

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №1 Р.П.СТЕПНОЕ

Принято на заседании
методического совета ЦЦО «IT-куб»
протокол от «19» апреля 2024 г. №1

УТВЕРЖДАЮ:
Директор МБОУ-СОШ №1 р.п.Степное
_____ /Исакина Н.Ю./
«23»_апреля 2024 г.

Согласовано с методическим советом
«Кванториум» ГАУДПО «СОИРО»

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«БАЗОВЫЙ КУРС.ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»
КУБ «ПРОГРАММИРОВАНИЕ НА PYTHON»**

Направленность: техническая
Возраст обучающихся: 12-17 лет
Срок реализации: 1 год

Федорович Наталья Николаевна,
педагог дополнительного образования

р.п.Степное,
2024г.

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Базовый курс. Программирование на языке Python» разработана на основании требований следующих нормативно-правовых актов:

–Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

–Концепция развития дополнительного образования детей до 2030 года и плана мероприятий по ее реализации, утвержденной распоряжением Правительства РФ от 31.03.2022 № 678-р;

–Указ Президента Российской Федерации «Стратегия научно-технологического развития Российской Федерации» (редакция от 15.03.2021г. N*143);

–Постановление Главного государственного санитарного врача РФ от 28.09.2020 N. 28

«Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

–Паспорт приоритетного проекта «Доступное дополнительное образование для детей», утвержденного президиумом Совета при Президенте РФ по стратегическому развитию и приоритетным проектам 30 ноября 2016 г;

–Приказ Министерства просвещения Российской Федерации от 27 июля 2022г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

–Приказ Министерства труда и социальной защиты РФ № 652-н от 21.09.2021 г «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;

–Порядка организации и осуществления образовательной деятельности дополнительным общеобразовательным программам, утвержденным приказом Министерства просвещения Российской Федерации № 196 от 09.11.2018г;

–за основу взяты программы Центра цифрового образования детей «IT-КУБ» г. Магнитогорска.

Актуальность программы:

Python используется в различных областях, включая разработку веб-приложений, научное моделирование, анализ данных, искусственный интеллект, машинное обучение, автоматизацию задач и многое другое. Россия имеет сильное присутствие во многих из этих областей, и изучение Python позволит российским специалистам оставаться конкурентоспособными на мировой арене. В России

существует большой спрос на IT-специалистов, в том числе на программистов, владеющих Python. Изучение Python может увеличить шансы на рынке труда и открыть двери новыми возможностями карьерного роста и развития.

Таким образом, изучение Python имеет большую актуальность для России. Этот язык программирования может быть полезен в различных сферах деятельности и обеспечить успешное будущее в IT-индустрии.

Данная программа позволяет обучающимся с разным уровнем знания информатики освоить основы программирования научиться проектировать и разрабатывать приложения на языке программирования Python.

Педагогическая целесообразность программы «Базовый курс. Программирование на языке Python» основана на применении технологий индивидуализации обучения, дифференцированного и развивающего обучения. Это обусловлено особенностями педагогических технологий.

Особенности реализации технологии индивидуализации обучения:

- оказание каждому обучающемуся индивидуальной педагогической помощи;
- учет и преодоление недостатков семейного воспитания, мотивации, воли;
- оптимизация учебного процесса для способных и одаренных обучающихся;
- формирование общеучебных умений и навыков;
- формирование адекватной самооценки учащихся;
- использование технических средств обучения.

Особенности реализации технологии дифференцированного обучения:

- учет индивидуальных возможностей обучающихся;
- вариативность учебно-познавательной деятельности;

– ориентирование на адаптацию и развитие обучающихся. Особенности реализации технологии развивающего обучения:

- обучающийся находится в центре педагогического процесса;
- цель учебного процесса в решении и организации познавательных задач;
- смысл технологии заключается в развитии мышления, а не только использовании памяти и ранее полученных знаний

Отличительная черта программы:

1. Создание реальных и практических проектов – обучающиеся получают теоретические знания, а также применяют полученные знания на практике для создания проектов с применением языка программирования.

2. Индивидуальный подход – в процессе обучения учитываются особенности каждого обучающегося, уровень развития, интересы, возможности и т.д.

3. Развитие коммуникации в коллективе – в процессе обучения обучающиеся разрабатывают работы и проекты в командах, что способствует развитию коммуникативных навыков и возможностью в дальнейшем работать в коллективе.

