УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ

МО «СУДОГОДСКИЙ РАЙОН»

ВЛАДИМИРСКОЙ ОБЛАСТИ

МБОУ «Судогодская СОШ №2»

|  |  |
| --- | --- |
| РАССМОТРЕНО  На МС школы  Протокол №  от « »\_\_\_\_\_\_\_2023 г.  Руководитель МС\_\_\_\_\_\_\_\_\_\_\_ | УТВЕРЖДАЮ  Директор школы  \_\_\_\_\_\_\_\_\_\_Н.Г. Куликова  Приказ №  от «\_\_\_\_»\_\_\_\_\_\_\_\_2023 г. |

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

**«Программирование на Python»**

Социально-гуманитарная направленность

Возрастная категория: 13-15 лет

Срок реализации: 1 год

Ознакомительный уровень

Программу составила:

Свистунова Снежана Владимировна,

педагог дополнительного образования

**г. Судогда, 2023**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» научно-технической направленности. Программа предназначена для школьников 13 -15 лет, рассчитана на 36 часов.

Программа ориентирована на использование издания Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2018.

**Программа основывается на нормативные документы:**

1. Федеральный Закон от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации» (далее – ФЗ);
2. [Распоряжение Правительства Российской Федерации от 4 сентября 2014 г. № 1726-р «Концепция развития дополнительного образования детей»](https://www.rmc.viro33.ru/images/docs/norm.dok/osn/5.pdf);
3. 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";
4. Постановление Главного государственного санитарного врача РФ от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685- 21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» *(рзд.VI. Гигиенические нормативы по устройству, содержанию и режиму работы организаций воспитания и обучения, отдыха и оздоровления детей и молодежи»);*
5. Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
6. Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (разработанные Минобрнауки России совместно с ГАОУ ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования», АНО ДПО «Открытое образование», 2015 г.) (Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242).
7. Письмо Минобрнауки РФ от 11.12.2006 №06-1844 «О примерных требованиях к программам дополнительного образования детей».

**Актуальность.**

В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтения универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктовое мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Характерной чертой развития общества на протяжении последних десятилетий является его все более расширяющаяся информатизация. Отражением и следствием этой тенденции явилась потребность в подготовке подрастающего поколения к вступлению в информационное общество, любая профессиональная деятельность в котором будет связана с информатикой и информационными технологиями. Изучение данного курса имеет важное значение для развития мышления подростков. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие у школьников теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений; открывает новые возможности для овладения такими современными методами научного познания, как формализация, моделирование, компьютерный эксперимент и т.д. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте.

**Педагогическая целесообразность.**

Направленность дополнительной образовательной программы — научно-техническая. Она заключается в развитии технического и творческого мышления у обучающихся среднего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

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

**Новизна.**

**Программа составлена с учетом задач национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций, необходимых для дальнейшего участия в разработке и поддержке инфраструктур передачи, обработки и хранения данных, а также цифровых продуктов для граждан, бизнеса и власти; разработка, внедрение цифровых технологий и платформенных решений.**

**Отличительные особенности** **данной образовательной программы**:

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. Программа базируется на решении кейсов и проектной деятельности. Для повышения качества образования данные кейсы основываются на реальных ситуациях и проблемах потенциальных работодателей. Это дает возможность ранней профессиональной ориентации обучающихся.

**Возрастной диапазон освоения программы:** 13-15 лет.

**Срок реализации программы -** 1 год.

**Наполняемость группы**– 10-15 человек.

**Режим занятий.** Учебные занятия проводятся в группах один раз в неделю по 2 часа. Продолжительность одного академического часа - 40 мин.

**Общий объем реализации программы** –36 часов.

1 год обучения: 68 часов (1 раз в неделю по 2 часа)

**Особенности набора детей:** набор на обучение по программе - свободный, по желанию ребенка и с согласия родителей. Состав группы постоянный.

**Уровень программы**: ознакомительный.

**Форма обучения:** очная **(**школьная лекция, практическое занятие, проектная деятельность).

Лекционные занятия в группе 10-15 человек. Практические задания ( в том числе проектную работу) планируется выполнять индивидуально, в парах и в малых группах. Занятия проводятся в виде бесед, семинаров, лекций: для наглядности изучаемого материала используется различный мультимедийный материал – презентации, видеоролики.

**Основная цель программы:**

способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

**Задачи программы**:

***Обучающие:***

* Познакомить с базовыми понятиями основ программирования.
* Познакомить с особенностями синтаксиса языка программирования Python.
* Познакомить с принципами объектно-ориентированного программирования.
* Сформировать навык правильного оформления кода.
* Сформировать навык использования итеративного подхода при решении различных задач.
* Формировать навык решения задач и разработки проектов с помощью Python.
* Формировать навык разработки игр с помощью языка программирования Python.
* Формировать навык тестирования и оптимизации созданных проектов.

***Развивающие:***

* Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.
* 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.
* Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
* Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

***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:**

**уметь:**

- работать в среде программирования;

- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

- настраивать рабочую среду Python.

**знать:**

* этапы решения задачи на компьютере;
* типы данных;
* базовые конструкции изучаемых языков программирования;
* принципы структурного и модульного программирования;
* принципы объектно-ориентированного программирования;
* основы создания приложений в Python;
* инструментальные средства Python.

**Механизм оценки полученных знаний:**

- опрос, беседа;

- текущий контроль за выполнением заданий;

- защита итоговой проектной работы.

В качестве критериев оценки результата/продукта деятельности учащегося используются следующие критерии:

- актуальность темы проекта;

- формулировка цели и задач проекта в соответствии с темой проекта;

- выбор средств и методов, адекватных поставленным целям;

- планирование, определение сроков и последовательности выполнения работ;

- раскрытие темы и идеи проекта через содержание;

На промежуточном этапе (предзащита индивидуального проекта) оценка результата проектной деятельности осуществляется руководителем.

