Муниципальное общеобразовательное учреждение
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА С.ХОПЁРСКОЕ БАЛАШОВСКОГО РАЙОНА САРАТОВСКОЙ ОБЛАСТИ»
Россия, 412341, Саратовская область, Балашовский район, с. Хоперское, ул. Советская, д.72А, тел.: 75-1-22. E-mail: hopshol@ yandex.ru

Принята на заседании
педагогического совета
Протокол № 1 от 30.08 2023г.

Утверждаю:
Директор школы
_____ И.А.Смотрова
Приказ № 180 от 31.08 2023г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
Технической направленности

«Олимпиадное программирование»

Возраст обучающихся: 11-17 лет

Форма обучения: очно/заочная

Срок реализации 1 год

Автор:
педагог дополнительного образования
МОУ СОШ с.Хоперское
Кузнецов О.А.

2023 г.

1. Пояснительная записка

1.1. Общая характеристика

Дополнительная общеразвивающая программа «Олимпиадное программирование» имеет техническую направленность и ознакомительный уровень.

Программа разработана в соответствии со следующими нормативно правовыми документами: Федеральным законом № 273 -ФЗ «Об образовании в Российской Федерации» от 29 декабря 2012 года, Приказом Минобрнауки РФ от 29 августа 2013 г. N 1008, Концепцией развития дополнительного образования детей (утверждена Распоряжением Правительства Российской Федерации от 4 сентября 2014 года № 1726 -р), Письмом МО и Н РФ от 11 декабря 2006 г. N 06-1844 «О примерных требованиях к программам дополнительного образования детей», «Санитарно-эпидемиологическими требованиями к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» (СанПиН 2.4.4.3172-14).

Настоящая программа рассчитана на освоение учащимися 4-11-х классов. Основное назначение программы для обучающихся состоит во введении в алгоритмизацию и программирование, а также решении олимпиадных задач различной сложности.

Программа курса является авторской, специально адаптированной для детей с ОВЗ и детей-инвалидов,

1.1.1. Возраст детей

Программа адресована детям 8 -15 лет. В программе учитываются возрастные особенности детей.

Условия набора детей в группу: принимаются все желающие, не имеющие медицинских противопоказаний и соответствующие входным требованиям.

Наполняемость в группах составляет от 4 до 10 человек.

1.1.2. Сроки реализации программы

Программа рассчитана на 1 год обучения, 76 часов в год.

Обучение по программе осуществляется в очной форме.

1.1.3. Формы и режим занятий

Форма организации образовательной деятельности обучающихся – индивидуально-групповая, индивидуальная, групповая.

Продолжительность академического часа - 40 минут.

Программа предполагает организацию только аудиторных занятий, однако при необходимости возможно проведение и дистанционных занятий. Занятия проводятся в следующих формах: учебное занятие, игра, дискуссия, семинар, лекция, проектная работа, тренинг, экскурсия, творческая мастерская, лабораторные занятия.

1.1.4Актуальность

Развитие информационно -коммуникационных технологий и все более глубокое проникновение их во все сферы жизни требует повышения информационно - коммуникационной культуры, а также повышения профессиональной грамотности любой профессии в сфере ИКТ. Это вдвойне (и даже втройне) важно в отношении тех, кто создает и развивает эти технологии.

Поэтому очень важно с детства прививать культуру работы с вычислительной техникой, формировать алгоритмическое мышление у школьников, знакомить их с программированием, прививать интерес к этой деятельности.

Язык Python является одним из инструментов, который позволяет решать указанные задачи. Язык программирования Python в настоящее время активно развивается и набирает популярность.

Это универсальный язык высокого уровня, поддерживающий большинство стилей программирования (структурное, объектно-ориентированное, функциональное программирование и др.).

Интерпретатор языка распространяется свободно на основании лицензии подобной GNU General Public Licence, под которой распространяется большинство свободного программного обеспечения. Кроме того, язык имеет ясный синтаксис. Так, в других языках для указания программного блока (например, цикла или функции) используется больше специальных символов и ключевых слов; в результате код получается более громоздким. В Питоне же важны отступы: они то и выполняют роль "указателей" определенного программного блока. В результате программный код, написанный на языке Питон, получается более компактным и легко читаемым, что может иметь
важное значение для начинающих программистов.