4. Дифференцированный подход – процесс обучения должен быть построен таким образом, чтобы обучающиеся с разной учебной подготовкой могли работать на своем уровне, получая необходимую поддержку и содействие. Также необходимо предоставить обучающимся дополнительные материалы и задания для более продвинутых детей или организовать поддержку для тех, кто испытывает трудности.

Адресат программы – дополнительная общеобразовательная общеразвивающая программа рассчитана для обучения детей в возрасте 12-17 лет. Вступительные испытания не предусмотрены. Специальных знаний, умений и навыков в предметной области не требуется.

Срок реализации программы – 1 год (36 недель)

Объем программы - 144 часа.

Направленность программы – техническая.

Уровень освоения программы – базовый.

Форма обучения – очная.

Формы организации – в подгруппах до 12 человек.

Форма организации занятий – индивидуально-групповая.

Методы обучения - словесный, наглядный, объяснительно-иллюстративный, практический, проектный.

Режим занятий – 2 занятия в неделю (4 часа). Структура двухчасового занятия:

–45 минут – рабочая часть;

–10 минут – перерыв (отдых);

–45 минут – рабочая часть.

Занятия проходят так, чтобы каждый обучающийся смог принять участие в изучении нового материала. Само занятие состоит из лекционного материала, практических заданий, обсуждений пройденного материала, повторение. В конце каждого занятия педагог проводит опрос по пройденному материалу, а также разбирает плохо усвоенный материал и пожелания, обучающихся на следующие занятия, по желанию обучающихся педагог проводит небольшие развлекательные игры. Все это способствует проведению интересных занятий, в которых присутствует коммуникация обучающихся и педагога. Каждый ребенок может активно участвовать в процессе обучения и развивать свои навыки и знания.

Цель программы – формирование и развитие у обучающихся 12-17 лет знаний, умений и навыков в области программирования на языке Python для решения практических и образовательных задач.

Задачи:

Образовательные:

–способствовать систематизации и пополнению знаний в области

алгоритмизации;

–дать базовые знания, умения и навыки о принципах и методах функционального программирования;

–дать базовые знания, умения и навыки о принципах и методах объектно-ориентированного программирования;

–сформировать навыки работы в интегрированной среде разработки на языке Python;

–сформировать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.

Метапредметные:

–сформировать навыки поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;

–способствовать развитию самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;

–познакомиться с навыками проектной деятельности

Личностные:

–совершенствовать коммуникативные навыки при работе в паре, коллективе;

–воспитать самостоятельность при решении задач и умение работать в команде.

Содержание программы

Учебный план

| № | Наименование раздела, темы | Количество часов | | | Форма контроля |
|-----|---|------------------|----------|-------|--|
| | | Теория | Практика | Всего | |
| 1 | Модуль 1. Типы и модели данных языке Python | 7 | 13 | 20 | |
| 1.1 | Тема 1.1. Техника безопасности и правила пользования компьютером. Знакомство со средой программирования на языке Python. Переменные, выражения и инструкции | 1 | 1 | 2 | Текущий контроль: дискуссия, самостоятельная работа |
| 1.2 | Тема 1.2. Математические вычисления в Python. Ввод и вывод данных | 1 | 1 | 2 | Текущий контроль: самостоятельная работа, блиц-опрос |

| | | | | | |
|----------|---|-----------|-----------|-----------|---|
| 1.3 | Тема 1.3. Использование модулей. Математические вычисления с использованием модуля math | 1 | 1 | 2 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 1.4 | Тема 1.4. Строки в Python | 1 | 1 | 2 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 1.5 | Тема 1.5. Списки в Python | 1 | 1 | 2 | Текущий контроль: самостоятельная работа |
| 1.6 | Тема 1.6. Словари и кортежи в Python | 2 | 2 | 4 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 1.7 | Тема 1.7. Решение задач по изученным темам | - | 4 | 4 | Текущий контроль: самостоятельная работа |
| 1.8 | Тема 1.8. Контрольная работа | - | 2 | 2 | Промежуточная аттестация: контрольная работа |
| 2 | Модуль 2. Основные конструкции языка Python | 16 | 28 | 44 | |
| 2.1 | Тема 2.1. Условия if-elif-else и операторы сравнения. Модуль random | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.2 | Тема 2.2. Циклы for и while. Модуль array | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.3 | Тема 2.3. Исключения и их обработка | 1 | 1 | 2 | Текущий контроль: самостоятельная работа, блиц-опрос |

