

Информация для размещения на официальном сайте ГБПОУ
«Светлоградский региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	108
Дата	26.02.2025
Время	9-50-11-10
Наименование УД/МДК/УП/П П	Информатика
Ф.И.О. преподавателя	Сахарчук Т.В., Сахарчук Н.О.
Электронная почта	saharchyk777@mail.ru , SNO54@yandex.ru
Основная литература	<ol style="list-style-type: none">1. Михеева Е.В. Информатика (2-е изд., стер.) учебник –Москва:ОИЦ «Академия» 2022г.2. Угринович Н.Д. Информатика (для СПО)Учебник –Москва:КноРус 2023г. https://www.book.ru/book/9241893. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б Основы информатики (СПО)Учебник–Москва: КноРус 2023г. https://www.book.ru/book/927691 Дополнительные источники: <ol style="list-style-type: none">1. Михеева Е.В. Информатика. Практикум (2-е изд., стер.) учеб. пособие –Москва:ОИЦ «Академия» 2022г.2. Иопа Н.И Информатика (конспект лекций) –Москва: КноРус 2023г. https://www.book.ru/book/917889
Тема № 37-38	Практические занятия на тему: Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма
Задание	<p>1. Теоретические сведения к практической работе</p> <p>1.1. Основные алгоритмические конструкции.</p> <p>Алгоритм является фундаментальным понятием информатики. Представление о нем необходимо для эффективного применения вычислительной техники к решению практических задач.</p> <p><i>Алгоритм</i> - это последовательность действий, которая должна быть выполнена для достижения желаемого результата.</p> <p><i>Алгоритм решения</i> некоторой задачи - это алгоритм, приводящий к решению этой задачи за конечное число действий.</p> <p>1.2. Свойства алгоритма и его исполнители.</p> <ol style="list-style-type: none">1. Дискретность. Разделение алгоритма на последовательность законченных действий - шагов. Каждое действие должно быть закончено прежде, чем исполнитель приступит к выполнению следующего шага.2. Результативность. Получение из исходных данных результата за конечное число шагов.3. Массовость. Возможность применения алгоритма к большому количеству различных исходных данных.4. Детерминированность. Выполнение команд алгоритма в строго определенной последовательности.5. Выполнимость и понятность. Алгоритм не должен содержать предписаний,

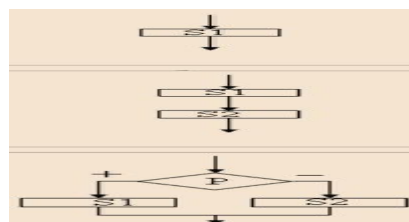
смысл которых может восприниматься неоднозначно.

6. Точность. Запись алгоритма должна быть такой, чтобы на каждом шаге его выполнения было известно, какую команду нужно выполнять следующей.

7. Конечность. Завершение работы алгоритма за конечное число шагов.

Наиболее понятно структуру алгоритма можно представить с помощью блок-схемы, в которой используются геометрические фигуры (блоки), соединенные между собой стрелками, указывающими направление потоков информации (последовательность выполнения действий). Приняты определенные стандарты графических изображений блоков. Например, команду обработки информации помещают в блок, имеющий вид прямоугольника, проверку условий - в ромб, команды ввода или вывода - в параллелограмм, а овалом обозначают начало и конец алгоритма.

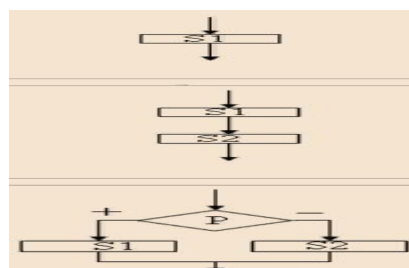
Структурной элементарной единицей алгоритма является простая команда, обозначающая один элементарный шаг переработки или отображения информации. Простая команда на языке схем изображается в виде функционального блока.



Данный блок имеет *один вход* и *один выход*. Из простых команд и проверки условий образуются составные команды, имеющие более сложную структуру и тоже *один вход* и *один выход*.

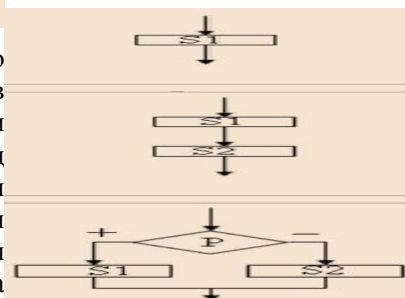
Структурный подход к разработке алгоритмов определяет использование только базовых алгоритмических структур (конструкций): следование, ветвление, повторение, которые должны быть оформлены стандартным образом.

