



МУНИЦИПАЛЬНОЕ КАЗЕННОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №9» С. ВОЗДВИЖЕНСКОЕ
АПНАСЕНКОВСКОГО МУНИЦИПАЛЬНОГО ОКРУГА
СТАВРОПОЛЬСКОГО КРАЯ

| | | |
|--|--|---|
| <p>«Рассмотрено» на педагогическом совете протокол № <u>1</u> от «<u>30</u>» <u>08</u> 2024 г.</p> | <p>«Согласовано» Руководитель центра образования естественно- научного и технологического направленностей «Точка роста»  <u>И.А. Галяцкая</u> подпись Ф.И.О «<u>30</u>» августа 2024 г.</p> | <p>«Утверждено» приказ № <u>1/2024</u> от «<u>30</u>» августа 2024 г. ВРИО директора школы  <u>Галяцкая И.А.</u> подпись Ф.И.О</p>  |
|--|--|---|

Рабочая программа курса дополнительного образования

«Юный программист»

(центра образования естественно-научного и технологического
направленностей «Точка роста»)

Направление: технологическое

Возраст обучающихся: 12-14 лет

Срок реализации: 1 год

Составитель: Галяцкая Ирина Александровна

2024-2025 учебный год

Содержание

| | |
|---|------------|
| 1. Пояснительная записка | 3 |
| 2. Учебно-тематический план..... | 8 |
| 3. Содержание программы..... | 9 |
| 4. Планируемые результаты | 13 |
| 5. Календарно-тематическое планирование | 15 |
| 6. Условия реализации программы | 24 |
| 7. Методические материалы | 25 |
| 8. Оценочные материалы | 27 |
| 9. Список | 32 |
| | литературы |
| 10. Приложения | 34 |

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 воспитанникам 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 программирование 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 Паскале, учащиеся прочнее

усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

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

Задачи программы:

Обучающие:

- сформировать у воспитанников 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.

Отличительной особенностью данной дополнительной образовательной программы от уже существующих является интеграция курса «Юный программист» с курсом объектно-ориентированного программирования.

Возраст детей, участвующих в реализации программы, 12-14 лет.

Срок реализации данной дополнительной образовательной программы – 1 год. Общее количество часов, необходимое для реализации программы, - 204 (по 6 часов в неделю).

Формы организации образовательного процесса:

Групповые формы.

Воспитанники работают в группах или в парах. Эту форму работы удобно использовать, при освоении новых программных средств, при работе над проектами, при недостаточном количестве компьютеров. Воспитанники 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.

Воспитанники 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.

Ожидаемые результаты и способы определения результативности

Учебный уровень достижений:

Обучающиеся должны знать:

- о концепциях и идеях структурного программирования;
- алгоритмические конструкции языка программирования Pascal;
- возможности инструментальных средств системы Pascal;
- основные приемы написания программ-приложений;
- требования к написанию и оформлению программ-приложений;
- типы данных и их представление в памяти компьютера, операции над данными основных типов;
- способы представления одномерных и двумерных массивов и строк;
- различие между текстовыми и бинарными файлами, особенности организации текстовых файлов;
- назначение и способы организации проектов.

Обучающиеся должны уметь:

- использовать все доступные источники (интерактивные компьютерные справочные системы, книги, справочники, технические описания) для самостоятельного решения задач с помощью компьютеров;
- составлять алгоритмы в словесной форме для решения разнообразных задач;
- применять метод пошаговой детализации при составлении алгоритмов;
- переводить алгоритмы на язык программирования;
- составлять алгоритмы и программы для новых методов решения задач;
- работать с различными структурами данных (массив, запись, файл, множество);
- решать поставленную задачу, реализовывать алгоритмические конструкции на языке программирования Pascal;
- правильно интерпретировать получаемые результаты в ходе тестирования и отладки программных продуктов.

Личностный уровень достижений

Данный уровень можно отследить посредством диагностики, которая направлена на изучение динамики таких познавательных процессов, как алгоритмическое и логическое мышление, а также творческих способностей.

Формами подведения итогов реализации дополнительной образовательной программы является активное участие воспитанников в международных, всероссийских, областных и районных конкурсах по информатике: международная олимпиада по основам наук, всероссийские дистанционные конкурсы «КИТ», «Инфознайка», интернет - олимпиады по программированию

Принципы построения программы

- 1) Принцип доступности – при изложении материала учитываются возрастные особенности детей, один и тот же материал по-разному преподаётся, в зависимости от возраста и субъективного опыта детей. Материал располагается от простого к сложному. При необходимости допускается повторение части материала через некоторое время.
- 2) Принцип наглядности – человек получает через органы зрения почти в 5 раз больше информации, поэтому на занятиях используются как наглядные материалы, так и обучающие, тестирующие программы.
- 3) Принцип развития выражается в возможности постоянного расширения и обновления системы задач, решаемых с помощью программирования и средств их достижения.
- 4) Принцип сознательности и активности – для активизации деятельности детей используются такие формы обучения, как занятия-игры, конкурсы, совместные обсуждения поставленных вопросов и свободное творчество.
- 5) Принцип индивидуализации базируется на том, что эффективность обучения прямо пропорциональна индивидуализации деятельности учащихся в ходе обучения.
- 6) Принцип практической направленности – в ходе обучения обучающиеся выполняют творческие проекты, разрабатывают собственные программы и внедряют их.
- 7) Принцип вариативности предоставляет педагогу возможность варьировать программу с учетом особенностей восприятия ее воспитанниками.

Формы подведения итогов реализации программы

Главный показатель – личностный рост каждого ребенка, его творческих способностей, превращение группы в единый коллектив, способный к сотрудничеству и совместному творчеству.

Проверка эффективности данного курса осуществляется через итоговые занятия. По окончании обучения по данной образовательной программе, учащиеся должны уметь создавать программы разного уровня сложности. Для оценки достижения обязательной подготовки целесообразно использовать дихотомическую шкалу типа зачет или незачет, анализ детских работ, определяющий творческий рост школьника, а также педагогическое наблюдение.

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 Паскале.

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 ABC Pascal. Закрепление рассмотренных вопросов на практике.

4. 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 Паскаль. Организация ввода и вывода данных.

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.

6. 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.

7. 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.**

8. Многократно повторяющиеся действия.

Теоретическая часть. **Знакомство учащихся с оператором цикла FOR.**

Применение циклов со счетчиком.

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

9. Циклы с условием.

Теоретическая часть. Циклы с предусловием. Циклы с постусловием.
Практическая часть. Особенности составления циклических программ с предусловием и с постусловием.

10. 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.

11. Вспомогательные алгоритмы. Процедуры и функции.

