

ПРИНЯТА

Педагогическим советом АНО ОШ ЦПМ

(протокол от 28 августа 2024 г. №99)

УТВЕРЖДЕНА

приказом директора АНО ОШ ЦПМ

от 29 августа 2024 г. №677-ОД24

РАБОЧАЯ ПРОГРАММА

курса внеурочной деятельности «**Основы программирования**»

для обучающихся 9-10 класса

Москва, 2024 год

Пояснительная записка

1. Общая характеристика учебного курса

Курс «Основы программирования» в 10 классе является частью общей программы по программированию, освоение которой продолжится в 11 классе. Курс преследует своей целью практическое изучение программирования от базового и функционального языка Python, который применяется в большинстве сфер сегодня, до более сложных и узкоспециализированных в старших классах. Курс составлен так, чтобы даже учащийся, не имеющий никакого опыта программирования смог свободно освоить курс с самых азов, а учащиеся, имеющие опыт программирования, смогли вспомнить базу для успешного освоения курса в дальнейшем. Освоение Python – первое погружение в среду актуального программирования, курс дает базу знаний для будущих разработчиков в любой сфере, позволяя реализовывать собственные проекты в течение курса и по его завершении, а также подготовиться к реализации проектов и сложных задач в высших учебных заведениях.

Среди целей освоения курса выделены следующие:

Профориентационная – практическое освоение языка Python позволяет определить наличие интереса к сфере программирования

Развивающая – изучение нового языка программирования формирует системное мышление и позволяет решать нетривиальные задачи

Подготовительная – курс позволит свободно программировать на языке Python и даст необходимую базу для освоения других языков программирования, так как принципы формирования и логика схожи между собой

2. Место учебного курса в учебном плане

Данный курс реализуется в рамках специализации «Основы программирования» Универсального профиля Школы ЦПМ и предлагается для включения в учебный план в 9 и 10 классе. Обучение на данном курсе не предполагает предварительной подготовки в области программирования, но допускает ее и является базовым для продолжения специализации в старших классах.

3. Планируемые результаты освоения курса

Обучающийся, проходящий этот курс, получит теоретические и практические навыки в программировании на языке Python.

Личностные результаты:

1. Формирование критического мышления: Учащиеся учатся анализировать проблемы и оценивать различные подходы к их решению.
2. Развитие устойчивости при неудачах: При отладке программ учащиеся учатся воспринимать ошибки как часть учебного процесса и не сдаваться.
3. Увеличение интереса к технологиям: Учащиеся начинают осознавать важность программирования в современном мире, что может вызвать стремление развивать свои навыки дальше.
4. Формирование ответственности: Учащиеся учатся планировать выполнение проектов, устанавливать сроки и нести ответственность за результаты своей работы.
5. Сотрудничество и командная работа: При выполнении групповых проектов учащиеся развивают навыки взаимодействия и совместного решения задач.

Метапредметные результаты:

1. Навыки организации работы: Учащиеся учатся планировать и структурировать свои действия при выполнении программных заданий.
2. Критическое восприятие информации: Учащиеся учатся анализировать исходные данные и выбирать наиболее подходящие методы их обработки.
3. Развитие алгоритмического мышления: Учащиеся учатся разрабатывать и применять алгоритмы для решения задач в различных ситуациях.
4. Навыки самообучения: Учащиеся смогут искать необходимую информацию и изучать дополнительные ресурсы для углубления своих знаний.
5. Проблемное обучение: Учащиеся учатся ставить перед собой цели и находить способы их достижения, используя системный подход.

Предметные результаты:

1. Основы синтаксиса Python: Учащиеся овладеют основными конструкциями языка, такими как переменные, типы данных, операторы и функции.
2. Создание простых программ: Учащиеся смогут разрабатывать простые программы, используя навыки структурирования кода и базы алгоритмов.
3. Работа с библиотеками: Учащиеся научатся использовать стандартные библиотеки Python для решения различных задач.
4. Отладка кода: Учащиеся освоят основные методы отладки программ и технику поиска ошибок в коде.
5. Общие принципы объектно-ориентированного программирования: Учащиеся получат представление о принципах ООП и научатся применять их при написании собственных классов и объектов.

В целом, на протяжении курса учащийся поднимается от базовых знаний и примеров до более сложных задач, таким образом, улучшая свои программные навыки. В конце курса у обучающегося формируется полная картина практического применения языка Python, что позволит создавать полноценные ПО и выполнять проекты, используя тот или иной функционал языка программирования.

Содержание курса

Раздел 1. Введение в программирование и Python.

- Работа с базовыми типами. Типы int, float, str. Арифметические операции. Ввод-вывод данных с консоли. Форматированный ввод-вывод. Целочисленная арифметика.

- Условный оператор. Тип bool. Работа с условными операторами. Вложенные условия. Использование логических операторов.
- Циклы. Цикл while. Цикл for. Операторы управления циклами.

Раздел 2 Работа со списками и строками.

- Одномерные списки. Индексация. Методы списков. Считывание данных в списки. Поиск элементов. Метод map.
- Работа со строками. Строки и их методы. Срезы. Форматированный вывод
- Двумерные списки. Генераторы списков. Двумерные списки и их индексация. Считывание двумерных списков. Поиск элемента в двумерных списках.

Раздел 3. Функции, модули и продвинутое контейнеры.

- Функции. Создание собственных функций. Области видимости. Параметры по умолчанию.
- Работа с файлами. Считывание и запись в файлы.
- Множества. Формирование множеств. Основные операции со множествами: объединение, пересечение, разность множеств
- Словари. Создание словарей. Операции со словарями. Удаление, добавление элементов в словарь. Обход словарей.
- Рекурсия. Создание рекурсивных функций. Рекурсивный перебор.