| | | | | | |
|----------|---|-----------|-----------|-----------|---|
| 2.4 | Тема 2.4. Решение задач по изученным темам | - | 4 | 4 | Текущий контроль: самостоятельная работа |
| 2.5 | Тема 2.5. Контрольная работа | - | 2 | 2 | Промежуточная аттестация: контрольная работа |
| 2.6 | Тема 2.6. Функции. Рекурсия. Модули datetime и time. Часть 1. | 2 | 4 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.7 | Тема 2.7. Функции. Рекурсия. Модули datetime и time. Часть 2. | 2 | 2 | 4 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.8 | Тема 2.8. Работа с файлами | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.9 | Тема 2.9. Создание своего модуля | 1 | 1 | 2 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 2.10 | Тема 2.10. Решение задач по изученным темам | - | 4 | 4 | Текущий контроль: самостоятельная работа |
| 2.11 | Тема 2.11. Промежуточная аттестация за 1 полугодие | - | 2 | 2 | Промежуточная аттестация:: контрольная работа |
| 3 | Модуль 3. Введение в ООП | 10 | 16 | 26 | |
| 3.1 | Тема 3.1. Основы ООП. Классы, свойства и методы. Инкапсуляция и полиморфизм | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |

| | | | | | |
|----------|--|-----------|-----------|-----------|---|
| 3.2 | Тема 3.2. Конструктор и деструктор класса. Перегрузка операторов | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 3.3 | Тема 3.3. Наследование. Часть 1. | 2 | 2 | 4 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 3.4 | Тема 3.4. Наследование. Часть 2. | 2 | 2 | 4 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 3.5 | Тема 3.5. Решение задач по изученным темам | - | 4 | 4 | Текущий контроль: самостоятельная работа |
| 3.6 | Тема 3.6. Контрольная работа | - | 2 | 2 | Промежуточная аттестация: контрольная работа |
| 4 | Модуль 4. Графический интерфейс пользователя | 12 | 16 | 28 | |
| 4.1 | Тема 4.1. Библиотека Tkinter. Создание окна программы. Виджеты. Виды разметок виджетов | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 4.2 | Тема 4.2. События. Диалоговые окна | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 4.3 | Тема 4.3. Контрольная работа | - | 2 | 2 | Промежуточная аттестация: контрольная работа |
| 4.4 | Тема 4.4. Графика в Tkinter | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |

| | | | | | |
|----------|---|-----------|-----------|------------|---|
| 4.5 | Тема 4.5. Анимация в Tkinter | 3 | 3 | 6 | Текущий контроль: самостоятельная работа, блиц-опрос |
| 4.6 | Тема 4.6. Контрольная работа | - | 2 | 2 | Промежуточная аттестация: контрольная работа |
| 5 | Модуль 5. Проектная деятельность | 1 | 25 | 26 | |
| 5.1 | Тема 5.1. Постановка проблемы, задачи, утверждение темы. | 1 | 3 | 4 | Текущий контроль: блиц-опрос, дискуссия |
| 5.2 | Тема 5.2 Определение проектного решения. Разработка дорожной карты проекта. | - | 6 | 6 | Текущий контроль: наблюдение |
| 5.3 | Тема 5.3. Реализация проекта. Часть 1. | - | 6 | 6 | Текущий контроль: наблюдение |
| 5.4 | Тема 5.4. Реализация проекта. Часть 2. | - | 6 | 6 | Текущий контроль: наблюдение |
| 5.5 | Тема 5.5. Подготовка презентации репетиция выступления. | - | 2 | 2 | Текущий контроль: наблюдение |
| 5.6 | Тема 5.6. Защита проектов на Фестивале детских проектов. | - | 2 | 2 | Защита проекта |
| | ИТОГО | 46 | 98 | 144 | |

Содержание программы

Модуль 1. Типы и модели данных в языке Python

Тема 1.1 Техника безопасности и правила пользования компьютером. Знакомство со средой программирования на языке Python. Переменные, выражения и инструкции.

Теория: Инструктаж по технике безопасности. Изучение правил организации рабочего места и работы за компьютером. Общие понятия программирования.

Знакомство со средой программирования на языке Python. Изучение понятий «переменная», «выражение»,

«инструкция». Операторы ввода/вывода.

Практика: Создание своих переменных. Вывод данных на экран. Запуск программы в среде PyCharm. Первая программа «Привет, мир!» и её вариации.

Тема 1.2 Математические вычисления в Python. Ввод и вывод данных

Теория: Целые и вещественные типы данных. Операторы присвоения, ввода/вывода, простые математические операции. Применение оператора is для сравнения переменной с None.