1.2 Цель и задачи программы

Цель программы:

- понимание значения алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладение базовыми понятиями теории алгоритмов;
- освоение понятием алгоритма и особенности реализации алгоритмов в виде программ, написанных на языке программирования Питон.

Задачи программы:

- ознакомиться с понятиями «алгоритм», «язык программирования»;
- научиться составлять и читать блок-схемы;

- сформировать навыки выполнения технологической цепочки от записи алгоритмов на языке блок-схем, перевода этих алгоритмов на алгоритмический язык до разработки программ средствами языка программирования;
- изучить основные конструкции языка программирования;
- научиться отлаживать и тестировать программы, делать выводы о работе этих программ.

1.3 Содержание программы

Учебно-тематический план

№ п / п	**Название раздела, темы**	**Характеристика основных содержательных линий и тем** (понятий, терминов, явлений, законов, фактов и т.д.)	**Количество часов**	**Виды деятельности**	**Формы организации занятий**
1.	**Тема 1. Создание простейших программ**	Набор и запуск программы. Ввод-вывод данных. Преобразование типов данных. Пошаговая отладка программы	**4**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 2. Переменные, присваивание значений, целочисленная арифметика	Переменные. Типы данных. Операции с целочисленными типами данных. Обмен значениями двух переменных. Понятие «целочисленного деления» и «остатка от деления». Применение деления с остатком: десятичная запись числа, задачи на де- нежные суммы в рублях и копейках, задачи на минуты и секунды. Деление с округлением вверх.	**5**	познавательная деятельность	круглый стол

2.	**Тема 3. Условная инструкция, логические операции.**	Полная и неполная условная инструкция. Примеры — модуль числа, максимум двух чисел. Последовательные условные инструкции Операции сравнения. Логические операции (дизъюнкция, конъюнкция, цикл). Простейшие правила преобразования логических операций. Поиск максимума из двух элементов. Задачи на перемещения шахматных фигур. Задачи на перебор всех возможных вариантов.	**4**	познавательная деятельность	проектная и исследовательская деятельность
3.	**Тема 4. Цикл for**	Цикл for. Функция range, различные способы использования функции range. Инструкции управления циклом break и continue. Ветка else к циклу for. Задачи на суммирование числовых рядов. Задачи на считывание и обработку последовательности из n чисел по данному n. Задачи на перебор целых чисел и анализ десятичной записи чисел.	**3**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 5. Действительная арифметика, операции с действительными числами.	Тип данных float. Понятие о записи чисел с плавающей точкой (мантисса, порядок) Особенности работы с типом данных float (неточность вычислений) Функции математической библиотеки (разные виды округления, тригонометрические и обратные тригонометрические функции). Решение квадратного уравнения. Вычисление значения	**5**	познавательная деятельность	проектная и исследовательская деятельность

		многочлена по схеме Горнера. Решение системы линейных уравнений. Решение кинематических задач с использованием тригонометрических функций			
	Тема 6. Строковый тип данных и операции со строками	Тип данных str. Ввод- вывод, преобразование типов. Операции со строками (конкатенация, умножение на число). Длина строки. Сравнение строк. Лексико-графический порядок. Обращение к отдельным символам строки. Прямая и обратная индексация элементов строки. Срезы строк. Разные формы срезов. Методы объекта str (count, find, replace). Работа с отдельными сим-волами строки. Функции ord и chr. ASCII-коды, структура ASCII-таблицы.	**7**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 7. Цикл while и обработка последовательностей	Синтаксис цикла while	**8**	познавательная деятельность	круглый стол
	Тема 8. Функции, рекурсия	Функция, параметры функции, вызов функции, возвращаемое значение. Локальные и глобальные переменные. Рекурсия. Примеры рекурсивных алгоритмов. Алгоритм Евклида. Алгоритм быстрого возведения в степень. Задача про Ханойскую башню.	**8**	познавательная деятельность	проектная и исследовательская деятельность