Критериями оценки защиты проекта являются:

- качество представления доклада;

- качество ответов на вопросы;

- использование демонстрационного материала;

- оформление демонстрационного материала.

**Формы подведения итогов реализации дополнительной образовательной программы:** публичная защита итоговой работы.

**Учебный план**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  п/п | Наименование раздела, темы | Количество часов | | | Формы  аттестации/контроля |
| Всего | Теория | Практика |
| 1. | **Раздел 1. Введение в Python** | **4** | **2** | **2** | **Тест по модулю** |
|  | * 1. Инструкции и структура программы. | 2 | 1 | 1 | Устный опрос |
|  | * 1. Установка Python. Вввод и вывод данных. | 2 | 1 | 1 | Решение задач |
| 2. | **Раздел 2. Типы данных и операции** | **12** | **6** | **6** | **Тест по модулю** |
|  | * 1. Переменные. Операторы. | 4 | 2 | 2 | Тест по теме |
|  | 2.2 Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math) | 8 | 4 | 4 | Решение задач |
| 3. | **Раздел 3. Инструкции и синтаксис** | **14** | **6** | **8** | **Тест по модулю** |
|  | * 1. Условный оператор if, Альтернативный условный оператор elif, else. | 4 | 2 | 2 | Решение задач |
|  | * 1. Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа. | 10 | 4 | 6 | Решение задач с использованием циклов |
| 4. | **Раздел 4.** [**Функции и модули в программировании**](https://younglinux.info/python/function.php) | **16** | **5** | **11** | Тест по модулю |
|  | * 1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат. | 6 | 2 | 4 | Тест по теме |
|  | * 1. Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. | 10 | 3 | 7 | Решение задач с использованием строк |
| 5. | **Раздел 5. Сложные типы данных** | **13** | **5** | **8** | **Тест по модулю** |
|  | * 1. Списки. Срезы списков. Решение задач со списками | 7 | 2 | 5 | Решение задач |
|  | * 1. Матрицы | 2 | 1 | 1 | Решение задач |
|  | * 1. Словари | 2 | 1 | 1 | Решение задач |
|  | * 1. Множества в языке Python | 2 | 1 | 1 | Решение задач |
|  | **Раздел 6. Написание и отладка программ. Самостоятельная работа** | **9** | **1** | **8** | **Защита проекта** |
|  | **Итого** | **68** | **25** | **43** |  |

**Содержание программы**

**Раздел 1. Введение в Python (4 ч.)**

**Инструкции и структура программы.** Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.

**Установка Python. Ввод и вывод данных.**Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

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

**Раздел 2. Типы данных и операции (12 ч.)**

**Переменные. Операторы.** 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.

Практическая работа: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.

**Типы данных. Базовые функции для работы с различными типами данных.** Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math)

Практическая работа: Решение математических задач с использованием функции import math.

**Раздел 3. Инструкции и синтаксис (14 ч.)**

**Условные операторы if, elif, else.** 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.

Практическая работа: Практическое закрепление знаний по условным операторам. Создание программ.

**Циклы while, for. Обработка исключений. Случайные числа.** 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. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

**Раздел 4. Функции и модули в программировании (16 ч.)**

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

Практическая работа: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

**Строки - последовательности символов.** 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.

Практическая работа: Написание программ. Работа со строками.

**Раздел 5. Сложные типы данных (13 ч.)**

**Списки. Срезы списков.** Списки. Тип 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. Функция range. Списки: примеры решения задач.

Практическая работа: Написание программ. Работа со списками.

**Матрицы.** Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Установка Python. Доступ к документации. Вввод и вывод данных. Первая программа на языке Python.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

**Кортежи.** Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

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

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

**Множества в языке Python.** Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

**Раздел 6. Написание и отладка программ. Самостоятельная работа (9 ч.)**

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

**Методическое обеспечение программы**

**При реализации программы используются различные методы:**

 словесный: рассказ, беседа, лекция;

 наглядный: опыт, иллюстрация, дидактический, наглядный материал, образцы и т.п.;

 практический: показ, постановка опытов; - по характеру деятельности:

 объяснительно-иллюстративный (рассказ, показ, лекция, фильм, карточки и т.п.);

 репродуктивный (воспроизведение, действие по алгоритму);

 проблемный (постановка проблемных вопросов, создание проблемных ситуаций);

 исследовательский метод (опыты, лабораторные, эксперименты, опытническая работа);

 проектный метод (разработка проектов, моделирование ситуаций, создание творческих работ);

 метод игры (развивающие, ролевые, деловые).

Педагогические технологии: индивидуального обучения, группового обучения, коллективного взаимообучения, дифференцированного обучения, разноуровнего обучения, проблемного обучения, дистанционного обучения педагогической мастерской, ТРИЗ, игровой деятельности, коллективной творческой деятельности, критического мышления, портфолио и др.

**Список литературы**

**Литература для учащихся:**

1. Свейгар.Эл. Учим python, делая крутые игры M: Эксмо, 2018. – 416 с.
2. Семакин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2016. - 304 с.
3. **Дополнительная:**

Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.

**Литература для педагога:**

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.
2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.
3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.
4. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
5. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.
6. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.

**ИНТЕРНЕТ ИСТОЧНИКИ**

1. ООП на Python: концепции, принципы и примеры реализации – Режим доступа: <https://proglib.io/p/python-oop/>

2. Уроки по Python для начинающих – Режим доступа: <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>

3. Алгоритмизация. Программирования Python 3 – Режим доступа: <https://stepik.org/course/7215/promo>