Практика: Неявные и явные преобразования типов. Ввод и вывод данных. Простые математические операции над переменными. Написание программ для решения математических задач: «Простая арифметика», «Площадь и объем», «Площадь треугольника» и т.д.

Тема 1.3 Использование модулей. Математические вычисления с использованием модуля math

Теория: Особенности подключения модулей. Вычисления с использованием модуля math.

Функции и методы модуля math

Практика: Подключение модуля math. Использование методов модуля math. Написание программ для решение математических задач с использованием изученного модуля.

Тема 1.4 Строки в Python

Теория: Что такое строки. Работа со строками, операции со строками. Экранирование. Методы работы со строками: конкатенация, дублирование строк, получение длины строки, доступ по индексу, срезы и т.д. Однострочные и многострочные комментарии.

Практика: Создание строковых переменных. Комментирование и документирование кода.

Написание программ: «Анаграммы», «Слово по буквам», «Анализатор текста» и т.д.

Тема 1.5 Списки в Python

Теория: Определение списка. Создание списка. Добавление и удаление элементов списка.

Функции и методы списков: сортировка списка, подсчет количества элементов и т.д.

Практика: Создание различных списков. Написание программ: «Список

продуктов»,

«Веселая чепуха» и т.д.

Тема 1.6 Словари и кортежи в Python

Теория: Что такое кортеж. Создание кортежа. Различия между списком и кортежем. Функции и методы кортежей. Определение словаря. Создание словарей. Отличия от списка и кортежа. Функции и методы словарей.

Практика: Создание и работа с кортежами. Создание и работа со словарями. Использование кортежей для индексов словаря. Написание программ с использованием словарей и кортежей.

Тема 1.7 Решение задач по изученным темам

Практика: Решение практических задач по темам «Знакомство со средой программирования на языке Python. Переменные, выражения и инструкции», «Математические вычисления в Python. Ввод и вывод данных», «Использование модулей. Математические вычисления с использованием модуля math», «Строки в Python», «Списки в Python», «Кортежи в Python», «Словари в Python».

Тема 1.8 Контрольная работа

Практика: Решение индивидуальных контрольных заданий по темам «Знакомство со средой программирования на языке Python. Переменные, выражения и инструкции»,

«Математические вычисления в Python. Ввод и вывод данных», «Использование модулей. Математические вычисления с использованием модуля math», «Строки в Python», «Списки в Python», «Кортежи в Python», «Словари в Python».

Модуль 2. Основные конструкции языка Python

Тема 2.1 Условия if-elif-else и операторы сравнения. Модуль random

Теория: Что такое условные инструкции в языке Python и зачем они нужны. Синтаксис инструкций if, if-else, if-elif-else. Операторы сравнения: равно, неравно, строго больше, строго меньше, больше или равно, меньше или равно. Логические операторы: and, or, not. Операторы in, not in.

Практика: Самостоятельное использование условных конструкций и оператор сравнения при решении задач. Написание программы «Угадай число».

Тема 2.2 Циклы for и while. Модуль array

Теория: Определение циклов. Циклы while и for, их синтаксис и различия. Операторы continue и break. Пузырьковая сортировка её улучшения, модификации, мутации и разновидности.

Практика: Самостоятельное использование циклов при решении задач. Модификация программ: «Угадай число», «Камень, ножницы, бумага» и т.д.

Тема 2.3 Исключения и их обработка

Теория: Виды ошибок в программе и их различия. Синтаксические ошибки и исключения. Что такое исключения, виды исключений, обработка исключений. Конструкция try – except. Инструкции finally и else.

Практика: Самостоятельное использование конструкций для обработки исключений.

Модификация программы «Угадай число».

Тема 2.4 Решение задач по изученным темам

Практика: Решение практических задач по темам «Оператор выбора и операторы сравнения. Модуль random», «Циклы. Модуль array» и «Исключения и их обработка».

Тема 2.5 Контрольная работа

Практика: Решение индивидуальных контрольных заданий по темам «Оператор выбора и операторы сравнения. Модуль random», «Циклы. Модуль array» и «Исключения и их обработка».

Тема 2.6 Функции. Рекурсия. Модули datetime и time. Часть 1.

Теория: Что такое функция, определение и вызов функции. Именные и анонимные функции. Аргументы и параметры функции. Инструкция return. Что такое рекурсия. Рекурсивные функции. Область видимости переменных. Глобальные и локальные переменные. Быстрая сортировка.

