмотреть сообщения о возможных ошибках

Раздел 9. Объектно-ориентированное программирование в языке Python

9.1. Базовые принципы объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм

Теория. Базовые принципы объектно-ориентированного программирования. Абстракция. Абстракция данных. Объект. Инкапсуляция. Наследование. Полиморфизм

Практика. Необходимо описать сущность объекта «студент». Какие свойства и методы не должны быть доступны для изменения, а какие могут изменяться извне?

9.2. Классы в языке Python. Инструкция class

Теория. Атрибуты(свойства). Экземпляры(объекты) класса. Процедуры и функции для работы с классами. Абстрактная сущность.

Практика. Прокомментировать класс программы.

9.3. Создание экземпляра класса.

Теория. Присвоение переменной имя класса.

Практика. Создайте класс «Школьник», имеющий атрибут «класс»

9.4. Атрибуты класса и экземпляра класса. Закрытые аргументы

Теория. Атрибуты класса. Имена переменных, имена функций.

Наследование имен. Синтаксис. Закрытые аргументы. Защищаемый аргумент и частный аргумент.

Практика. Создать класс «школьник», который будет содержать атрибуты, направление подготовки и год поступления

9.5. Методы класса. Конструктор класса `_init_()`

Теория. Отличие метода класса от функции. Специальный метод `_init_()`

Практика. Создать класс школьник, имеющий атрибуты: фио, номер класса оценка за четверть. Организовать ввод и вывод информации о школьниках

9.6. Использование ссылки на экземпляр класса. Статистические

методы. Закрытые методы. Специальные методы.

Теория. Создание ссылки на базовый класс. Декоратор @staticmethod. Метод _init_(). Метод _del_(). Метод _doc_(). Функция dir

Практика. Реализовать в программе перевод школьников одновременно на следующий год обучения. Создать статический метод, который переводит школьников на следующий год. Создать приватный метод и добавить в код обработчик исключений.

9.7. Наследование. Простое наследование. Множественное наследование

Теория. Использование свойств родительского класса в классе-наследнике

Практика. Создать новые классы компьютерных и рабочих столов с дополнительными методами.

9.8. Абстрактные методы.

Теория. Использование абстрактных методов. Встроенный декоратор @abstractmethod.

Практика. Записать программу прокомментировать результат выполнения

Раздел 10. Разработка графических интерфейсов в программе на языке Python.

10. 1. Событийно-ориентированное программирование. Событие. Обработчик события. Цикл обработки событий.

Теория. Понятие событийно-ориентированного программирования. Понятие события. Обработчик события. Цикл обработки событий. Источники события. Цель события. Объект события

Практика. Составить концептуальную модель обработчика нажатия ЛКМ на кнопку панели инструментов «сохранить» в MS Office. Составить блок-схему обработчика события

10.2. Инструменты для создания графических интерфейсов пользователя. Создание базового окна. Отслеживание событий

Теория. Графический интерфейс пользователя. Окно. Виджет. Button, label, Message, Entry, text, frame, radiobutton, checkbutton, listbox, menu, menubutton, canvas, scrollbar, scale, toplevel. Наличие и размещение виджетов в главном и дочернем окне. Отслеживание событий пользователя в модуле tkinter.

Практика. Предложить свой интерфейс. Предложить как можно задействовать радиокнопки

10.3. Инструменты для создания графических интерфейсов пользователя. Создание базового окна

Теория. Схема создания базового окна

Практика. Создать окно размером 300*300 с названием «Школьники»

10.4. Элементы графического интерфейса. Создание и конфигурирование виджета. Менеджер размещения

Теория. Объектная переменная. Использование объектно-ориентированного подхода. Метод `mainloop()`. Использование относительных и абсолютных координат. Размещение на двумерной сетке

Практика. Создать окно «Школьники» и разместить кнопки «Вывести список», «ввести данные», «создать запрос», «Выйти». Сравнить результат и прокомментировать преимущества и недостатки использования методов.

Раздел 11. Написание и отладка программ. Самостоятельная работа

Практика. Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python»

2.Комплекс организационно - педагогических условий

2.1. Календарный учебный график

Таблица 2.1.1.

Количество учебных недель	36
Количество учебных дней	(по УП)
Продолжительность каникул	с 01.01.2023-08.01.2023 01.06.2023-31.08.2023
Даты начала и окончания обучения по программе	С 05.09.2022-31.05.2023
Сроки промежуточной аттестации	по окончании вводного модуля
Сроки итоговой аттестации	по окончании вводного модуля

2.2. Условия реализации программы

Таблица 2.2.1.