	Тема 9. Списки	Списки, создание списков. Ввод-вывод списков. Преобразование списков в строки и обратно. Методы split и join. Операции со списками (конкатенация, умножение на число). Отличие списков от строк (списки — изменяемые объекты). Срезы списков. Модификация списков при помощи срезов. Обработка списков при помощи цикла. Функции, работающие со списками (min, max, sum) и методы списков (append, insert, pop, extend).	**6**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 10. Квадратичные алгоритмы сортировки	Линейный поиск элемента в списке при помощи цикла while. Использование барьерных элементов при поиске элемента. Алгоритм сортировки выбором. Алгоритм сортировка вставками. Алгоритм пузырьковой сортировки. Алгоритм сортировки под-счетом.	**8**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 11. Двумерные массивы	Двумерные массивы (таблицы). Представление в форме вложенных списков. Создание вложенных списков. Ввод-вывод вложенных списков. Обход вложенных списков при помощи вложенных циклов. Геометрия двумерных таблиц (повороты, отражения) Заполение двумерных таблиц «змейкой», «диагоналями», «спиралью». Построение многомерных списков.	**8**	познавательная деятельность	проектная и исследовательская деятельность

	Тема 12. Генераторы списков	Генераторы списков. Генераторы вложенных списков. Решение задач на генераторы списков.	**8**	познавательная деятельность	проектная и исследовательская деятельность
	Тема 13. Арифметические алгоритмы и задачи	Алгоритм Евклида, рекурсивная и нерекурсивная реализации. Алгоритм проверки числа на простоту. Алгоритм разложения на множители. Решето Эратосфена. Подсчет количества делителей числа перебором до корня из числа.	**4**	познавательная деятельность	круглый стол
		Итого часов:	**76**		

Тематическое планирование

№ п/п	Название раздела, темы	Всего часов	Из них:			Формы контроля
			теоретические	лабораторные и практические	проекты, учебные исследования	
1	История языков программирования. Простейшие программы. Реализация вычислений и ветвлений.	8	4	2	2	учебные исследования и проекты, , научнопрактические конференции
2	Реализация циклических, вспомогательных алгоритмов алгоритмов. Рекурсия.	15	5	2	8	учебные и исследования проекты, соревнования
3	Словари. Массивы. Обработка массивов	14	7	4	3	учебные и исследования проекты

4	Символьные строки. Обработкасимвольных строк.	15	5	8	2	учебные и исследования проекты
5	Матрицы.обработка Чтеие и файлов. Ввод, запись вывод,матриц. текстовых	15	4	6	5	олимпиады
6	Резерв времени	1	1			
	Итого:		25	22	21	
	Всего:	76				

1.4 Планируемые результаты

Разбивая получаемые навыки, можно выделить:

Личностные результаты:

● формирование ответственного отношения к учению, готовности и способности учащегося к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

● формирование целостного мировоззрения, соответствующего современному уровню общества;

● развитие осознанного и ответственного отношения к собственным поступкам в сфере использования информации;

● формирование коммуникативной компетентности в различных сферах деятельности.

Метапредметные результаты:

знать:

- отдельные способы планирования деятельности:
  - составление плана предстоящего проекта в виде рисунка, схемы, словесного описания;
  - составление плана предстоящего проекта в виде таблицы объектов, и свойств и взаимодействий;
  - разбиение задачи на подзадачи;
- распределение ролей и задач в группе;

уметь:

- составить план проекта, включая:
  - выбор темы;
  - анализ предметной области;

    - разбиение задачи на подзадачи;
- проанализировать результат и сделать выводы;
- найти и исправить ошибки;
- публично выступить с докладом;
- наметить дальнейшие пути развития проекта;

иметь первичные навыки:

- работы в группе;
- ведения спора;
- донесения своих мыслей до других.

Предметные результаты:

Учащийся должен знать:

- Алгоритмы и блоки:
    - понятие алгоритма,
    - исполнитель,
    - система команд исполнителя,
    - реализация алгоритмов.
- Математический базис:
    - отрицательные числа,
    - декартова система координат,
    - десятичные дроби,
    - операции отношения,
    - логические операции «И» и «ИЛИ»,
    - случайные числа,
    - арифметические операции и функции,
    - градусную меру угла.
    - .

уметь:

- работать в среде с языком программирования Python.

Методы и формы обучения

В соответствии с Федеральным государственным образовательным стандартом, в основу курса положен системно-деятельностный подход, который обеспечивает:

- формирование готовности к саморазвитию и непрерывному образованию;
- проектирование и конструирование социальной среды развития обучающихся в системе образования;
- активную учебно познавательную деятельность обучающихся;

● построение образовательного процесса с учётом индивидуальных возрастных, психологических и физиологических особенностей обучающихся.

Для организации образовательного процесс а используются такие формы обучения, как лекции, тесты, семинары, зачетные работы, практические работы, компьютерные проекты.

В учебном процессе ученики используют преимущественно следующие виды деятельности: аналитическую, поисковую, практическую.