Практика: Написание собственных функций. Написание программы, сортирующей элементы методом быстрой сортировки, и разработка игры «Крестики-нолики»

Тема 2.7 Функции. Рекурсия. Модули datetime и time. Часть 2.

Теория: Что такое функция, определение и вызов функции. Именные и анонимные функции. Аргументы и параметры функции. Инструкция return. Что такое рекурсия. Рекурсивные функции. Область видимости переменных. Глобальные и локальные переменные. Быстрая сортировка.

Практика: Написание собственных функций. Написание программы, сортирующей элементы методом быстрой сортировки, и разработка игры «Крестики-нолики»

Тема 2.8 Работа с файлами

Теория: Чтение из файла и запись в файл методами языка Python. Встроенные средства Python для работы с файлами: открытие / закрытие, чтение и запись. Работа с файлами формата `xlsx`, `txt`, `docx`, `csv`.

Практика: Самостоятельная работа с файлами. Написание программ «Чтец», «Список покупок» и т.д.

Тема 2.9 Создание своего модуля

Теория: Как создать и вызвать собственный модуль. Импорт функций и переменных модуля. Работа с модулями: создание, подключение инструкциями `import` и `from`. Подключение модуля из стандартной библиотеки

Практика: Написание и вызов собственного модуля.

Тема 2.10 Решение задач по изученным темам

Практика: Решение практических задач по темам «Функции. Модули `datetime` и `time`»,

«Работа с файлами. Модуль `os` и `os.path`» и «Создание своего модуля».

Тема 2.11 Промежуточная аттестация за 1 полугодие

Практика: Решение индивидуальных контрольных заданий по темам «Функции. Модули `datetime` и `time`», «Работа с файлами. Модуль `os` и `os.path`» и «Создание своего модуля».

Модуль 3. Введение в ООП

Тема 3.1 Основы ООП. Классы, свойства и методы. Инкапсуляция и полиморфизм

Теория: Определение ООП. Применение парадигм ООП: инкапсуляция и полиморфизм. Абстракции. Определение класса, свойства и методы в ООП. Рассмотрение жизненного цикла объекта

Практика: Создание собственных классов и вызов их экземпляров. Написание программ:

«Моя зверюшка», «Храбрые богатыри».

Тема 3.2 Конструктор и деструктор класса. Перегрузка операторов.

Теория: Определение конструктора и деструктора класса, их значимость. Что такое перегрузка оператора, зачем и когда необходимо использовать. Рассмотрение статистических и динамические атрибуты класса.

Практика: Написание программы «Проигранное сражение» модификация программы

«Храбрые богатыри».

Тема 3.3 Наследование. Часть 1.

Теория: Уровни доступа атрибута и метода. Применение наследования. Родительские классы и подклассы. Множественное наследование. Базовые и производные классы, вызов методов суперкласса в подклассе.

Практика: Самостоятельное написание подклассов, модификация программы «Храбрые богатыри».

Тема 3.4 Наследование. Часть 2.

Теория: Парадигмы ООП: наследования. Родительские классы и подклассы. Множественное наследование. Полное переопределение метода класса. Дополнение метода.

Практика: Самостоятельное написание подклассов, модификация программы «Храбрые богатыри».

Тема 3.5 Решение задач по изученным темам

Практика: Решение практических задач по темам «Основы ООП. Классы, свойства и методы. Инкапсуляция и полиморфизм» и «Объекты и экземпляры класса. Наследование».

Тема 3.6 Контрольная работа

Практика: Решение индивидуальных контрольных заданий по темам «Основы ООП. Классы, свойства и методы. Инкапсуляция и полиморфизм» и «Объекты и экземпляры класса. Наследование».

Модуль 4. Графический интерфейс пользователя

Тема 4.1 Библиотека Tkinter. Создание окна программы. Виджеты

Теория: Знакомство с библиотекой Tkinter. Создание главного окна программы. Виджеты: button, label, entry, text, frame, scrollbar, scale. Виды разметок виджетов. Упаковщики: pack, bind, grid.

Практика: Самостоятельное создание оконных программ с использованием виджетов поразметке.

Тема 4.2 События. Диалоговые окна

Теория: Типы событий. Модуль messagebox – стандартные диалоговые окна. Привязка событий к виджету. Определение и применение виджетов: menu, radiobutton, checkbutton. Диалоговые окна.

Практика: Создание событий для элементов интерфейса программы из

прошлой темы.