1.3. Рассмотрим основные структуры алгоритма.

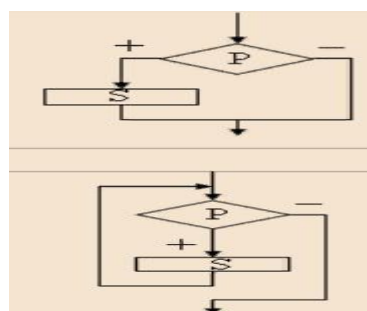


Алгоритм следования состоит только из простых команд. На рисунке простые команды имеют условное обозначение *S1* и *S2*. Из команд следования образуются линейные алгоритмы. Примером линейного алгоритма будет нахождение суммы двух чисел, введенных с клавиатуры.

Алгоритм ветвления - это команда алгоритма, в которой в зависимости от условия *P* выполняется или другое *S2* действие. Из команд ветвления составляются разветвляющиеся алгоритмы (алгоритмы ветвления). Примером разветвляющегося алгоритма будет нахождение большего из двух введенных с клавиатуры.

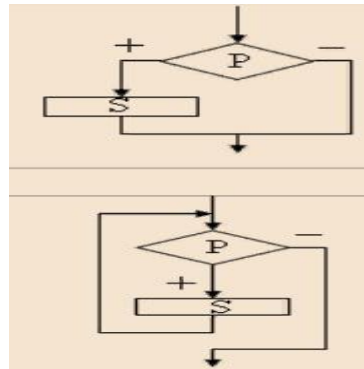


составная зависимости одно *S1*, или следования и (алгоритмы) будет чисел,



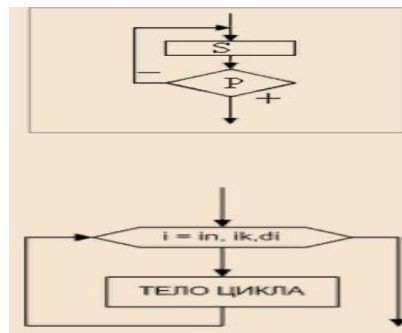
Алгоритм ветвления может быть полной и неполной формы. Неполная форма используется тогда, когда необходимо выполнять действие *S* только в случае соблюдения условия *P*. Если условие *P* не соблюдается, то команда ветвления завершает свою работу без выполнения действия. Примером неполной формы ветвления

будет уменьшение в два раза только четного числа.



Алгоритм повторения - это составная команда алгоритма, в которой в зависимости от условия P возможно многократное выполнение действия S (тело цикла). Из команд следования и команд повторения составляются циклические алгоритмы (алгоритмы повторения). На рисунке представлен циклический алгоритм с предусловием. Называется он так потому, что вначале проверяется условие, а уже затем выполняется действие (тело цикла). Причем действие выполняется, пока условие верно. Пример циклического алгоритма может быть следующий.

Пока с клавиатуры вводятся положительные числа, алгоритм выполняет нахождение их сумм.



В алгоритме повторения с постусловием сначала выполняется действие S (тело цикла) и лишь затем, проверяется условие P . Причем действие повторяется до тех пор, пока условие ложно. Примером будет уменьшение положительного числа до тех пор, пока оно неотрицательное. Как только число становится отрицательным, алгоритм повторения заканчивает

свою

работу.

Безусловный циклический алгоритм, его удобно использовать, если известно, сколько раз необходимо выполнить тело цикла. Выполнение безусловного циклического алгоритма начинается с присвоения переменной i стартового

значения in . Затем следует проверка, не превосходит ли переменная i конечное значение ik . Если превосходит, то цикл считается завершенным, и управление передается следующему за телом цикла оператору. В противном случае выполняется тело цикла, и переменная i меняет свое значение в соответствии с указанным шагом di . Далее снова производится проверка значения переменной i и алгоритм повторяется. Переменную i называют параметром цикла, так как это переменная, которая изменяется внутри цикла по определенному закону и влияет на его окончание.

С помощью соединения только этих элементарных конструкций (последовательно или вложением) можно "собрать" алгоритм любой степени сложности.

2. Порядок выполнения заданий

2.1. Способы описания алгоритма

Задача: Найти сумму двух чисел 2 и 3.

1. Словесный способ

Алгоритм представляет собой описание на естественном языке

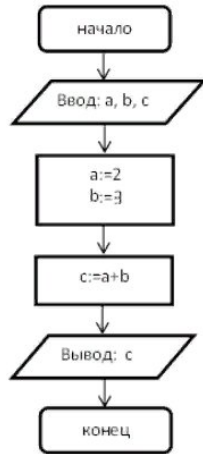
последовательных этапов обработки данных.