Аспекты	Характеристика
Материально-техническое обеспечение	<p>Для реализации данной программы дополнительного образования технопарк располагает специальными помещениями, расположенными по адресу г. Улан-Удэ, ул. Иванова, 19. Предоставлены аудитория (квантум) для проведения занятий семинаров и практикумов, для практической реализации проектов: Hi-tech цех, коворкинг, лекторий. Материально-техническая база соответствует действующим противопожарным правилам и нормам.</p> <p>Учебный компьютерный кабинет, удовлетворяющий санитарно-гигиеническим требованиям, для занятий группы 12 человек (компьютеры, парты, стулья, доска, шкаф для УМК и библиотеки), укомплектованный выделенным каналом выхода в Интернет.</p> <p>1) Аппаратное обеспечение:</p> <ol style="list-style-type: none">1. Персональный компьютер – 12шт;2. Процессор intel ryzen 5430 и выше;3. оперативная память 4ГБ и больше;4. видеокарта, поддерживающая 256-битный цвет и разрешение 1920x12005. дисплей с диагональю 15 дюймов <p>2) Программное обеспечение:</p> <ol style="list-style-type: none">1. Операционная система: Windows 102. Редакторы Рucharm 20223. Python 3.7.104. Текстовый редактор Блокнот5. Microsoft Power point <p>3) Оборудование, необходимое для реализации программы:</p> <ol style="list-style-type: none">1. Мультимедийная проекционная установка;2. Принтер черно-белый, цветной;3. МФУ (сканер, ксерокс);

Аспекты	Характеристика
	5) Канцелярские принадлежности: ручки, карандаши, маркеры, корректоры; блокноты, тетради; бумага разных видов и формата (А3, А4, А2); клей; файлы, папки и др.
Информационное обеспечение	-аудио - видео - фото - интернет источники
Кадровое обеспечение	Реализовывать программу могут педагоги ДО, обладающие достаточными знаниями в области в специализации педагога).

2.3. Формы аттестации

Формами аттестации являются Тестирование

- Творческая работа
- Защита проекта
- Соревнования
- Конкурс

Критерии оценивания

Критерии		
ФИО обучающегося, тема проекта		
<p>Постановка цели, проблематизация:</p> <p>1.Проектная работа соответствует цели и отвечает на проблемные вопросы – 3 балла</p> <p>2.Проектная работа соответствует цели и отвечает на некоторые проблемные вопросы – 2 балла</p> <p>3.Проектная работа не совсем точно отражает цель проекта и его проблемные вопросы – 1 балл</p>		
<p>Формулировка задач проекта:</p> <p>1.Поставленные задачи ведут к достижению цели проекта – 3 балла</p> <p>2. Не все задачи ведут к достижению цели проекта – 2 балла</p> <p>3.Представленные задачи не ведут к достижению цели проекта – 1 балл</p>		
<p>Результаты работы:</p> <p>1.Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствии с правилами – 3 балла</p> <p>2. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки в оформлении – 2 балла</p> <p>3.Результаты работы, представленные при помощи</p>		

компьютерных средств, содержат значительные ошибки в оформлении – 1 балл		
<p>Выступление:</p> <p>1. Устное выступление участника логично, отсутствуют грамматические и лексические ошибки – 3 балла</p> <p>2. Устное выступление участника логично, присутствуют незначительные грамматические и лексические ошибки, не мешающие пониманию материала – 2 балла</p> <p>3. Устное выступление участника не всегда логично, присутствуют грамматические и лексические ошибки, которые затрудняют понимание – 1 балл</p>		
<p>Соответствие выступления и презентации:</p> <p>1. Выступление не повторяет текст презентации или публикации – 3 балла</p> <p>2. Выступление частично повторяет текст презентации или публикации – 2 балла</p> <p>3. Выступление полностью повторяет текст презентации или публикации – 1 балл</p>		
<p>Ответы на вопросы:</p> <p>1. В ходе устного выступления даны ответы на все вопросы – 3 балла</p> <p>2. В ходе устного выступления даны ответы на некоторые вопросы – 2 балла</p> <p>3. Обучающийся затруднялся давать правильные ответы на вопросы – 1 балл</p>		

2.4. Оценочные материалы

Таблица 2.4.1.