Тема 4.3 Решение задач по изученным темам

Практика: Решение практических задач по темам «Библиотека tkinter. Создание окна программы. Виджеты. Виды разметок виджетов» и «События. Диалоговые окна».

Тема 4.4 Контрольная работа

Практика: Решение индивидуальных контрольных заданий по темам «Библиотека tkinter. Создание окна программы. Виджеты. Виды разметок виджетов» и «События. Диалоговые окна».

Тема 4.5 Графика в Tkinter

Теория: Виджет canvas, его определение. Добавление элементов на canvas. Методы canvas для двухмерного рисования. Работа с точками и координатами. Создание линий Рисование и заливка фигур.

Практика: Написание программы, которая будет выводить несложный рисунок, нарисованный средствами Tkinter.

Тема 4.6 Анимация в Tkinter

Теория: Идентификаторы, теги и анимация. Методы move и itemconfig. Работа с точками и координатами. Постепенное движение фигуры в указанных точках. Хранение событий в атрибутах осей X и Y.

Практика: Создание анимации отрисованных объектов.

Тема 4.7 Контрольная работа

Практика: Решение индивидуальных контрольных заданий по темам «Графика в Tkinter» и «Анимация в Tkinter».

Модуль 5. Проектная деятельность

Тема 5. Постановка задачи, командообразование, утверждение темы.

Теория: Знакомство обучающихся с мероприятием «Фестиваль детских проектов», беседас обучающимися для определения темы будущего проекта

Практика: Выбор темы проекта и составление плана его разработки.

Тема 5.2 Определение проектного решения. Разработка дорожной карты проекта.

Практика: Введение в проектную деятельность. Работа над итоговым проектом в группе или самостоятельно.

Тема 5.3 Реализация проекта. Часть 1.

Практика: Работа над итоговым проектом в группе или самостоятельно.

Тема 5.4 Реализация проекта. Часть 1.

Практика: Работа над итоговым проектом в группе или самостоятельно.

Отладка кода.

Тема 5.5 Подготовка презентации репетиция выступления.

Практика: Создание презентации для представления готового проектного решения.

Репетиция выступление перед группой.

Тема 5.6 Защита проектов на Фестивале детских проектов.

Практика: Защита индивидуальных и групповых проектов. Подведение итогов курса.

Планируемые результаты

Образовательные:

- владеть знаниями в области алгоритмизации;
- владеть базовыми знаниями, умениями и навыками о принципах и методах функционального программирования;
- владеть базовыми знаниями, умениями и навыками о принципах и методах объектно-ориентированного программирования;
- ориентироваться в интерфейсе и функциональных возможностях среды разработки на языке Python;
- владеть навыками разработки программ на языке программирования Python.

Метапредметные:

- владеть навыками эффективного поиска, анализа и применения информации;
- владеть навыками применения творческого подхода для решения практических и учебных задач;
- познакомиться с навыками проектной деятельности в формате разработки, реализации и защиты группового проекта.

Личностные:

- уметь выстраивать эффективные коммуникации при работе в паре, коллективе;
- уметь принимать самостоятельные решения при выполнении задач.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

Условия реализации программы

Материально-техническое обеспечение:

Площадка проведения занятий оснащена спектром оборудования, средств обучения и воспитания для развития проектной деятельности обучающихся общеобразовательных организаций.

Для наиболее эффективного усвоения учениками данной образовательной программы, занятия необходимо проводить в светлых помещениях с хорошей вентиляцией. Для того, чтобы работа с проектором была продуктивной, необходимо затемнять зону проектора, а рабочие места обучающихся должны быть достаточно освещены.

Перечень оборудования, необходимого для освоения общеобразовательной программы:

| Наименование оборудования | Кол-во, шт. |
|--|-------------|
| Стул обучающегося | 12 |
| Стул педагога | 1 |
| Стол обучающегося | 12 |
| Стол педагога | 1 |
| Магнитно-маркерная доска | 1 |
| Проектор | 1 |
| Персональный компьютер обучающегося/планшет для обучающегося | 12 |
| Персональный компьютер педагога | 1 |
| Программное обеспечение PyCharm | 13 |

Среда PyCharm находится в свободном для скачивания и установки доступе. Среда доступна для установки на Windows. Это значит, что PyCharm может быть установлена на компьютер/ноутбук с ОС Windows. .