К двум прибавляем три, получаем пять.

2. Графический способ

Изображение алгоритма в виде последовательности связанных между собой функциональных блоков.

Блок-схема позволяет сделать алгоритм более наглядным и выделяет в алгоритме основные алгоритмические структуры (линейная, ветвление, выбор и цикл).



Элемент блок-схемы	Назначение элемента блок-схемы
	Прямоугольник с закругленными углами, применяется для обозначения начала или конца алгоритма
	Параллелограмм, предназначен для описания ввода или вывода данных, имеет один вход сверху и один выход внизу
	Прямоугольник, применяется для описания линейной последовательности команд, имеет один вход сверху и один выход внизу
	Ромб, служит для обозначения условий в алгоритмических структурах «ветвление» и «выбор», имеет один вход сверху и два выхода (налево, если условие выполняется, и направо, если условие не выполняется)

3. Псевдокод

Система обозначений и правил, предназначенная для единообразной записи алгоритмов.

Алг Сумма

дано а, b, с;

надо $c=a+b$;

нач $a:=2, b:=3$;

$c:= a+b$;

кон.

4. Программный способ (алгоритмический)

Алгоритм, предназначенный для записи на компьютере, должен быть записан на понятном ему языке. Такой язык называется языком программирования, а запись алгоритма на этом языке – программа.

Языки программирования предназначены для создания **программ**, которые могут быть исполнены ЭВМ или другими автоматическими устройствами, например, станками с числовым программным управлением. Система Pascal ABC предназначена для обучения программированию на языке Паскаль. Как и любой алгоритм, являющийся последовательностью инструкций, программа на языке Паскаль состоит из команд (операторов), записанных в определенном порядке и формате. Команды позволяют получать, сохранять и обрабатывать данные различных типов (например, целые числа, символы, строки символов, т.д.).

Кроме команд в записи программы участвуют еще так называемые "служебные слова", организующие структуру программы.

Правила языка Паскаль предусматривают единую для всех программ форму основной структуры

Program ;

;

Begin

End.

Здесь слова **Program**, **Begin** и **End** являются служебными. Правильное и уместное употребление этих слов является обязательным.

Пример. Ввести в компьютер два целых числа, найти их сумму, результат вывести на экран с поясняющим текстом.

Внимание! Две косые черты (//) отделяют комментарии, их набирать не нужно.

```
program gaschet;// название программы
uses crt;// подключаемые модули
var x, y, s:integer;// объявление имен переменных и их типа
begin// начало исполнительской части
writeln('Введите два целых числа');//написать на экране текст
readln(x,y);//прочитать данные с клавиатуры и запомнить их в переменных
s:=x+y; // выполнить расчет и запомнить его в переменной
writeln('Сумма чисел =',s); //написать на экране текст и значение переменной
end. //конец программы
```

Линейный алгоритм

Пример записи алгоритма в виде блок-схемы, псевдокоде и на языке Паскаль.

Блок-схема	Псевдокоды	Паскаль
<pre>graph TD Start([начало]) --> Input[/ввод a, b/] Input --> Process[g := √(a * b)] Process --> Output[/вывод g/] Output --> End([конец])</pre>	<p>алг среднее геометрическое</p> <p>вещ a, b, g</p> <p>нач</p> <p>ввод a, b</p> <p>$g := (a * b)^{(1/2)}$</p> <p>вывод g</p> <p>кон</p>	<pre>program Srednee_geometr; var a, b, g: real; begin readln (a, b); s := sqrt(a * b); writeln (g) end.</pre>

3. Задания для самостоятельной работы

ХОД РАБОТЫ:

Задание № 1. Некий злоумышленник выдал следующий алгоритм за алгоритм получения кипятка:

1. Налить в чайник воду.
2. Открыть кран газовой горелки.
3. Поставить чайник на плиту.
4. Ждать, пока не закипит вода.
5. Поднести спичку к горелке.
6. Зажечь спичку.
7. Выключить газ.

