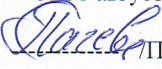




**МОУ «Новомичуринская средняя общеобразовательная школа №2  
им. И.В. Мичурина»**

<p>Согласовано заместителем директора по ВР от 31 августа 2023 г.  Пачева С.А./</p>	<p>Программа рассмотрена и утверждена на заседании педагогического совета Протокол №1 от 30 августа 2023г.</p>	<p>Утверждаю Директор МОУ «НСОШ №2» приказ № 76 от 31 августа 2023г.  В.Н. Климакина/ </p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Дополнительная общеразвивающая  
общеобразовательная программа**

**«Юный программист»**

**Антошкиной Надежды Алексеевны,  
учителя информатики  
высшей квалификационной категории**

**2023 - 2024 уч. год.**

## Содержание

Пояснительная записка .....	3
Учебно-тематический план.....	8
Содержание программы.....	9
Планируемые результаты .....	13
Календарно-тематическое планирование .....	15
Условия реализации программы .....	24
Методические материалы .....	25
Список литературы .....	32

### **Пояснительная записка**

Изучение основ программирования связано с развитием целого ряда таких умений и навыков, которые носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач. Изучение программирования развивает мышление школьников, способствует формированию у них многих приемов умственной деятельности. Изучая программирование на Паскале, учащиеся прочнее усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

Процесс развития аппаратного и программного обеспечения и оснащения им школ за последние годы существенно изменил курс информатики. Основное внимание стало уделяться информационным технологиям. Эти тенденции отражены и в новом «Стандарте» по информатике. В рамках часов, отводимых программой базового курса информатики на алгоритмизацию и

программирование, дается явно недостаточно времени, а школьники, которые проявляют интерес к данному вопросу, безусловно, есть. Программа кружка по информатике «Юный программист» расширяет базовый курс информатики, дает возможность познакомиться с интересными нестандартными вопросами.

**Новизна программы** состоит в более углубленном изучении основ программирования. Знания по программированию воспитанник получает в контексте практического применения, это дает возможность изучать теоретические вопросы в их деятельно-практическом аспекте.

**Актуальность программы заключается в следующем:** впечатляющие успехи информатики, которые мы наблюдаем сегодня, связаны с реализацией на персональном компьютере большого количества алгоритмов. Умение эффективно использовать реализованные алгоритмы вырабатывается полноценным усвоением идей и методов программирования. Наступило такое время, когда человека нельзя назвать образованным, если он не знает, как работать на компьютере и не знаком хотя бы с одним языком программирования.

В программе представлены темы, выходящие за рамки традиционного курса программирования: обработка символьной информации, работа с файлами.

Позитивно влияют на изучение программирования и олимпиады по информатике, значимость которых возрастает в связи с новыми правилами приема в вуз, соответственно возрастает роль, которую помогает выполнить кружок по программированию.

Программа имеет научно – техническую **направленность**, так как ее содержание способствует развитию алгоритмического мышления школьников, формированию многих общеучебных, общеинтеллектуальных умений и навыков. Изучая программирование на Python, учащиеся прочнее усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

**Цель программы:** овладение умениями и навыками программирования на языке Pascal как основы развития алгоритмического и логического мышления детей среднего и старшего подросткового возраста.

**Задачи программы:**

**Обучающие:**

- сформировать представление об алгоритме, основных алгоритмических структурах;
- изучить основы алгоритмизации и программирования с помощью языка Pascal;
- обучить приемам написания и отладки программ разного уровня сложности;
- сформировать навыки проектной деятельности, конструирования.

### **Развивающие:**

способствовать развитию алгоритмического мышления воспитанников с помощью изучения основ алгоритмизации и программирования;

способствовать развитию познавательных интересов, творческих способностей;

способствовать развитию творческого и познавательного потенциала воспитанников.

### **Воспитательные:**

- воспитать трудолюбие, самостоятельность, ответственность, активность, аккуратность;
- формировать умение планировать деятельность, ставить цели и выделять главное для решения задачи;
- воспитать культуру общения.

**Возраст детей, участвующих в реализации программы, 14-16 лет.**

**Срок реализации** данной дополнительной образовательной программы – 21 год. Общее количество часов, необходимое для реализации программы, - 64 (по 2 часов в неделю) .

Формы организации образовательного процесса:

### **Групповые формы.**

Ученики работают в группах или в парах. Эту форму работы удобно использовать, при освоении новых программных средств, при работе над проектами, при недостаточном количестве компьютеров. Ученики обмениваются друг с другом информацией, вместе обсуждают задачу, оценивают решение каждого. Сверяют свои ответы и если допущены ошибки, то пытаются вместе найти ответ. Усвоение знаний и умений происходит результативнее при общении учащихся с более подготовленными товарищами.

Надо отметить, что преимущественно групповой работы в том, что ученик учится высказывать и отстаивать собственное мнение, прислушиваться к мнению других, сопоставлять, сравнивать свою точку зрения с точкой зрения других. Вырабатываются навыки контроля над действиями других и самоконтроля, формируется критическое мышление. Групповое обсуждение, дискуссия оживляют поисковую активность учащихся.

### **Дифференцированно - групповая форма.**

Ученики отличаются друг от друга умственной гибкостью, активностью, самостоятельностью мышления. Одни способны перебирать многообразие способов решения задач, чтобы найти верный путь решения. Другие привыкают работать по шаблону и не пытаются искать других подходов.