Теоретическая часть. Алгоритм «сверху – вниз». 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.

15. Графический режим работы.

Теоретическая часть. Особенности работы с графикой. Графические операторы.

Практическая часть. Работа с графическими операторами.

16. Повторение материала.

Теоретическая часть. Повторение пройденного материала.

Практическая часть. Работа с графическими и строковыми функциями. Работа с массивами.

17. Выполнение проектов.

Теоретическая часть. Правила организации творческих проектов учащихся.

Практическая часть. Выполнение творческих проектов учащимися.

Планируемые результаты

Личностные результаты

- 1) сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
- 2) готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
- 3) навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;
- 4) эстетическое отношение к миру, включая эстетику научного и технического творчества;
- 5) осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

Метапредметные результаты

- 1) умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
- 2) умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;
- 3) владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;
- 4) готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных

источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

5) умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Предметные результаты

1) владение системой базовых знаний, отражающих *вклад информатики* в формирование современной научной картины мира;

2) владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов;

3) овладение понятием *сложности алгоритма*, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;

4) владение стандартными приёмами *написания на алгоритмическом языке программы* для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации;

5) владение *универсальным языком программирования высокого уровня* (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;

6) владение умением *понимать программы*, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц;

7) владение навыками и опытом *разработки программ* в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ.

Календарно-тематическое планирование

| № | Дата по плану | Дата факт | Наименование разделов и тем | Общее кол-во учебных часов |
|-----|---------------|-----------|---|----------------------------|
| 1. | | | Инструктаж по технике безопасности. Алгоритм, свойства алгоритма | 2 |
| 2. | | | Формы записи алгоритмов | 2 |
| 3. | | | Разработка алгоритмов, работа с готовыми алгоритмами | 2 |
| 4. | | | Разработка алгоритмов, работа с готовыми алгоритмами | 2 |
| 5. | | | Линейные алгоритмы | 2 |
| 6. | | | Алгоритмы с ветвлениями | 2 |
| 7. | | | Циклические алгоритмы | 3 |
| 8. | | | Вывод сообщений на экране | 2 |
| 9. | | | Этапы создания компьютерной программы | 2 |
| 10. | | | Работа в среде редактирования | 2 |
| 11. | | | Работа в среде редактирования | 2 |
| 12. | | | Запуск компилятора | 2 |
| 13. | | | Выполнение программы | 2 |
| 14. | | | Оформление текста на экране | 2 |
| 15. | | | Оформление текста на экране | 2 |
| 16. | | | Понятие переменной Integer | 2 |
| 17. | | | Стандартные функции типа Integer | 2 |
| 18. | | | Вещественный тип данных | 2 |
| 19. | | | Форматы записи вещественных переменных | 2 |
| 20. | | | Разработка алгоритмов, работа с готовыми алгоритмами | 2 |
| 21. | | | Вещественные операции | 2 |
| 22. | | | Стандартные функции типа Real | 2 |
| 23. | | | Преобразование типов | 2 |
| 24. | | | Разработка программ | 2 |
| 25. | | | Разработка программ | 2 |
| 26. | | | Константы | 2 |
| 27. | | | Константы | 2 |
| 28. | | | Составление программ | 2 |
| 29. | | | Кодовые таблицы | 2 |
| 30. | | | Кодовые таблицы | 2 |
| 31. | | | Тип Char | 2 |
| 32. | | | Стандартные функции | 2 |
| 33. | | | Логический тип данных Boolean | 2 |
| 34. | | | Логический тип данных Boolean | 2 |
| 35. | | | Операции отношения | 2 |
| 36. | | | Ввод, вывод булевских переменных | 2 |
| 37. | | | Логические операции | 2 |
| 38. | | | Логические операции | 2 |
| 39. | | | Логические операции (исключающее или, инверсия) | 2 |
| 40. | | | Разработка программ с логическими операциями | 2 |
| 41. | | | Разработка программ с логическими операциями | 2 |
| 42. | | | Ветвления в алгоритме | 2 |
| 43. | | | Сложные ветвления в алгоритме | 2 |
| 44. | | | Разработка программ | 2 |
| 45. | | | Разработка программ | 2 |
| 46. | | | Полная форма оператора if | 2 |
| 47. | | | Неполная форма оператора if | 2 |

Условия реализации программы.

Важнейшим условием реализации программы является создание развивающей, здоровьесберегающей образовательной среды как комплекса комфортных, психолого-педагогических и социальных условий, необходимых для развития творческих интересов и способностей детей.

Материально-техническое обеспечение:

- компьютерный класс с 16 персональными компьютерами для обучающихся;
- локальная сеть с доступом в Интернет;
- проектор и демонстрационный экран;
- доска школьная.

Программное обеспечение для компьютеров: Pascal ABC

Список литературы для педагога:

1. Андреева Т. А., Городня Л. В. Задачи по теме «Линейные алгоритмы». Информатика и образование №2, 2002г. – 97 с.
2. Босова Л. Л., Розова В. М. Разноуровневые дидактические материалы по информатике. Информатика в уроках и задачах №3, 2001 г. – 115 с.
3. Босова Л. Л., Розова В. М. Разноуровневые дидактические материалы по информатике. Информатика в уроках и задачах №4, 2001 г. – 127 с.
4. Джагаров Ю. А. Планирование темы «Циклы с заданным числом повторений». Информатика и образование №6, 2000 г. – 98 с.
5. Златопольский Д. М. Я иду на урок информатики. Задачи по программированию. – М. Первое сентября, 2002 г. – 207 с.
6. Лапчик М. П., Семакин И. Г., Хеннер Е. К. Методика преподавания информатики. – М: АСАДЕМА, 2003 г. – 623.
7. Магдюков В. С. Задачи по информатике. Информатика в школе №2, 2002г. – 80 с.
8. Ракитина Е. А., Галыгина И. В. Контрольная работа по теме «Алгоритмизация и программирование. Информатика в школе №1, 2003 г. – 103 с.
9. Ракитина Е. А., Галыгина И. В. Решение типовых задач по информатике. Информатика в школе №1, 2004 г. – 151 с.

10. Сулейманов Р. Р. Составление задач учащимися. Информатика и образование 36, 2000 г. – 98 с.
11. Сулейманов Р. Р. Некоторые вопросы методики обучения решению задач по программированию. Информатика и образование №12, 2004 г. – 99 с.
12. Чернов А. А. Конспекты уроков информатики в 9-11 классах. – В.: Учитель, 2004 г. – 235 с.