	<p>Исправьте алгоритм, чтобы предотвратить несчастный случай.</p> <p>Задание № 2. Имеются два кувшина емкостью 3 л и 8 л. Напишите алгоритм на естественном языке, выполняя который можно набрать из реки 7 л воды. (Разрешается пользоваться только этими кувшинами.)</p> <p>Задание № 3. Построить блок-схему линейного алгоритма вычисления значения выражения по вариантам и написать программу на псевдокоде.</p> <table border="1" data-bbox="486 414 1476 772" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;"> Вариант 1 $z = ctg\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$ </td> <td style="width: 33%;"> Вариант 5 $z = tg3\alpha$ </td> <td style="width: 33%;"> Вариант 9 $z = 2\sin\alpha$ </td> </tr> <tr> <td> Вариант 2 $z = \frac{\cos\alpha + \sin\alpha}{\cos\alpha - \sin\alpha}$ </td> <td> Вариант 6 $z = \frac{1}{\sqrt{b+2}}$ </td> <td> Вариант 10 $z = \frac{\sqrt{m} - \sqrt{n}}{m}$ </td> </tr> <tr> <td> Вариант 3 $z = \cos^2\alpha + \cos^4\alpha$ </td> <td> Вариант 7 $z = \frac{4-a^2}{2}$ </td> <td> Вариант 11 $z = -\sqrt{m}$ </td> </tr> <tr> <td> Вариант 4 $z = \frac{1-tg\alpha}{1+tg\alpha}$ </td> <td> Вариант 8 $z = \frac{1}{\sqrt{a+\sqrt{2}}}$ </td> <td> Вариант 12 $z = \sqrt{\frac{x+3}{x-3}}$ </td> </tr> </table> <p>Задание № 4. Перед выходным днем папа сказал своему сыну: «Давай спланируем свой завтрашний день. Если будет хорошая погода, то проведем день в лесу. Если же погода будет плохая, то сначала займемся уборкой квартиры, а во второй половине дня сходим в зоопарк». Что получится на выходе блок-схемы, если: а) <i>погода хорошая</i>; б) <i>погода плохая</i>?</p>	Вариант 1 $z = ctg\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$	Вариант 5 $z = tg3\alpha$	Вариант 9 $z = 2\sin\alpha$	Вариант 2 $z = \frac{\cos\alpha + \sin\alpha}{\cos\alpha - \sin\alpha}$	Вариант 6 $z = \frac{1}{\sqrt{b+2}}$	Вариант 10 $z = \frac{\sqrt{m} - \sqrt{n}}{m}$	Вариант 3 $z = \cos^2\alpha + \cos^4\alpha$	Вариант 7 $z = \frac{4-a^2}{2}$	Вариант 11 $z = -\sqrt{m}$	Вариант 4 $z = \frac{1-tg\alpha}{1+tg\alpha}$	Вариант 8 $z = \frac{1}{\sqrt{a+\sqrt{2}}}$	Вариант 12 $z = \sqrt{\frac{x+3}{x-3}}$
Вариант 1 $z = ctg\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$	Вариант 5 $z = tg3\alpha$	Вариант 9 $z = 2\sin\alpha$											
Вариант 2 $z = \frac{\cos\alpha + \sin\alpha}{\cos\alpha - \sin\alpha}$	Вариант 6 $z = \frac{1}{\sqrt{b+2}}$	Вариант 10 $z = \frac{\sqrt{m} - \sqrt{n}}{m}$											
Вариант 3 $z = \cos^2\alpha + \cos^4\alpha$	Вариант 7 $z = \frac{4-a^2}{2}$	Вариант 11 $z = -\sqrt{m}$											
Вариант 4 $z = \frac{1-tg\alpha}{1+tg\alpha}$	Вариант 8 $z = \frac{1}{\sqrt{a+\sqrt{2}}}$	Вариант 12 $z = \sqrt{\frac{x+3}{x-3}}$											
Контрольный тест	<p>ДОМАШНЕЕ ЗАДАНИЕ</p> <ol style="list-style-type: none"> Изучить материал занятия. Выполнить Задания для самостоятельной работы (согласно варианта: 1 вариант – 1 номер по списку в журнале; 2 вариант – 2; .. и т.д.). Сделать вывод о проделанной практической работе. 												