Информационное обеспечение:

Для реализации дополнительной общеразвивающей программы «Базовый курс: Программирование на языке Python» используются следующие материалы:

- учебно-методические пособия;
- конспекты лекций;
- комплект практических работ (Приложение 1);
- презентации;
- примеры программного кода;

Кадровые условия реализации программы

Программа реализуется педагогом дополнительного образования, обладающим навыками программирования на языке Python.

Формы аттестации обучающихся

Текущий контроль направлен на проверку уровня усвоения нового материала и выявление затруднений на ранней стадии. Текущий контроль проводится в следующих формах: дискуссия, наблюдение, самостоятельная работа. Результаты наблюдения фиксируются в листы наблюдений.

Промежуточная аттестация проводится в форме теста и выполнения практического задания по изученным темам.

Аттестация по итогам освоения программы проводится в форме представления и защиты проекта. Итоговая работа демонстрирует знания базовых навыков языка программирования Python в среде PyCharm, так же развитие пространственного и творческого мышления для решения поставленной задачи, проектирование дизайн и архитектуру приложения дополненной или виртуальной реальности. Тему итоговой работы определяет педагог в соответствии с уровнем усвоения программы, интересами и личностными особенностями обучающихся. Выполнение итоговой работы оценивается по следующим параметрам:

| Набранные баллы | Уровень освоения |
|------------------------|-------------------------|
| 0-49 | Низкий |
| 50-79 | Средний |
| 80-100 | Высокий |

Описание уровней освоения:

- «Высокий уровень» - обучающийся самостоятельно выполняет все задачи на высоком уровне, его работа отличается оригинальностью идеи, грамотным исполнением и творческим подходом.
- «Средний уровень» - обучающийся справляется с поставленными перед ним задачами, но прибегает к помощи преподавателя. Работа выполнена, но есть незначительные ошибки.
- «Низкий уровень» - обучающийся выполняет задачи, но делает грубые ошибки (по невнимательности или нерадивости). Для завершения работы необходима постоянная помощь преподавателя.

Оценочные материалы

Для отслеживания и фиксации результатов предусмотрены следующие формы контроля: опрос, дискуссия, самостоятельная работа, наблюдение.

| № | Наименование | Краткая характеристика оценочного средства |
|---|------------------------|---|
| 1 | Наблюдение | Позволяет оценить групповую и индивидуальную работу обучающихся без непосредственного вмешательства педагога. |
| 2 | Блиц-опрос | Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу |
| 3 | Дискуссия | Оценочные средства, позволяющие включить обучающихся в процесс обсуждения спорного вопроса, проблемы и оценить их умение аргументировать собственную точку зрения. |
| 4 | Самостоятельная работа | Различают задачи и задания: а) репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; б) реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей; в) творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения. |

Промежуточная аттестация проводится в форме выполнения тестового задания по изученному материалу.

Аттестация по итогам освоения программы проводится в форме представления и защиты проекта. Итоговая работа демонстрирует навыки программирования, установления причинно- следственных связей, применения алгоритмического подхода, пространственного и творческого мышления для решения поставленной проблемы.

Методические материалы

Методы обучения – словесный, наглядный, объяснительно-иллюстративный, практический, проектный.

Формы организации образовательного процесса – в группах до 12 человек.

Дифференциация обучения – объединение в группу детей по принципу

учета состояния здоровья. Заключается в организации работы различной по содержанию, объёму, сложности, методам, приёмам и средствам в зависимости от психофизических возможностей ребенка (Л. А. Дружинина).

Индивидуальный подход – гибкое использование педагогом различных форм и методов педагогического воздействия с целью достижения оптимальных результатов образовательного процесса по отношению к каждому ребенку.

Индивидуальный подход в воспитании необходим в двух отношениях: во-первых, он обеспечивает развитие индивидуального своеобразия, давая возможность максимального проявления имеющихся у ребенка способностей; во-вторых, без учета индивидуальных особенностей ребенка любое педагогическое воздействие не может быть эффективным. Вот почему для осуществления индивидуального подхода, как в обучении, так и в воспитании, необходимо изучение психологических особенностей детей.