Список литературы для воспитанников:

1. Гейн А. Г., Сенокосов А. И. Информатика и информационные технологии. Учебник для 9 класса общеобразовательных учреждений. – М.: Просвещение, 2006 г. – 298 с.
2. Информатика. Задачник-практикум: В 2 т./ Под ред. И.Г. Семакина: Т.1. М.: БИНОМ. Лаборатория знаний, 2002.
3. Житкова О. А., Кудрявцева Е. К. Справочные материалы по программированию на языке Паскаль. – М.: «Интеллект-центр», 2002 г. – 80 с.
4. Ларина Э. С. Олимпиадные задания с решениями 9-11 классы. – В.: Учитель, 2006 г. – 111 с.
5. Ларина Э. С. Ларина «Создание программ на языке Паскаль» - В.: Учитель, 2008 г.
6. Ляхович В. Ф. Основы информатики. – Р.: ЕНИКС, 2003 г.
7. Окулов С.М. Основы программирования. – М.: Юнимедиастилл, 2002.
8. Ушаков Д. М., Юркова Т. А. «Паскаль для школьников» - М.: Питер, 2008 г.
9. Чернов А. А., Чернов А. Ф. «Контрольные и самостоятельные работы по программированию» - В.: «Учитель», 2009 г.

Приложение **Диагностика результатов**

Система отслеживания результатов программы

Для определения исходного уровня развития алгоритмического мышления учащихся мною была разработана входная диагностика.

Диагностика № 1

1. Решить задачу и объяснить ход рассуждений:

1.1. Имеется пять монет, среди которых одна фальшивая (легче других). Придумайте способ нахождения фальшивой монеты за минимальное число взвешиваний на чашечных весах без гирь.

1.2. Из четырех внешне одинаковых деталей одна отличается по массе от трех остальных, однако неизвестно, больше ее масса или меньше. Как выявить эту деталь двумя взвешиваниями на чашечных весах без гирь?

2. Как с помощью 7-литрового ведра и 3-литровой банки налить в кастрюлю ровно 5л воды?

3. Имеются два кувшина емкостью 3л и 8л. Запишите алгоритм, выполняя который можно набрать из реки 7 л воды (другими емкостями не пользоваться).

4. Решить задачу и объяснить ход рассуждений:

По каналу плывут одна за другой лодки *А* и *Б*. Навстречу им движутся одна за другой лодки *В* и *Г*. Канал настолько узок, что две лодки в нем

разъехаться не могут, однако в нем с одной стороны есть небольшой залив, куда может поместиться только одна лодка. Могут ли лодки разъехаться так, чтобы продолжать свой путь в прежнем порядке?



1. Четыре белые и четыре черные шашки расположены так, как показано на рисунке:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Требуется переставить белые шашки на клетки с номерами 1, 2, 3, 4, а черные – на клетки с номерами 6, 7, 8, 9 с соблюдением следующих условий:

- 1) каждая шашка может перескочить на ближайшую клетку или через одну клетку, но не дальше;
- 2) никакая шашка не должна возвращаться на клетку, где она уже побывала;
- 3) в каждой клетке не должно быть более одной шашки;
- 4) начинать ходы надо с белой шашки.

| Критерии | Уровни | | |
|---|--|--|---|
| | Высокий | Средний | Низкий |
| 1. Степень осознанности приемов алгоритмического мышления | -глубокое осознанное владение алгоритмическим языком, алгоритмическими структурами; - умение видеть в примерах алгоритмическую структуру; -осознанное использование алгоритмических знаний и умений при решении задач; | - фрагментарное осознанное владение алгоритмическим языком, алгоритмическими структурами; - фрагментарное осознанное использование алгоритмических знаний и умений при решении задач; | -слабая осознанность знаний алгоритмического языка; -слабое осознанное использование алгоритмических знаний при решении задач. |

| | | | |
|---|--|--|---|
| 2. Степень владения приемами алгоритмического мышления. | -свободное использование знаний алгоритмического языка при решении проблемных задач из различных дисциплин. | -частичные затруднения при решении проблемных задач из различных дисциплин. | -формальное владение приемами алгоритмического мышления при решении проблемных задач. |
| 3. Тезаурус алгоритмического мышления | -системность знаний, способность видеть альтернативные решения; -самостоятельность в дальнейшем развитии алгоритмического мышления. | -системность знаний; -частичные затруднения при выборе альтернативных решений; - для получения новых знаний требуется иногда помощь учителя. | -несистемный набор знаний; - затруднения при выборе альтернативных решений; - получение новых знаний с помощью учителя. |

Для каждого воспитанника выставлялась оценка (от 1 до 6 баллов) по каждому критерию (приложение 1), затем вычислялся обобщенный показатель развития алгоритмического мышления и по оценкам определялся соответствующий уровень: низкий уровень (1-3 баллов), средний (4-5 баллов), высокий (6 баллов)

Методика обучения алгоритмизации.

Задачи на составление алгоритмов можно разбить на четыре типа:

- найти ошибку в алгоритме;
- определить, каков результат выполнения алгоритма;
- сформулировать условие задачи;
- построить математическую модель, составить алгоритм.

Рассмотрим примеры задач первого типа:

1) Найти ошибку в алгоритме:

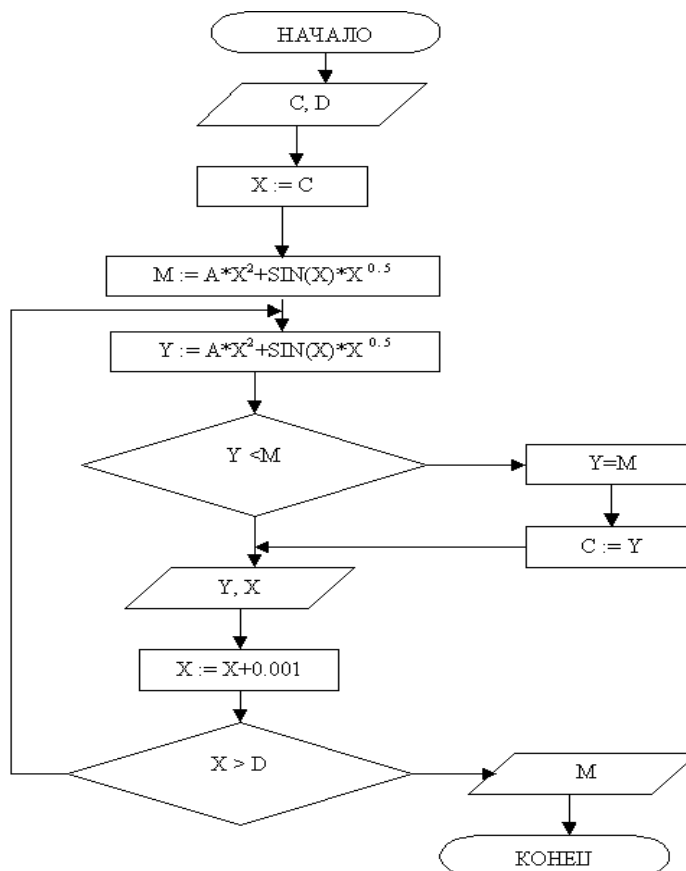
- а) значение b умножить на два;
- б) a присвоить значение b ;
- в) вывести значение b на экран.