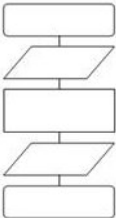
Дата 26.02.2025

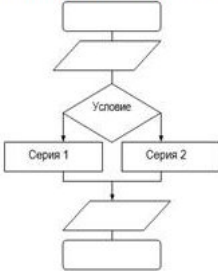
Подпись

Ф.И.О. преподавателя

Информация для размещения на официальном сайте ГБПОУ
«Светлоградский региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	108
Дата	27.02.2025
Время	13.20-14-40
Наименование УД/МДК/УП/П П	Информатика
Ф.И.О. преподавателя	Сахарчук Т.В., Сахарчук Н.О.
Электронная почта	saharchyk777@mail.ru , SNO54@yandex.ru
Основная литература	<ol style="list-style-type: none">1. Михеева Е.В. Информатика (2-е изд., стер.) учебник –Москва:ОИЦ «Академия» 2022г.2. Угринович Н.Д. Информатика (для СПО)Учебник –Москва:КноРус 2023г. https://www.book.ru/book/9241893. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б Основы информатики (СПО)Учебник–Москва: КноРус 2023г. https://www.book.ru/book/927691 Дополнительные источники: <ol style="list-style-type: none">1. Михеева Е.В. Информатика. Практикум (2-е изд., стер.) учеб. пособие –Москва:ОИЦ «Академия» 2022г.2. Иopa Н.И Информатика (конспект лекций) –Москва: КноРус 2023г. https://www.book.ru/book/917889
Тема № 39-40	Практические занятия на тему: Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма
Задание	<p>Типы алгоритмов <u>Алгоритмы бывают:</u></p> <ul style="list-style-type: none">• <i>линейные</i>• <i>разветвляющиеся</i>• <i>циклические</i> <p>Алгоритм, в котором команды выполняются последовательно одна за другой, называется линейным алгоритмом.</p> 

	<p>В разветвляющихся алгоритмах входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (серий).</p> <p>В алгоритмической структуре «ветвление» та или иная серия команд выполняется в зависимости от истинности условия.</p> <p>Условие может быть либо истинным, либо ложным.</p>  <p>Практическое задание</p> <p>Задание 1. составить в виде блок-схемы алгоритм нахождения середины отрезка при помощи циркуля и линейки опираясь на пример алгоритма естественного языка Дан отрезок АВ. Пример: «Алгоритм деления отрезка АВ пополам».</p> <ol style="list-style-type: none"> 1. поставить ножку циркуля в т.А; 2. установить раствор циркуля равным длине отрезка АВ; 3. провести окружность; 4. поставить ножку циркуля в т.В; 5. провести окружность; 6. через точки пересечения окружностей провести прямую; 7. отметить точку пересечения этой прямой с отрезком АВ. <p>Практическое задание</p> <p>Проектное задание Постройте структурную схему алгоритма поиска среднего роста учащихся в колледже, а также минимального и максимального значений роста. Используйте массив для описания списка учащихся, циклическую алгоритмическую конструкцию для поиска минимума и максимума, суммирования всех элементов этого массива. Результат представьте, как итог вычисления среднего арифметического, а для минимального и максимального значений роста учащихся сообщите соответствующие номера этих учащихся в списке группы.</p>
Контрольный тест	<p>Сделать отчет по практическому заданию.</p> <p>Содержание отчета</p> <p>Отчет должен содержать:</p> <ol style="list-style-type: none"> 1. Название работы. 2. Цель работы. 3. Задание и его решение. 4. Вывод по работе. <p>Задание 2. Используйте ресурсы Интернета для нахождения определения свойств алгоритма и запишите их в тетрадь. Приведите примеры.</p> <p>Задание 3. Допишите в тетради в основные алгоритмические конструкции недостающие правила блок-схем.</p> <p>Задание 4. Сделать вывод о проделанной практической работе</p>

Дата 27.02.2025

Подпись

Ф.И.О. преподавателя

Информация для размещения на официальном сайте ГБПОУ
«Светлоградский региональный сельскохозяйственный колледж»