Информационные ресурсы и литература

Список литературы для педагога:

Книги:

1. Босова, Л.Л. Информатика 8-9 классы. Начало программирования на языке Python. Дополнительные главы к учебникам / Л.Л.Босова, Н.А.Аквиланов, И.О.Кочергин и др. – М.:Бином.Лаборатория знаний, 2020. – 96 с.: ил.
2. Васильев, А.Н. Программирование на Python в примерах и задачах / А.Н. Васильев. – М.: Бомбора, 2021. – 616 с.
3. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. –М.: Форум, 2020. – 342 с.
4. Дауни, А. Основы Python. Научитесь думать как программист / Аллен Б. Дауни; пер. с англ. С.Черникова. – М.:Манн, Иванов и Фербер, 2021. – 304 с.
5. Доусон, М. Програмируем на Python / М. Доусон. – СПб.: Питер, 2020. – 416 с.: ил.
6. Жуков, Р.А. Язык программирования Python. Практикум. Учебное пособие / Р.А.Жуков. – М.: ИНФА-М, 2019. – 216 с.
7. Реализация дополнительной общеобразовательной программы по тематическому направлению «Программирование на языке Python» с использованием оборудования центра цифрового образования детей «IT-куб». Методическое пособие. Под ред. Григорьева С. Г. – Москва, 2021.

Список литературы для учащихся и родителей:

Книги:

1. Бхаргава, А. Грокаем алгоритмы: иллюстрированное пособие для программистов илюбопытствующих / А. Бхаргава. — СПб.: Питер, 2019. — 288 с.
2. Лутц, М. Изучаем Python. Том 1, пер. с англ. 5-е изд. / М. Лутц. — СПб.: Символ Плюс, 2019. — 848 с.
3. Пэйн, Б. Python для детей и родителей / Б. Пэйн; пер. с англ. М.А.Райтман. –М.:Издательсов «Э», 2017. – 352 с.: ил.
4. Свейгарт, Э. Учим Python, делая крутые игры / Э. Свейгарт; пер. с англ. М.А.Райтман. –М.: Бомбора, 2021. – 416 с.
5. Стивенсон, Б. Python. Сборник упражнений / Б. Стивенсон; пер с англ. А.Ю. Гинько. –М.:ДМК_Пресс, 2021. – 238 с.
6. Таке, А. Програмируем с детьми. Создайте 50 крутых игр на Python / А. Таке; пер. санг. М.А.Райтман. – М.: Бомбора, 2021. – 288 с.
7. Шуманн, Х.-Г. Python для детей / Х.-Г. Шуманн; пер. с англ. М.А.Райтман. – М.:ДМК-Пресс, 2019. – 344 с.

**Практические работы к дополнительной общеобразовательной
общеразвивающей программе**

«Базовый курс: Программирование на языке Python»

1. Практическая работа «Получение буквы по ее номеру в алфавите»: Применение линейных алгоритмов.
2. Практическая работа «Площадь поверхности цилиндра»: Применение линейных алгоритмов.
3. Практическая работа «Принадлежит ли точка кругу?»: Применение операторов ветвления.
4. Практическая работа «Существует ли треугольник?»: Применение операторов ветвления.
5. Практическая работа «Угадай число!»: Применение циклов for и while.
6. Практическая работа «Переверни-ка»: Применение циклов for и while.
7. Практическая работа «Является ли число простым?»: Применение циклов for и while.
8. Практическая работа «Числа Фибоначчи»: Применение циклов for и while.
9. Практическая работа «Проверка уникальности элементов списка»: Применение циклов for и while, операторов ветвления, работа со списками.
10. Практическая работа «Решето Эратосфена»: Применение циклов for и while, операторов ветвления, работа со списками.
11. Практическая работа «Наименьшее общее кратное»: Применение циклов for и while, операторов ветвления, работа с функциями.
12. Практическая работа «Циклический сдвиг»: Применение циклов for и while, операторов ветвления, работа с функциями.
13. Практическая работа «Обмен столбцов»: Применение циклов for и while, операторов ветвления, работа с матрицами.
14. Практическая работа «Сумма элементов диагоналей»: Применение циклов for и while, операторов ветвления, работа с матрицами.
15. Практическая работа «Создание словаря из нескольких списков»: Применение циклов for и while, операторов ветвления, работа со словарями и списками.
16. Практическая работа «Поход в магазин»: Применение циклов for и while,

операторов ветвления, работа со словарями и списками.

17. Практическая работа «Записать словарь в файл»: Применение циклов for и while, операторов ветвления, работа со словарями и файлами.

18. Практическая работа «Переместить содержимое файла в список»: Применение циклов for и while, операторов ветвления, работа с файлами и списками.

19. Практическая работа «Замена табуляции пробелами»: Применение циклов for и while, операторов ветвления, работа с файлами

20. Практическая работа «Моя зверюшка»: Применение объектно-ориентированного программирования.

21. Практическая работа «Храбрые богатыри»: Применение объектно-ориентированного программирования.