2) Злоумышленник поменял местами действия в алгоритме вычисления среднего арифметического квадратов трех чисел.

- а) Присвоить a значение $(a^2+b^2+c^2)/3$.
- б) Запросить a, b, c .
- в) Сообщить "Среднее арифметическое квадратов равно".
- г) Сообщить a .

Восстановить правильный порядок чисел.

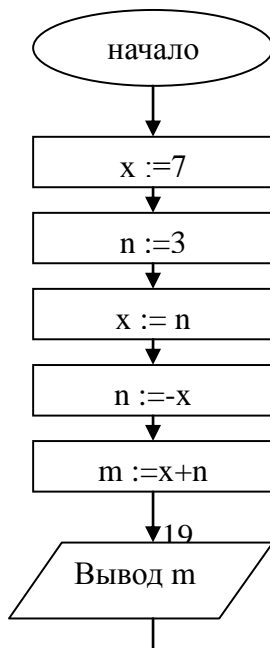
3) Определите местонахождение ошибок в алгоритмическом решении следующей задачи: найти минимальное значение функции $Y=A*X^2+\sin(X)*X^{0.5}$, для X изменяющемся на отрезке $[C,D]$ с шагом 0,01.



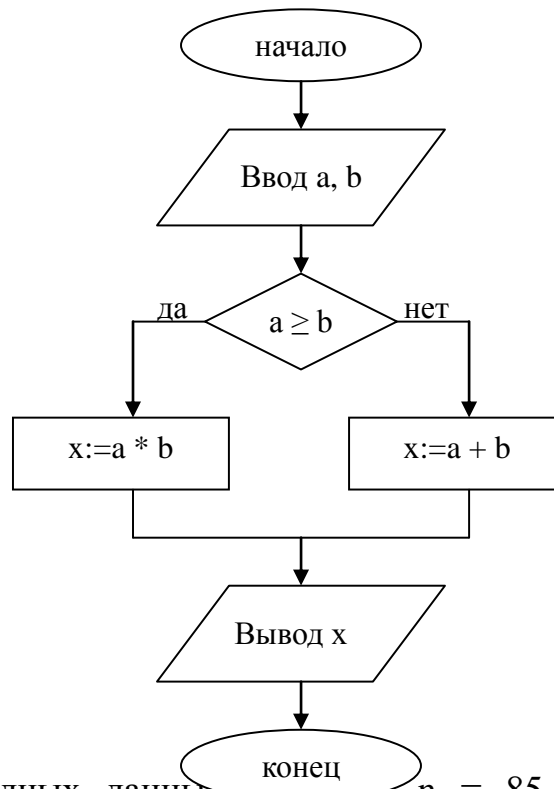
Для развития алгоритмического мышления воспитанников было важно постоянно тренироваться в упражнениях, которые имели алгоритмические ошибки. Важным этапом использования таких упражнений был анализ ошибок, который ставил обучающихся в рефлексивную позицию и способствовал процессу развития алгоритмического мышления воспитанников.

Рассмотрим несколько заданий, относящихся ко второму типу.

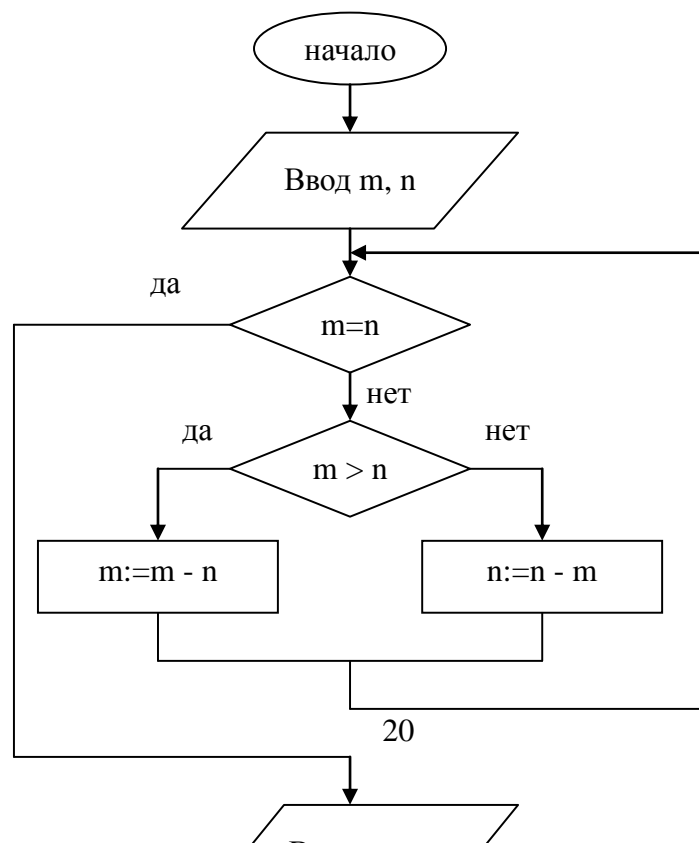
1) Определите, что получится в результате выполнения алгоритма, описанного блок-схемой.



2) При исходных данных $a = 5$, $b = 4$ определите результат выполнения алгоритма, изображенного в виде блок – схемы.



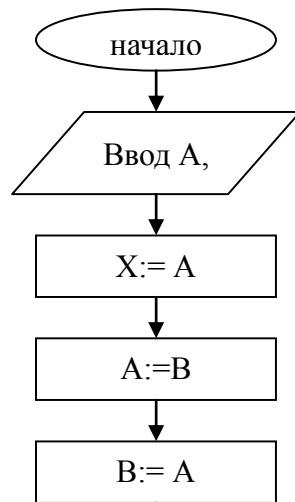
3) При исходных данных $m = 10$, $n = 85$ определите результат выполнения алгоритма, изображенного в виде блок – схемы.