Для электронного обучения

Группа	108
Дата	27.02.2025
Время	14-50-16-10
Наименование УД/МДК/ УП/ПП	Информатика
Ф.И.О. преподавателя	Сахарчук Т.В., Сахарчук Н.О.
Электронная почта	saharchyk777@mail.ru , SNO54@yandex.ru
Основная литература	<p>1. Михеева Е.В. Информатика (2-е изд., стер.) учебник –Москва:ОИЦ «Академия» 2022г.</p> <p>2. Угринович Н.Д. Информатика (для СПО)Учебник –Москва:КноРус 2023г. https://www.book.ru/book/924189</p> <p>3. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б Основы информатики (СПО)Учебник–Москва: КноРус 2023г. https://www.book.ru/book/927691</p> <p>Дополнительные источники:</p> <p>1. Михеева Е.В. Информатика. Практикум (2-е изд., стер.) учеб. пособие –Москва:ОИЦ «Академия» 2022г.</p> <p>2. Иopa Н.И Информатика (конспект лекций) –Москва: КноРус 2023г. https://www.book.ru/book/917889</p>
Тема № 41-42	Практические занятия на тему: Структурированные типы данных. Массивы. Вспомогательные алгоритмы. Задачи поиска элемента с заданными свойствами. Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов
Задание	<p>Цель работы: овладеть основными приемами работы с одномерными и двумерными массивами, уметь различать в двумерном массиве обработку строк и столбцов, а также отличать нахождение первых и последних элементов последовательности, обладающих некоторым свойством.</p> <p>Общие сведения</p> <p><i>Скалярный тип</i> – простой тип данных. Скалярное данное неделимо. <i>Массив</i> – это структурированный тип данных. Массив состоит из нескольких элементов. Ко всему массиву можно обращаться по его имени. Можно обращаться к его элементу, но для этого надо задать индекс (индексы). Массивы бывают одномерные и многомерные. Для объявления массива необходимо задать типы его индексов и компонент:</p> <p>ARRAY [Тип индексов] OF <Тип компонент>;</p> <p><i>Тип компонент массива</i> – это просто тип данных, ассоциированный с каждой компонентой массива. Тип компонент может быть любым REAL, INTEGER, CHAR, BOOLEAN, перечислимым, интервальным. В качестве компоненты массива может быть взят и тип массив.</p> <p>Тип индекса должен быть одним из упорядоченных типов, т.е. любым скалярным типом, кроме REAL: INTEGER, CHAR, интервальный, перечислимый. Тип индекса определяет границы изменения индекса. Если сделана попытка использовать несуществующую компоненту, то возникает ошибка (ошибка неверного индекса).</p> <p>Одномерные массивы</p> <p>Одномерный массив можно задать (объявить) двумя способами:</p> <p>1. С помощью служебного слова TYPE описывается тип массива, а затем с помощью VAR вводится переменная этого типа.</p>

Общая форма записи

TYPE <тип массива> = ARRAY [тип индекса] OF <тип компонент>;

VAR <переменная>: <тип массива>;

2. С помощью слова VAR сразу описывается переменная типа массив.

Общая форма записи

VAR <переменная>: ARRAY [тип индекса] OF <тип компонент>;

Например, объявление массива из 100 элементов типа REAL можно осуществить двумя способами:

1. type R100 = array [1..100] of real;

var A: R100;

2. var A: array [1..100] of real.

Здесь задан массив с именем A, и его элементы имеют имена: A[1],...,A[100]. Чаще всего для типа индекса используют интервальный тип на основе типов INTEGER и CHAR. Однако можно в качестве индексов брать перечислимый тип.

Пример 1. Подсчет числа вхождений букв в текст определенной длины.

```
program COUNTER;
var COUNT: array ['a'..'z'] of integer;
CH: char; N: integer;
begin
for CH := 'a' to 'z' do
COUNT [CH] := 0; N := 0;
repeat
read (CH); N := N + 1;
if (CH >= 'a') and (CH <= 'z') then
COUNT [CH] := COUNT [CH] + 1;
until CH = '!';
for CH := 'a' to 'z' do
writeln (CH, COUNT [CH]:5);
end.
```

Пояснение. В этом примере тип индекса есть интервальный тип на базе типа CHAR, а тип компонент есть целое число. Таким образом, элементы массива – числа, а их индексы – буквы, т.е. число элементов массива равно 26 (по числу букв латинского алфавита).

Рассмотрим теперь случай, когда тип индекса задан перечислимым типом, а компоненты массива представлены компонентами интервального типа на базе типа INTEGER.

Пример 2. Присваивание переменной с именем месяца числа дней этого месяца.

```
program NUMBRDAY;
type MONAT = (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG,
SEP, OKT, NOV, DEC);
var DAY : array [MONAT] of 28..31; T : MONAT;
begin
for T := JAN to DEC do
case T of
JAN, MAR, MAY, JUL, AUG, OKT, DEC: DAY [T] := 31;
APR, JUN, SEP, NOV: DAY [T] := 30;
FEB : DAY [T] := 28;
end;
end.
```

Многомерные массивы

Для определения позиции элемента в двумерном массиве необходимы два индекса. Любой двумерный массив есть матрица, а матрица есть таблица. Поэтому удобно описывать двумерные массивы путем указания границ изменения индексов (номеров) строк и столбцов.

Например, таблица символов $M \times N$, где M – число строк и N – число столбцов, может быть описана:

```
var TAB : array [1..M, 1..N] of char;
```

Общая форма записи

```
VAR <имя> : ARRAY [тип индекса строки, тип индекса столбца]  
OF <тип компонент>;
```

Однако двумерный массив можно интерпретировать как вектор-столбец, каждый элемент которого, в свою очередь, является одномерным массивом (вектор-строка). Этот подход к определению двумерного массива влечет его описание с помощью двух строк, где первая содержит описание строки, а вторая – описание столбца:

```
type LINE = array [1..N] of char;  
STOLB = array [1..M] of LINE;  
var TAB : STOLB.
```

Здесь TAB [I] – переменная типа LINE, а TAB [I][J] – переменная типа CHAR.