Формы проведения занятий – мастер-классы, лекции, выставки компьютерных проектов, семинары, практические занятия, выступления.

Формы организации деятельности - индивидуальная, групповая, индивидуально групповая, по подгруппам.

Повышению интереса учащихся к курсу способствует высокий уровень доступности изложения материала, логически связанное размещение отдельных условно самостоятельных элементов курса, использование по дробных описаний порядка действий учащегося при выполнении той или иной операции.

2.2 Условия реализации программы

Техническое обеспечение образовательного процесса

- Персональный компьютер . Операционная система может быть любая.
- Доступ в интернет со скоростью не менее 1 Мбит/сек.
- Браузер – любой.
- Актуальная версия java-plugin в используемом браузере.
- On-line версия языка программирования Python.

Кадровое обеспечение программы

Для реализации программы необходимы преподаватели: знающие предметную область “Информатика”;

- владеющие методикой преподавания “Информатики и ИКТ” в начальной и средней школе;
- имеющие навыки программирования на любом языке программирования высокого уровня;
- имеющие навыки программирования на Python;
- освоившие материал представляемой программы.

2.3 Формы аттестации

Результативность обучения обеспечивается применением различных форм, методов и приемов, которые тесно связаны между собой и дополняют друг друга. В ходе практической деятельности педагог тактично контролирует, советует, направляет учащихся. Большая часть

занятий отводится практической работе, по окончании которой проходит обсуждение и анализ. 2.1. Стартовый контроль Определяет наличие у учащихся умений и навыков, позволяющих им сразу приступить к обучению на курсе: умение работать в браузере; наличие электронной почты; регистрация на учебных ресукрсах. 2.2. Текущий контроль Текущий контроль осуществляется с целью оперативного управления учебным процессом и его коррекции. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися заданий по каждому разделу курса. Задания составлены с учетом возраста учащихся. При безоценочной системе курса итоговый контроль по разделам курса может быть реализован в форме итогового задания или проекта, которые позволяют определить достижение планируемых результатов. Итоговый контроль Функция итогового контроля заключается в определении полноты освоения содержа-ния программы. Итоговый контроль включает: выполнение заданий по пройденным темам; Основное требование к указанным выше работам – комплексный характер. При их выполнении ученик должен проявить все знания и умения, приобретенные на курсе.

2.4 Оценочные материалы

Методы определения результата Для определения достижений и результатов прохождения программы используются: педагогическое наблюдение; оценка продуктов творческой деятельности учащихся; выступление учащегося с сообщением, докладом по теме, определенной учителем или самостоятельно выбранной; беседы, опросы. Критерии оценивания обучающихся по курсу На курсе дополнительного образов ания «Программирование в среде Scratch» дей-ствует безоценочная система. Содержание и объем материала, подлежащего проверке, определяется программой. При проверке усвоен ия материала выявляются полнота и прочность усвоения учащимися теории , а также умение применять ее на практике в зна-комых и незнакомых ситуациях. Основными формами проверки знаний, умений и навыков учащихся являются: завершенные практические работы, самостоятельная работа, устный опрос.

2.5 Методические материалы

Учебно-методическое обеспечение программы Курс “Олимпиадное программирование ” состоит из 76 занятий. Из них 22,5 часа теоретических и 53,5 часа практических. Для мотивации, в качестве демонстрации, а также обучения используется открытый банк работ. Методика преподавания курса предусматривает проведение по каждой новой теме теоретического занятия, выполнение учащимися самостоятельного практического задания на каждом уроке. Содержание задания определяется учителем для каждого ученика индивидуально, с учетом возможностей, интересов и склонностей ребенка. Сложность практической работы ученик выбирает сам. Прохождение курса сопровождается созданием

учащимися проектов по предлагаемым темам. Занятия по программе проводятся на основе общих педагогических принципов: технологии проектного обучения . Включает в себя проектирование предполагаемого результата, который достигается в процессе обучения. Используемые методы: объяснительно-иллюстративный, тренинговый, проблемный, поисковый. Обучение должно быть доступным (принцип предполагает последовательное усложнение практических заданий в создании проектов программы); принцип систематичности обучения предполагает такое построение учебного процесса, в ходе которого про исходит связывание ранее усвоенного с новым разучиваемым материалом; принцип увлекательности (интересности) успешное осуществление обучения; этот прием делает сам процесс овладения программирования интересным, приносящим чувство радости и удовлетворение.