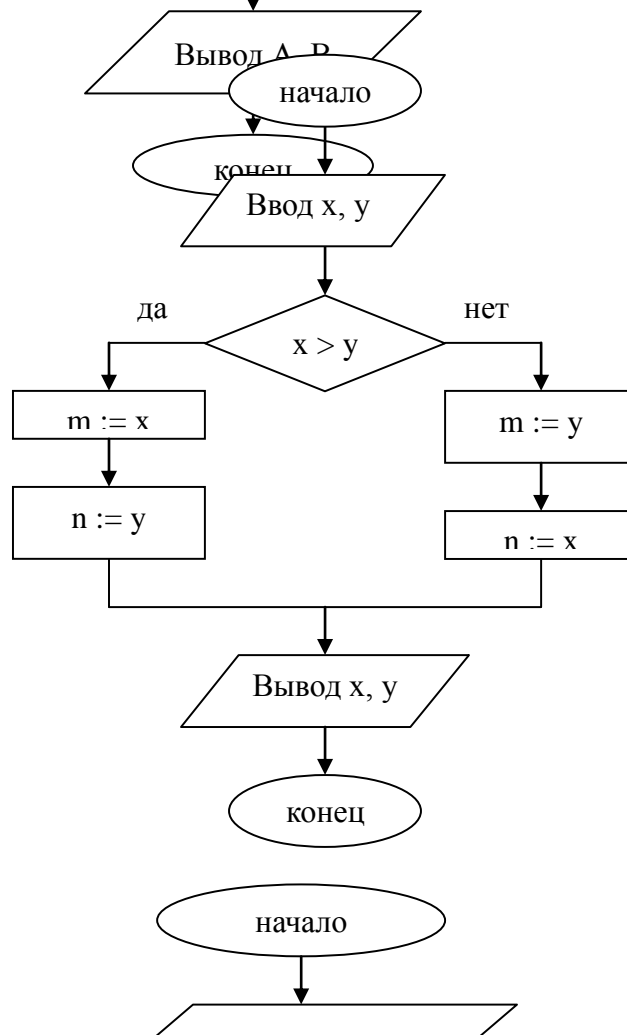
Задания, относящиеся к третьему типу:

По предложенной блок-схеме сформулируйте условие задачи.

1. а)



б)



в)

| | | | | |
|----|----|-----|-----|-----|
| | да | | нет | |
| да | | нет | | да |
| | | | | нет |

В качестве примера задачи четвертого типа можно использовать алгоритм игры Баше.

Правила игры определены так: в игре используются 11, 16, 21, 26 предметов. За один ход можно брать от 1 до 4 предметов. Проигрывает тот игрок, который берет последний предмет.

После того как ученики поиграли в игру по этим правилам, можно предложить им несколько заданий аналитического характера на тему игры Баше. Задания могут быть предложены в качестве домашней работы.

- 1) «Разгадать загадку» алгоритма, т.е. объяснить, почему второй игрок всегда выигрывает?
- 2) Составить алгоритм, по которому игрок, делающий первый ход, может выиграть только в том случае, если соперник не знает выигрышной тактики.
- 3) Попробуйте провести математический анализ игры Баше в общем случае для N камней. Определите правила игры (т.е. сколько камней можно брать за один ход), при котором имеется выигрышный алгоритм. Опишите этот алгоритм в виде последовательности команд.

Использование подобных игр приводит к выработке у воспитанников умения анализировать, сравнивать, т.е. логически и алгоритмически мыслить, обосновывать выдвигаемые положения.

Методика обучения программированию.

Для решения задачи с помощью компьютера (в средах программирования) знакомлю учащихся с этапами решения задач (к этапам, которые присутствуют при составлении алгоритмов, добавляются этапы, необходимость которых обусловлена тем, что реализация модели будет осуществляться техническим средством).

Этапы решения задачи на компьютере с использованием среды программирования.

| № | Название этапа | Содержание этапа |
|----------|--|---|
| 1 | Постановка задачи Формализация условий задачи | <ul style="list-style-type: none">• Что дано?• Что требуется?• Что будет считаться результатом (решением задачи)?• Каким условиям должны удовлетворять исходные данные?• Каким условиям должен удовлетворять результат? |

| | | |
|----|--|--|
| 2 | Выбор метода решения | Выбор метода решения зависит от точности, с которой мы хотим получить результат. |
| 3 | Построение математической модели | Установление связи между исходными данными и результатом (формулы, соотношения, неравенства и т. п.). Модель всегда основана на некотором упрощении. При построении модели также учитывается точность, с которой мы хотим получить результат. |
| 4 | Составление сценария работы с ЭВМ | <p>Определение:</p> <ul style="list-style-type: none"> • порядка ввода данных, • формы представления промежуточных результатов, • формы вывода результата, • реакции программы на вводимые команды и данные. |
| 5 | Разработка (конструирование) алгоритма | <p>При разработке алгоритма следует учитывать:</p> <ul style="list-style-type: none"> • имеющиеся ресурсы памяти, • возможности компьютера, • какие операции доступны для исполнения. |
| 6 | Перевод алгоритма в программу | Программа — алгоритм, закодированный по правилам выбранного языка программирования. Один и тот же алгоритм на разных языках программирования будет записан по-разному. |
| 7 | Ввод программы в память компьютера | Работа во встроенном текстовом редакторе среды программирования. |
| 8 | Отладка программы | Отладка — устранение синтаксических и семантических (смысловых) ошибок. Такого рода ошибки обнаруживаются на этапе компилирования программы. |
| 9 | Тестирование программы | Тестирование — устранение логических ошибок. Неоднократное выполнение программы с такими исходными данными, при которых результаты работы легко проверить. |
| 10 | Получение и анализ результатов | Получение результата решения задачи, для которой создавался алгоритм. |

Учащиеся должны усвоить, что многократное тестирование еще не гарантирует безошибочной программы, а единственная ошибка при тестировании говорит о наличии ошибки в программе (или алгоритме).

Ни одну более или менее сложную программу нельзя считать правильной и процесс ее написания законченным, если он не проверен путем исполнения. Велика обучающая роль исполнения программ - это, в конце концов, приводит к сознательному и прочному усвоению конструкций и правил алгоритмического языка. Преподавателю надо знать, что привить воспитанникам навыки программирования можно только путем обучения учащихся самостоятельно исполнять их.

Подобно задачам по теме "Алгоритмизация", задачи в теме "Программирование" можно разбить на следующие типы:

- исполнение программы;
- найти ошибку в программе,
- определить, каков результат выполнения программы,
- усложнение задачи;
- построить математическую модель, составить алгоритм, написать программу, проверить ее.

Задачи с заданием "Найти ошибку в программе" или "Определить, каков результат выполнения программы" рекомендуется предлагать учащимся систематически в качестве общего задания для группы в начале урока (аналогично "устному счету" в младших классах). Выполнение задания фронтально позволит преподавателю:

- с первых минут активизировать внимание воспитанников на занятии,
- еще раз проговорить алгоритм выполнения программы для воспитанников, которые не поняли эту тему на первых занятиях,
- обратить внимание учащихся на типичные ошибки, выявленные преподавателем после проведения диагностической работы.