Общая форма записи

```
TYPE <тип строки> = ARRAY [тип индекса] OF <тип компонент>;  
<тип столбца> = ARRAY [тип индекса] OF <тип строки>;  
VAR <переменная массива> : <тип столбца>;
```

Эти два вида определения массивов задают и два способа обращения к элементам массива: TAB [I, J] – в первом случае и TAB [I][J] – во втором. Вполне очевидно, что сказанное выше для двумерного массива распространяется и на массивы большей размерности. Например, описание VAR CUBE : ARRAY [1..M, 1..N, 1..K] OF INTEGER определяет задание трехмерного массива целых чисел.

Примеры работы с массивами

Обработка массивов включает в себя, как правило, следующие компоненты: ввод массива (с клавиатуры или с помощью датчика случайных чисел), вывод полученного массива на экран и собственно его обработка. Все эти компоненты рекомендуется оформлять в виде отдельных процедур. При этом надо учитывать следующий фактор: если процедуре (или функции) будет передаваться массив, то надо объявить в ней этот массив как параметр с атрибутом VAR даже в том случае, если значение массива внутри процедуры не изменяется. Это нужно для того, чтобы не тратить времени и памяти на размещение внутри процедуры копии массива. Заметим, что параметр обязательно должен относиться к типу, имеющему имя.

Пр и м е р. Сумма элементов таблицы над верхней диагональю.

```
program SUMMA;  
const M = 10; {число строк таблицы}  
N = 10; {число столбцов таблицы}  
type LINE = array [1..n] of integer;  
TAB = array [1..m] of LINE;  
var s, i, j: integer; MAS: TAB;  
procedure VVODMASSIV (var MAS: TAB);  
begin  
  for i := 1 to M do  
    for j := 1 to N do  
      readln (MAS [i][j]);  
end;  
procedure VIVODMASSIV (var MAS: TAB);  
begin  
  for i := 1 to M do  
    begin  
      for j := 1 to N do
```

```

write (MAS [i][j]:4,' '); writeln;
end;
end;
        procedure OBRABOTKA (MAS: TAB; var SUM: integer);
begin
SUM := 0;
for i := 1 to M do
for j := 1 to N do
if j > i then SUM := SUM + MAS [i][j];
end;
begin
VVOODMASSIV (MAS);
writeln ('исходный массив'); VIVODMASSIV (MAS);
OBRABOTKA (MAS, s); writeln;
writeln ('сумма элементов = ',s);
end.

```

Табличное представление информации одно из самых распространенных, поэтому массивы широко применяются в прикладных программах.

Перед выполнением работы необходимо ознакомиться с теоретическим материалом по теме «Обработка массивов».

Пример 1. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить сумму положительных элементов.

Решение. При написании процедур ввода и вывода следует обратить внимание, что элементы – вещественные числа, поэтому необходимо позаботиться о верной обработке дробной части. Вычисление суммы оформим в виде функции с одним аргументом - массивом. Локальными переменными функции будут индексная переменная i и дополнительная переменная s для хранения текущей суммы элементов. В начале тела функции обязательно обнуление s . Каждый элемент массива сравним с нулем, и, если значение положительно, добавим его к искомой сумме s . В конце функции запишем значение переменной s в результирующую переменную.

```

program massiv_1;
const N=10;
type mas=array [1..N] of Real;
procedure Vvodmas(var A:mas);
var i:Integer;
begin
for i:=1 to N do A[i]:=-50+Random(101)+random;
end;
procedure Vivodmas(A:mas);
var i:Integer;
begin
for i:=1 to N do Write(A[i]:8:2);
Writeln
end;
function Summa(A:mas):real;
var i: Integer; s:real;
begin
s:=0;
for i:=1 to N do if A[i]>0 then s:=s+A[i];
Summa:=s;
end;
var A: mas;
begin
Randomize; Vvodmas(A);
Writeln('Исходный массив:'); Vivodmas(A);
Writeln('Ответ: ', Summa(A):0:2);
Readln
end.

```

Пример 2. В двумерном массиве, состоящем из целочисленных элементов, в каждом столбце поменять местами наибольший по модулю и последний не принадлежащий интервалу (a, b) элементы массива.

Решение. Преобразования необходимо провести в каждом столбце массива, поэтому параметр внешнего цикла в процедуре обработки - номер столбца j , а вложенного – номер

строки i .