Раздел 4. Декомпозиция программ.

- Модули. Введение в модули. Импортирование модулей. Стандартная библиотека Python, наиболее полезные модули.
- Исключения. Введение в исключения. Оператор try/except. Обработка исключений. Создание пользовательских исключений.

Раздел 5. Объектно-ориентированное программирование.

- Создание классов. ООП как принцип проектирования. Атрибуты и методы класса. Конструктор класса. Создание и использование объектов. Общие и уникальные атрибуты и методы.
- Инкапсуляция данных.
- Композиция. Композиция классов. Делегирование и агрегирование.
- Наследование. Создание наследуемых классов. Родительские классы. Множественное наследование.
- Полиморфизм. Работа с классами-абстракциями. Работа с классами-интерфейсами.

Раздел 6. Библиотека Pygame.

- Введение в Pygame. Установка библиотеки. Работа с Pygame-дисплеем. Работа с клавиатурой и мышью.
- Спрайты. Создание спрайтов. Загрузка изображений. Привязка поведения к спрайту. Группировка спрайтов. Анимация спрайтов.
- Работа со звуком. Загрузка звуковых файлов. Прослушивание и управление звуком. Использование звуковых эффектов

Раздел 7. Создание Графических приложений.

- Введение в tkinter. Окно приложения. Простые виджеты и их параметры.
- Работа с виджетами. Кнопки. Работа с текстовыми полями.
- Обработка событий.
- Создание таблиц и деревьев
- Работа с окнами и вкладками
- Виджет canvas

Тематическое планирование курса

№	Раздел/Тема	ЭОР	Количество ак. Часов
1	Введение в программирование на Python		16
1.1	Переменные в Python. Команда print. Арифметические операции. Ввод-вывод	https://clck.ru/3CzR8u	4
1.2	Условный оператор	https://clck.ru/3CzR8u	4
1.3	Циклы	https://clck.ru/3CzR8u	8
1.4	Зачетная работа по разделу 1		
2	Работа со списками и строками		16
2.1	Одномерные списки	https://clck.ru/3CzR8u	6
2.2	Работа со строками. Срезы. Генераторы списков	https://clck.ru/3CzR8u	6
2.3	Двумерные списки	https://clck.ru/3CzR8u	4
3	Функции, модули и продвинутое контейнеры		30
3.1	Функции	https://clck.ru/3CzR8u	6
3.2	Работа с файлами	https://clck.ru/3CzR8u	6
3.3	Множества	https://clck.ru/3CzR8u	6
3.4	Словари	https://clck.ru/3CzR8u	6
3.5	Рекурсия	https://clck.ru/3CzR8u	6
3.6	Зачетная работа по разделам 2, 3		
4	Декомпозиция программ		10
4.1	Модули	https://clck.ru/3CzR8u	4
4.2	Исключения	https://clck.ru/3CzR8u	6
5	Объектно-ориентированное программирование		16
5.1	Создание классов	https://clck.ru/3CzR8u	4
5.2	Композиция	https://clck.ru/3CzR8u	4
5.3	Наследование	https://clck.ru/3CzR8u	4
5.4	Полиморфизм	https://clck.ru/3CzR8u	4
6	Библиотека Pygame		20
6.1	Введение в Pygame	https://clck.ru/3CzR8u	4
6.2	Работа со спрайтами	https://clck.ru/3CzR8u	8
6.3	Работа со звуками	https://clck.ru/3CzR8u	8
7	Создание графических приложений		28
7.1	Введение в tkinter	https://clck.ru/3CzR8u	4
7.2	Работа с простыми виджетами	https://clck.ru/3CzR8u	4
7.3	Обработка событий	https://clck.ru/3CzR8u	4
7.4	Создание таблиц и деревьев	https://clck.ru/3CzR8u	8

7.5	Работа с окнами и вкладками	https://clck.ru/3CzR8u	4
7.6	Виджет Canvas	https://clck.ru/3CzR8u	4
7.7	Сдача годового проекта		

Методика оценки успеваемости обучающихся

Студенты оцениваются по пятибалльной системе, которая определяется средним значением от оценок полученных за выполнение домашних заданий, самостоятельных работ и дополнительных заданий от преподавателя. Так же весомый вклад в итоговую оценку за второе полугодие внесет индивидуальный проект студента.

Каждый вид практической деятельности имеет разный вклад в итоговую оценку за курс. Так работа на занятии имеет вес, равный единице, выполнение домашней работы и дополнительных заданий – 2; выполнение самостоятельной работы по итогу блока и итоговый проект – 3.

Завершающей работой курса является проект, который должен продемонстрировать умение использовать, полученные за время обучения студентом навыки. В рамках итогового проекта студент должен будет написать программу, включающую в себя как минимум 2 из 4 пройденные темы.

Сдача проекта происходит в два этапа:

Этап I. Сдача программного кода проекта. На данном этапе студент демонстрирует преподавателю готовую программу. Успешная сдача работы на данном этапе предполагает, что программа студента работает без ошибок и прерываний, а также написана с использованием объектно-ориентированного подхода разработки;

Этап II. Защита проекта. На данном этапе студент защищает проект презентуя его перед классом. В своей презентации студент должен подробно рассказать о том, какую проблему решает его проект, а также, какие методики, полученные за время обучения, были использованы.

Оценка за итоговый проект рассчитывается, как среднеарифметическое значение от баллов, полученных за 2 этапа защиты. Максимальное количество баллов, которое студент может набрать равно пяти. В спорных ситуациях, округление оценки происходит в большую сторону при значении десятичной части не менее 5 десятых.