Задания на нахождение ошибок:

1) Найти ошибку в программе:

```
program 1;  
var a, b, c, S, p: real;  
begin  
write ('Введите длины сторон треугольника:');  
readln (a, b, c) ;  
p := (a + b + c)/2;  
S := sqrt (p*(p - a)*(p - b)*(p - c) ) ;  
End.
```

2) Найдите ошибки в записи условного оператора (типы данных считать допустимыми):

- a) if A=B and A=C then R:=2; else R:=5;
- б) if (A=B) or C Then R:=5 else R:=5;
- в) if A=B then begin R:=2; S:=5 end else R:=5; S:=2;

г) if A>10 then R:=2 else if A>15 then R:=6;

3) Найти ошибку в программе:

```
Var k, N, S: integer;  
Begin  
  Readln (N);  
  For k := 1 to N do S := S + k;  
  Writeln ('S = ', S);  
End.
```

Задания на определение результата выполнения программы можно, кроме того, предлагать воспитанникам для самостоятельного выполнения. Особенно они эффективны на первых занятиях знакомства с командами языка программирования.

Например,

1) x:=1.2; y:=x*2;

y:=y-5*x;

x:=abs(y);

write ('x=', x, 'y=', y);

2) Что будет выведено на экран дисплея после выполнения приведенных ниже операторов?

a:=-123.45; b:=0.0005;

writeln (a, b);

writeln ('a=', a, 'b=', b);

Задания на усложнение программы необходимо предусматривать на любом занятии. Для воспитанников была подобрана специальная система постепенно усложняющихся задач. Подобная организация работы воспитанников позволяет стимулировать и поддерживать их умственную активность.

С помощью специального отбора дидактического материала, происходило усвоение теоретических знаний, включивших в себя знания алгоритмического языка, основных алгоритмических конструкций. Проблемные задания способствовали формированию познавательного интереса.

Для развития алгоритмического мышления воспитанников было важно постоянно тренироваться в упражнениях, которые имели ошибки. Задания с ошибками стимулировали развитие алгоритмического мышления, так как требовали внимания к излагаемому материалу, активизировали деятельность воспитанников.

Задания для проверки знаний учащихся по разделу программирование.

Диагностическая работа №1.

Программирование линейных алгоритмов.

Вариант 1.

1. Вычислить и напечатать длину окружности и ее радиус, если площадь соответствующего круга равен $31,4 \text{ см}^2$.
2. Вычислить и напечатать результат выражения
$$x = \frac{a+b}{ab}, \quad a=10, b=5$$
3. Вычислить длину стороны A , площадь S и периметр P квадрата, описанного около окружности радиуса R .
4. Коническая куча зерна имеет высоту H м и длину окружности основания C м. Составить программу определения массы зерна, если масса 1 м^3 его равна M кг.

Вариант 2.

1. Вычислить и напечатать длину стороны и длину диагонали квадрата, если его периметр равен $10,4$ см.
2. Вычислить и напечатать результат выражения

$$x = \frac{5(a+2d)}{a+d} \quad \text{при } a=15, d=5$$

3. Вычислить длину стороны A , площадь S и периметр P квадрата, описанного около окружности радиуса R .
4. даны высота H и площади $Q1$ и $Q2$ верхнего и нижнего основания усеченной пирамиды. Написать программу определения объема усеченной пирамиды.

Диагностическая работа №2

Программирование разветвляющихся алгоритмов.

Вариант 1.

1. Написать программу, запрашивающую возраст пользователя. Если ему не менее 18 лет, сообщите, что он имеет право голосовать, в противном случае вычислите, через сколько лет ему будет предоставлено это право.
2. Написать программу, вычисляющую значение функции

$$y = \frac{ab}{a^2 - b^2}$$
3. Написать программу, определяющую существует ли треугольник, длины сторон которого равны числам A , B и C .
4. В ЭВМ поступают данные о количестве полных лет трех призеров спартакиады. Написать программу, выбирающую и печатающую возраст самого старшего призера.

Вариант 2.

1. Написать программу, уменьшающую четное число X в два раза.
2. Написать программу, вычисляющую значение функции

$$z = \frac{x+y}{x^2 + y^2}$$
3. Написать программу, определяющую по длинам сторон A , B и C , является ли треугольник прямоугольным.
4. В ЭВМ поступают результаты соревнований по плаванию для трех спортсменов. Написать программу, выбирающую и печатающую лучший результат.

Диагностическая работа №3.

Программирование циклических алгоритмов.

Вариант 1.

1. Составить программу вычисления суммы первых N членов натурального ряда.
2. Не пользуясь операцией умножения, составить программу умножения натурального числа A на натуральное число B .
3. В первый час работы рабочий изготавливает 25 деталей, за каждый последующий час на 3 детали больше, чем за предыдущий. Подсчитать, сколько рабочий изготовит деталей за 8 часов работы.

Вариант 2.

1. Найти сумму и произведение всех целых чисел от 1 до 20.

2. Составить программу возведения числа A в натуральную степень N .
3. В кинотеатре 30 рядов кресел. В первом ряду 20 кресел, в каждом последующем на 2 кресла больше, чем в предыдущем. Стоимость одного места в первом ряду – 5 рублей, а в каждом последующем – на 1 рубль больше, чем в предыдущем. Подсчитать выручку кинотеатра за один сеанс при целиком заполненном зале.

Диагностическая работа №4.

Программирование вспомогательных алгоритмов.

Вариант 1.

1. Написать программу вычисления значений функции при заданных значениях аргумента. Вычисление y оформить во вспомогательном алгоритме.

$$y = x^2 + x + 1, \quad x \in \{5; 9; 15\}$$

2. Разработать собственный шрифт. Написать программу, выводящую на экран слово «КАССА». Для вывода букв слова использовать вспомогательные алгоритмы для букв К, А и С.
3. Написать программу вычисления значения выражения

$$\frac{a! + b!}{c!}$$

Вариант 2.

1. Написать программу вычисления значений функции при заданных значениях аргумента. Вычисление y оформить во вспомогательном алгоритме.

$$y = \frac{1}{2x}, \quad x \in \{-1; 1,8; 2\}$$

2. Разработать собственный шрифт. Написать программу, выводящую на экран слово «БАНАН». Для вывода букв слова использовать вспомогательные алгоритмы для букв Б, А и Н.
3. Написать программу вычисления значения выражения

$$\frac{n!}{m!(n-m)!}$$

Диагностическая работа №5

Одномерные массивы

Вариант 1

1. С помощью датчика случайных чисел на интервале от 150 до 300 получить целочисленный массив, состоящий из 135 элементов.
2. Дан целочисленный массив заданный датчиком случайных чисел в интервале (100). Требуется умножить отрицательные значения элементов этого массива на 20.