Для организации учебного процесса необходимо распределить воспитанников на несколько групп: по уровню знаний, интересам, способностям и подобрать задания в соответствии с выявленными уровнями знаний, интересами, способностями учащихся. Заданиями могут быть следующими: с различными условиями, допускающие одинаковые, с точки зрения информатики, решения; взаимодополняющие задания с различными условиями; уровневые взаимодополняющие задания.

Дифференцированная форма обучения развивает у учащихся устойчивый интерес к предмету, формирует умение самостоятельно работать, заметно развивает навыки работы с учебным программным средством.

## Индивидуальные и парные формы.

При подборе заданий для индивидуальной самостоятельной работы учитываются уровни усвоения знаний учащимися: репродуктивный, репродуктивно - творческий, творческий. Работая один на один с компьютером (а точнее с программой), обучающийся в своем темпе овладевает знаниями, сам выбирает индивидуальный маршрут изучения учебного материала в рамках заданной темы занятия.

В парном обучении взаимодействие происходит между двумя учениками, которые могут обсуждать задачу, осуществлять взаимообучение или взаимоконтроль. Очень часто для учащегося помощь товарища оказывается полезнее, чем помощь учителя.

### Учебно-тематический план

№		Количество часов
1.	Вводное занятие.	1
2.	Основы алгоритмизации.	1
3.	Простые программы на Python. Числовые данные.	2
4.	Работа с символами.	4
5.	Джордж Буль и его логика.	4
6.	Анализ ситуации и последовательность выполнения команд.	2
7.	Многократно повторяющиеся действия.	4
8.	Циклы с условием.	8
9.	Массивы - структурированный тип данных.	14
10.	Вспомогательные алгоритмы.	6

11	Процедуры и функции.	6
12	Работа с символьными строками.	8
13	Процедуры и функции с параметрами.	4
14	Файлы.	4

## **Содержание программы.**

### **1. Вводное занятие.**

Теоретическая часть. Проведение техники безопасности в компьютерном классе. Рассмотреть важность и актуальность владения программированием в современном мире.

### **2. Основы алгоритмизации.**

**Теоретическая часть.** Понятие алгоритма. Основные алгоритмические структуры. Блок-схемы алгоритмов.

Практическая часть. Разработка и построение блок-схем алгоритмов.

### **3. Простые программы на Python.**

**Теоретическая часть.** Запуск программы. **Организация вывода сообщений** на экран дисплея. Сохранение программ в компьютере. Оформление текста на экране.

Практическая часть. Практическая работа: знакомство с окном программы Python . Закрепление рассмотренных вопросов на практике.

Числовые данные.

**Теоретическая часть. Знакомство учащихся с особенностями работы с целыми и вещественными числами. Совмещение целых и вещественных чисел. Организация констант в программе. Операторы ввода и вывода данных.**

**Практическая часть. Структура программы на языке Python. Организация ввода и вывода данных.**

4. Работа с символами. **Теоретическая часть. Знакомство с кодовыми таблицами (ASCII, Unicode и другие). Использование типа Char.**

5. Работа с символами. **Теоретическая часть. Знакомство с кодовыми таблицами (ASCII, Unicode и другие). Использование типа Char.**

**Практическая часть. Применение изученного материала на практике. Работа с данными символьного типа.**

5. Джордж Буль и его логика.

**Теоретическая часть. Рассмотреть данные логического типа. Логические переменные, константы и логические операции.**

**Практическая часть. Использование и особенности записи логических величин и логических операций в программе.**

6. Анализ ситуации и последовательность выполнения команд.

**Теоретическая часть. Проверка условия и ветвления. Блоки операторов. Ветвления по ряду условий.**

**Практическая часть. Организация полных и неполных ветвлений в программе.**

7. Многократно повторяющиеся действия.

**Теоретическая часть. Знакомство учащихся с оператором цикла FOR. Применение циклов со счетчиком.**

**Практическая часть. Организация циклических программ со счетчиком.**

8. Циклы с условием.

**Теоретическая часть. Циклы с предусловием. Циклы с постусловием.**



**Практическая часть. Особенности составление циклических программ с предусловием и с постусловием.**

9. Массивы - структурированный тип данных.

**Теоретическая часть. Понятие массива. Одномерные и двумерные массивы.** Заполнение и вывод массива. Поиск в массиве. Вычисление суммы и количества элементов. Поиск минимального и максимального элементов в массиве.

**Практическая часть. Отработка алгоритмов ввода и вывода элементов одномерного и двумерного массивов; поиск в массиве заданных элементов; вычисление суммы, произведения и количества элементов массива; нахождение минимальных и максимальных элементов. Работа с элементами главной и побочной диагонали двумерного массива.**

10. Вспомогательные алгоритмы. Процедуры и функции.

**Теоретическая часть.** Алгоритм «сверху – вниз».

11. Понятие процедуры и функции.

**Практическая часть. Работа с процедурой и функцией.**

12. Работа с символьными строками.

**Теоретическая часть. Символьный тип String.** Функции и процедуры работы со строками.

**Практическая часть. Работа со строковыми данными и функциями.**

13. Процедуры и функции с параметрами.

**Теоретическая часть.** Использование подпрограмм с параметрами. Способы передачи параметров.

**Практическая часть. Работа с процедурами и функциями с параметрами.**

14. Файлы.

**Теоретическая часть.** Работа с текстовым файлом. Сохранение двумерного массива чисел в текстовом файле.

**Практическая часть.** Создание входного и выходного файла.