Для перестановки двух элементов в столбце массива необходимо найти номера их строк $n1$ и $n2$, а затем поменять местами значения элементов с использованием промежуточной переменной p .

Чтобы найти наибольший по модулю элемент столбца, введем дополнительную переменную max , которая будет хранить максимальное по модулю значение в текущем столбце массива на данный момент. (Можно решить задачу без использования переменной max . Подумайте, как это сделать.)

Программа должна корректно работать с любыми входными данными, а значит и в тех случаях, когда некоторые или даже все столбцы массива содержат только элементы из интервала (a, b) , и обмен значений в некоторых столбцах или во всем массиве не нужен.

```
const n=10; m=7;
type mas=array [1..n,1..m] of Integer;
procedure Vvodmas(var D:mas);
  var i,j:Integer;
begin
  for i:=1 to n do
    for j:=1 to m do
      D[i,j]:=-50+Random(101);
    end;
  procedure Vivodmas(D:mas);
    var i,j:Integer;
  begin
    for i:=1 to n do
      begin
        for j:=1 to m do Write(D[i,j]:4);
          Writeln;
        end;
      end;
  procedure Obmen(a,b: real; var D:mas);
    var i,j,p,n1,n2,max: Integer;
  begin
    for j:=1 to m do
      begin
        n1:=1; max:=abs(D[1,j]);{считаем первый элемент столбца наибольшим по
модулю}
        for i:=2 to n do
          if abs(D[i,j])>max then {обнаружен больший элемент}
            begin n1:=i; max:=abs(D[i,j]) end;
          i:=n; {перебираем элементы столбца, начиная с последнего}
          while (i>=1)and (D[i,j]>a)and(D[i,j]
n2:=i;
          if n2<>0 then {если элемент, не принадлежащий интервалу (a,b), был найден}
            begin
              p:=D[n1,j]; D[n1,j]:=D[n2,j]; D[n2,j]:=p; {обмен значений}
            end;
          end;
        end;
      end;
  var D: mas; a,b:Real;
begin
  Randomize; Vvodmas(D);
  Writeln('Исходный массив:'); Vivodmas(D);
  Write('Введите через пробел концы интервала (a,b): '); Readln(a,b);
  Obmen(a,b,D);
  Writeln('Ответ:'); Vivodmas(D);
  Readln
end.
```

Варианты заданий

Задание 1. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить:

- сумму модулей отрицательных элементов массива;
- количество элементов массива, не принадлежащих интервалу (a, b) ;

	<p>с) наименьший из элементов массива, принадлежащих отрезку $[a, b]$;</p> <p>д) количество элементов массива, равных первому элементу;</p> <p>Задание 2. В двумерном массиве, состоящем из целочисленных элементов, поменять местами:</p> <p>а) в каждом столбце наибольший по модулю и последний положительный элементы;</p> <p>б) в каждом столбце первый и последний отрицательные элементы;</p> <p>с) в каждой строке наибольший и наименьший элементы;</p> <p>д) в каждом столбце первый принадлежащий отрезку $[a, b]$ и первый отрицательный элементы;</p> <p>Дополнительные задания</p> <p>1. Определить в одномерном массиве число соседств из двух чисел разного знака.</p> <p>2. Дан двумерный массив целых чисел. Поменять местами строку, содержащую максимум массива, со строкой, содержащей его минимум.</p>
Контрольный тест	<p>Сделать отчет по практическому заданию.</p> <p>Содержание отчета</p> <p>Отчет должен содержать:</p> <ol style="list-style-type: none"> 1. Название работы. 2. Цель работы. 3. Задание и его решение. 4. Вывод по работе. <p>Ответьте на вопросы:</p> <p>Контрольные вопросы</p> <ol style="list-style-type: none"> 1. Как описываются в языке Паскаль одномерный и двумерные массивы? 2. Может ли массив содержать разнотипные данные? 3. В каком порядке указываются индексы при обращении к элементам двумерного массива? 4. Привести пример массива, описание которого выглядит следующим образом: <code>var A: array [1..3, 20..24] of real.</code> 5. Можно ли при обработке двумерных массивов использовать однократные циклы? Если да, то приведите примеры. 6. Если в одномерном массиве проверяется «похожесть» его первой и второй части, то в каких границах надо писать оператор <i>for</i> для «прохождения» по этому массиву? 7. Каким образом надо находить первый и последний элементы одномерного массива, обладающие некоторым свойством (отрицательный, наибольший, входящий в интервал и <i>пр.</i>)?

Дата 27.02.2025

Подпись

Ф.И.О. преподавателя