Вариант 2

1. Составьте программу вычисления произведения тех значений элементов массива, которые меньше 15.

2. Дан одномерный массив. Из элементов массива, равных **100**, сформировать массив **b.**, а из элементов меньших **33**, сформировать массив **c.**

Двумерные массивы

Вариант 1


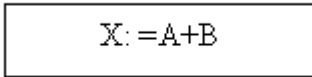
1. Из целочисленного массива $A(n,m)$ сформировать массив $B(n)$, в котором элемент $b[i]$ равен количеству положительных элементов i -й строки массива $A(n,m)$.
2. В целочисленном массиве $A(n,m)$ для каждого столбца найдите среднее арифметическое максимального и минимального элементов.
3. Удалите из массива $A(n,m)$ все столбцы, состоящие только из положительных элементов.
4. Вставьте в массив $A(n,m)$ после первой строки, первый элемент которой отрицателен, нулевую строку.

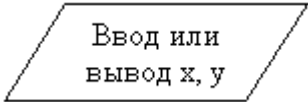
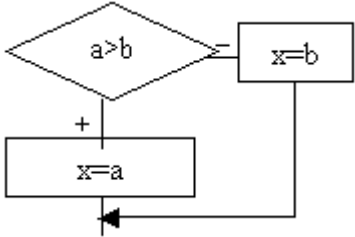
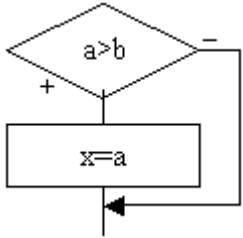
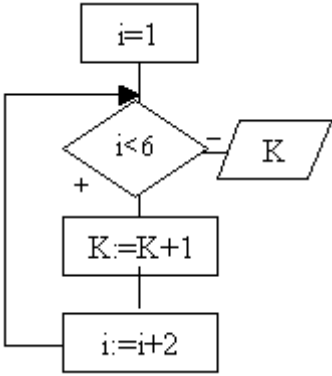
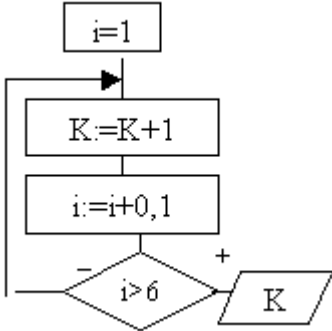
Вариант 2

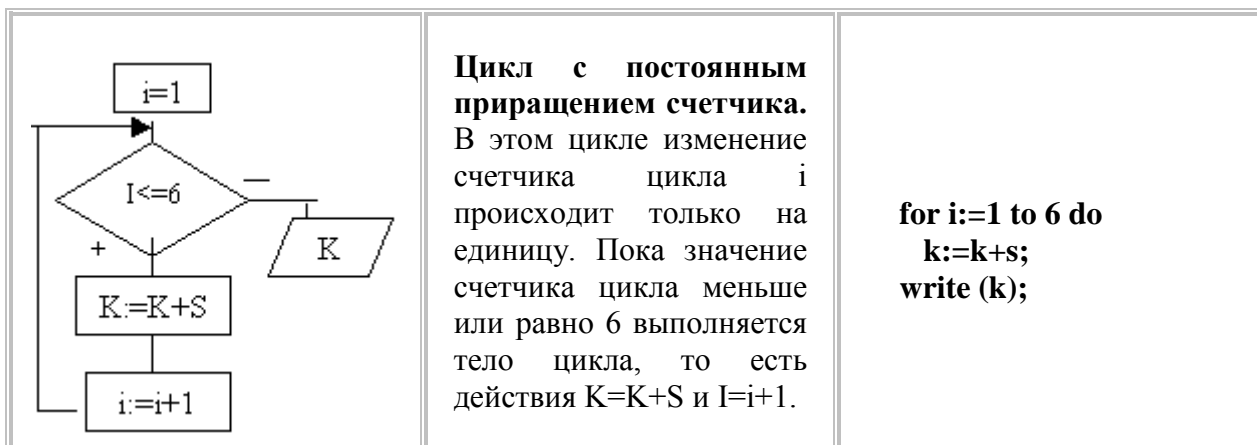
1. Из целочисленного массива $A(n,m)$ сформировать массив $B(n)$, в котором элемент $b[i]$ равен 1, если среди элементов i -ой строки есть минимальный элемент массива $A(n,m)$ или 0, если минимального элемента в строке нет.
2. В целочисленном массиве $A(n,m)$ для каждой строки найдите среднее арифметическое положительных чётных элементов.
3. Удалите из массива $A(n,m)$ первую строку с максимальным элементом (считается, что в массиве несколько максимальных элементов).
4. Вставьте в массив $A(n,m)$ после каждого столбца из положительных элементов столбец из 0.

Дидактический материал, используемый на занятиях

Таблица соответствия алгоритмических и программных фрагментов

| Фрагменты блок-схем алгоритмов | Назначение | Соответствующие фрагменты программ на языке Pascal |
|---|--|--|
|  | Начало и конец алгоритма. | Begin и End |
|  | Блок обработки (в нем вычисляются новые значения или производится вызов подпрограмм). | X := A + B |

| | | |
|---|--|---|
|  | <p>Ввод исходных данных или вывод результатов.</p> | <p>Read (x,y) или write (x,y)</p> |
|  | <p>Ветвление полное. Если значение переменной a больше b, то выполняется $x=a$, иначе $x=b$.</p> | <p>if a>b then x:=a else x:=b</p> |
|  | <p>Ветвление неполное. Если значение переменной a больше b, то выполняется $x=a$.</p> | <p>if a>b then x:=a</p> |
|  | <p>Цикл с предусловием. Пока значение условия $i<6$ истинно выполняется тело цикла, то есть действия $K=K+1$ и $i=i+2$. Переменная i определяет количество повторений и называется счетчиком цикла.</p> | <pre> i:=1; while i<6 do begin k:=k+1; i:=i+2; end write (k); </pre> |
|  | <p>Цикл с постусловием. Пока значение условия $i>6$ ложно выполняется тело цикла, то есть действия $K=K+1$ и $i=i+0,1$. Переменная i определяет количество повторений в цикле и называется счетчиком цикла.</p> | <pre> i:=1; repeat k:=k+1; i:=i+0.1; until i<6; write (k); </pre> |



На занятиях могут быть использованы материалы данных сайтов:

<http://comp-science.narod.ru/p.htm>

<http://forcoder.ru/pascal/>

<http://internika.org/umk-osnovi-algoritm-i-programm-na-Pascal>

<http://www.mirknig.com/knigi/programming/1181455889-vvedenie-v-yazyk-paskal.html>

<http://nsportal.ru/shkola/informatika-i-ikt/library/elektronnyi-uchebnik-po-teme-pascal>

<http://intepira.ru/?p=325>

Тест на знание учебного материала

Тест по теме «Алгоритмизация»

1. Алгоритм - это:

- а) правила выполнения определенных действий;
- б) ориентированный граф, указывающий порядок исполнения некоторого набора команд;

- в) понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;
- г) набор команд для компьютера;
- д) протокол вычислительной сети.