Показатели качества реализации ДООП	Методики
Изучения уровня освоения дополнительной общеобразовательной общеразвивающей программы	Организуется методом наблюдения, опроса, беседы, тестирования и защиты проектов обучающихся
Изучение уровня социализированности	«Изучение уровня социализированности личности учащегося» (приложение 1) Тест «Уровень социализации личности» (версия Р.И.Мокшанцева) приложение 2
Уровень развития творческого потенциала учащихся	Методика «Креативность личности» Д. Джонсона (приложение опросник)
Уровень удовлетворенности родителей предоставляемыми образовательными услугами	Изучение удовлетворенности родителей работой образовательного учреждения (методика Е.Н.Степановой)
Оценочные материалы (указать конкретно по модулям в соответствии с формами аттестации)	Тесты (Зачет)- https://onlinetestpad.com/ru/tests/python Защита проекта, кейса - критерии оценивания проектов

2.5. Методические материалы

Методы обучения:

- Словесный
- Наглядный
- Объяснительно-иллюстративный
- Репродуктивный
- Частично-поисковый
- Исследовательский
- Игровой
- Дискуссионный
- Проектный

Формы организации образовательной деятельности:

- Индивидуальная
- Индивидуально-групповая

- Групповая
- Практическое занятие
- Открытое занятие
- Акция
- Защита проекта
- Игра
- Презентация
- Олимпиада

Педагогические технологии:

- Технология индивидуального обучения
- Технология группового обучения
- Технология коллективного взаимодействия
- Технология модульного обучения
- Технология дифференцированного обучения
- Технология проблемного обучения
- Технология дистанционного обучения
- Технология исследовательской деятельности
- Проектная технология
- Здоровьесберегающая технология

Дидактические материалы:

- Раздаточные материалы
- Инструкции
- Задачи

2.7. Список литературы

Для педагога:

1.2.3

- 1) Айсманн, К. Ретуширование и обработка изображений в Photoshop: Учебное пособие. / К. Айсманн. - М. : Издательский дом «Вильямс», 2016. -234 с.
- 2) Базовый курс Windows и Интернет: Методическое пособие. [Электронный ресурс] / Сайт Поповой Натальи Алексеевны. – Режим доступа : <http://www.nat-soul.ru/?set=lib-inf&mc=3&full>.
- 3) Зайцева, Е.А., Компьютерная графика: Учебно-методическое пособие. / Е.А.Зайцева, Т.Г. Пунина. – Тамбов: Пролетарский светоч, 2013. – 116 с.
- 4) Кэлби, С. Хитрости и секреты работы в Photoshop 7. / С. Кэлби; Пер с англ. – М. : Издательский дом «Вильямс», 2014. – 167 с.
- 5) Технология работы с графической информацией: Лекция. [Электронный ресурс] / Режим доступа : http://www.ppf.krasu.ru/informatica/graph/slide_graph.htm -

2.8.

Для обучающегося:

1.2.3

1. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
4. <http://informatics.mccme.ru/course/view.php?id=156>
5. Python, например – СПб.: Питер, 2021. – 208 с.: ил. – (Серия «Библиотека программиста»).

СПИСОК ЛИТЕРАТУРЫ, ИСПОЛЬЗУЕМЫЙ ПРИ НАПИСАНИИ ПРОГРАММЫ

1. Программирование на языке высокого уровня Python: учеб, пособие для СПО /Д. Ю. Федоров. — М.: Издательство Юрайт, 2019. — 126 с. — (Серия: Профессиональное образование).
2. Учим Python, делая крутые игры/Эл Свейгарт; [пер. с англ. М.А. Райтмана]. – Москва: Эксмо, 2018. – 416с. – (Мировой компьютерный бестселлер).
3. Майк МакГрат «Программирование на Python для начинающих» Эксмо,

4. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д. М. Златопольский. — 3-е изд. (эл.). — Электрон. текстовые дан. (1 файл pdf : 226 с.). — М. : БИНОМ. Лаборатория знаний, 2015.
5. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
6. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. –М.: МПГУ, 2015.
5. <http://informatics.mccme.ru/course/view.php?id=156>
6. https://inf5.ru/podgotovka_k_olympiad/olym_zadachi_s_resheniyami.htm
7. <http://anngеorg.ru/olimp/materials>