2.6 Список литературы

Олимпиады для начинающих https://informatics.msk.ru/course/view.php?id=13

Алгоритмическое программирование 1 https://informatics.msk.ru/course/view.php?id=3150

Алгоритмическое программирование 2 https://informatics.msk.ru/course/view.php?id=4426

Алгоритмы и структуры данных 1 https://informatics.msk.ru/course/view.php?id=911

Информатика-Юниоры 2022 https://informatics.msk.ru/course/view.php?id=480

Сайт Д.П.Кириенко. Московский институт открытого образования Школа №179 http://www.179.ru/~dk/python.html

Сайт дистанционная подготовка по информатике http://informatics.mccme.ru/

Календарный учебный график реализации дополнительной общеразвивающей программы технической направленности "Олимпиадное программирование".

№ п/п	Тема урока	Количество часов			Дата изучения	Электронные цифровые образовательные ресурсы
		Всего	Контрольные работы	Практические работы		
1	Создание простейших программ	1				
2	Создание простейших программ	1				
3	Создание простейших программ	1				
4	Создание простейших программ	1				
5	Переменные, присваивание значений, целочисленная арифметика	1				
6	Переменные, присваивание значений, целочисленная	1				

	арифметика					
7	Переменные, присваивание значений, целочисленная арифметика	1				
8	Переменные, присваивание значений, целочисленная арифметика	1				
9	Переменные, присваивание значений, целочисленная арифметика	1				
10	Условная инструкция, логические операции.	1				
11	Условная инструкция, логические операции.	1				
12	Условная инструкция, логические операции.	1				
13	Условная инструкция, логические операции.	1				
14	Цикл for	1				
15	Цикл for	1				
16	Цикл for	1				
17	Действительная арифметика, операции с действительными числами.	1				
18	Действительная арифметика, операции с действительными числами.	1				
19	Действительная арифметика, операции с действительными числами.	1				
20	Действительная арифметика, операции с действительными числами.	1				
21	Действительная арифметика, операции с действительными числами.	1				
22	Строковый тип данных и операции со строками	1				
23	Строковый тип данных и операции со строками	1				

24	Строковый тип данных и операции со строками	1				
25	Строковый тип данных и операции со строками	1				
26	Строковый тип данных и операции со строками	1				
27	Строковый тип данных и операции со строками	1				
28	Строковый тип данных и операции со строками	1				
29	Цикл while и обработка последовательностей	1				
30	Цикл while и обработка последовательностей	1	1			
31	Цикл while и обработка последовательностей	1				
32	Цикл while и обработка последовательностей	1				
33	Цикл while и обработка последовательностей	1				
34	Цикл while и обработка последовательностей	1				
35	Цикл while и обработка последовательностей	1				
36	Цикл while и обработка последовательностей	1				
37	Функции, рекурсия	1				
38	Функции, рекурсия	1				
39	Функции, рекурсия	1				
40	Функции, рекурсия	1				
41	Функции, рекурсия	1				
42	Функции, рекурсия	1				
43	Функции, рекурсия	1				
44	Функции, рекурсия	1				
45	Списки	1				
46	Списки	1				
47	Списки	1				
48	Списки	1				
49	Списки	1				
50	Списки	1				
51	Квадратичные алгоритмы сортировки	1				
52	Квадратичные алгоритмы сортировки	1				

53	Квадратичные алгоритмы сортировки	1				
54	Квадратичные алгоритмы сортировки	1				
55	Квадратичные алгоритмы сортировки	1				
56	Квадратичные алгоритмы сортировки	1				
57	Квадратичные алгоритмы сортировки	1				
58	Квадратичные алгоритмы сортировки	1				
59	Двумерные массивы	1				
60	Двумерные массивы	1				
61	Двумерные массивы	1				
62	Двумерные массивы	1				
63	Двумерные массивы	1				
64	Двумерные массивы	1				
65	Двумерные массивы	1				
66	Двумерные массивы	1				
67	Генераторы списков	1				
68	Генераторы списков	1	1			
69	Генераторы списков	1		1		
70	Генераторы списков	1				
71	Генераторы списков	1				
72	Генераторы списков	1				
73	Генераторы списков	1				
74	Генераторы списков	1		1		
75	Арифметические алгоритмы и задачи	1				
76	Арифметические алгоритмы и задачи	1				