2. Укажите наиболее полный перечень способов записи алгоритмов:

- а) словесный, графический, программный;
- б) словесный;
- в) графический, программный;
- г) словесный, программный;

3. Суть такого свойства алгоритма как *результативность* заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

4. Алгоритм называется линейным:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

5. Какой из документов является алгоритмом?

- а) Правила техники безопасности.
- б) Инструкция по получению денег в банкомате.
- в) Расписание уроков.

6. В блок-схеме блок проверки условия обозначается:

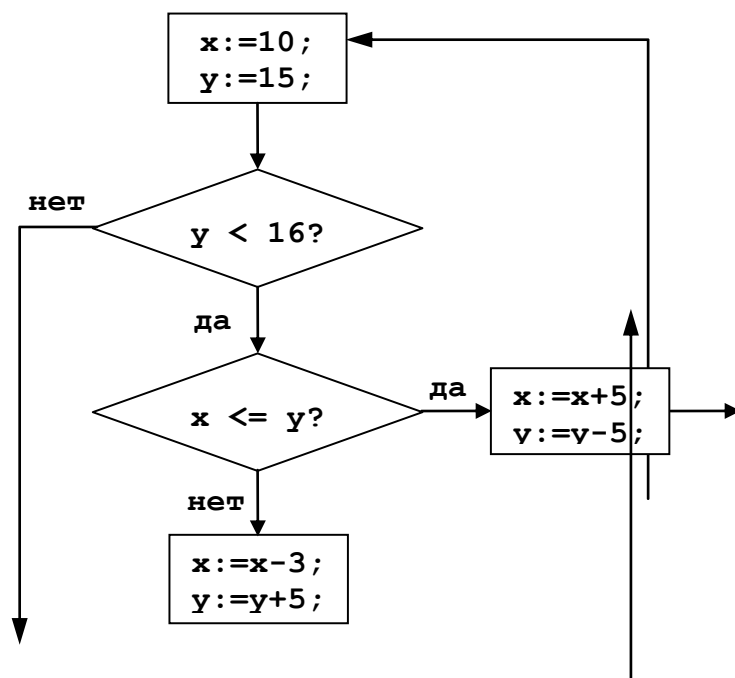
- а) ромбом;
- б) прямоугольником;
- в) овалом;
- г) шестиугольником.

7. На какой угол относительно своего начального положения повернется черепашка после выполнения команд:

НАЛЕВО 45 НАПРАВО 90 НАЛЕВО 30 НАПРАВО 15

а) 130; б) 30; в) 90; г) 50.

8. Определите значения переменных x и y после выполнения фрагмента алгоритма.



9. Одним из свойств алгоритма является *дискретность*. Выберите подходящее пояснение:

- а) это последовательное выполнение простых или ранее определенных шагов
- б) каждая команда алгоритма должна быть понятна тому, кто его исполняет
- в) состоит в совпадении получаемых результатов независимо от пользователя и применяемых технических средств
- г) означает возможность получения результата после выполнения конечного количества операций
- д) решение должно быть правильным для любых допустимых исходных данных

10. Какую фигуру начертит черепашка после выполнения команд:

ВПЕРЕД 40 НАЛЕВО 60 НАЗАД 40 НАЛЕВО 60 ВПЕРЕД 40

11. Черепашка выполнила следующий алгоритм:

ПОВТОРИ 4 [НАПРАВО 60 ВПЕРЕД 20 НАПРАВО 60]

Какая фигура появится на экране?

Тест по теме «Программирование»

1. Служебное слово *VAR* в программе на языке *Pascal* фиксирует начало раздела программы, содержащего:

- а) операторы;
- б) список меток;
- в) описание сложных типов данных;
- г) перечень констант;
- д) описание переменных.

2. В алфавит языка *Pascal* не входит служебное слово:

- а) THEN;
- б) BEGIN;
- в) END;
- г) STEP;
- д) IF;

3. Числа в языке *Pascal* различаются:

- а) как натуральные и целые;
- б) как целые и вещественные;
- в) как натуральные и вещественные;
- г) как целые и иррациональные;
- д) как целые и рациональные.

4. При истинности какого условия последовательность переменных *A, B, C* не является упорядоченной по возрастанию:

- а) $(A < B) \text{ AND } (\text{NOT}(B \geq C))$;
- б) $(A \leq B) \text{ AND } (B \leq C)$;
- в) $\text{NOT} ((A > B) \text{ OR } (B > C))$;
- г) $(A \leq B) \text{ AND } (\text{NOT}(B > C))$;
- д) $\text{NOT} ((A \leq B) \text{ AND } (B \leq C))$.

5. Сколько раз будут выполнены операторы тела цикла при выполнении следующего фрагмента программы:

A:=1; N:=0; S:=0;

*While A>1/1050 Do Begin A:=Exp(-N*Ln(2)); S:=S+A End;*

- а) 10; б) 1050; в) 11; г) 100; д) 1.

6. При наборе программы

Program T28;

Var a: array[1..8] of Integer; s, k: Integer;

Begin

For k:=1 to 8 Do Readln(a[k]);

s:=0;

```
For k:=1 to 8 Do If a[k]<0 Then s:=s+a[k];  
Writeln(s)  
End.
```

вычисления суммы отрицательных элементов массива вместо оператора $s:=s+a[k]$ ошибочно был записан оператор $s:=s+1$. Каким оказался ответ после исполнения неверной программы, если в качестве элементов массива были введены числа: -1, 3, -2, 4, -5, 6, -7, 8:

а) 8; б) -15; в) -3; г) 4; д) 6.

7. Формальные параметры процедуры:

- а) описываются в ее заголовке;
- б) перечисляются при вызове процедуры;
- в) указываются при описании данных в программе;
- г) декларируются при описании промежуточных переменных процедуры;
- д) присваиваются в процессе выполнения программы.

8. В какую из перечисленных ниже структур можно объединять данные различного типа:

- а) строка;
- б) массив;
- в) множество;
- г) файл;
- д) запись.