Колледж имени Кумаша Нургалиева

Практические работы по дисциплине

«Основы алгоритмизации и программированию»

**(Turbo Pascal )**

**Содержание**

Практическая работа № 1: Cвойства алгоритмов 2

Практическая работа № 2: Cтандартные алгебраические функции 3

Практическая работа № 3: Работа с целыми числами 6

Практическая работа № 4: Команда ветвления и логические выражения. Условный оператор 8

Практическая работа № 5: Команда выбора 10

# Практическая работа № 1: Свойства алгоритмов

 *Алгоритм* — это точное предписание, определяющее процесс перехода от исходных данных к результату.

*Исполнитель* – это робот или человек, или любое другое устройство, умеющее выполнять определенный набор команд.

*Система команд исполнителя* – набор команд, которые исполнитель может выполнить.

*Свойства алгоритма*: дискретность, понятность, определенность (детерминированность), результативность, массовость.

*Дискретность:* описываемый процесс должен быть разбит на последовательность отдельных шагов. Возникающая в результате такого разбиения запись представляет собой упорядоченную совокупность четко разделенных друг от друга предписаний, образующих прерывную(дискретную) структуру алгоритма.

*Понятность:* используемые на практике алгоритмы составляются с ориентацией на определенного исполнителя. Чтобы составить такой алгоритм, надо ознакомиться с системой команд исполнителя.

*Однозначность алгоритма:* единственность толкования правил выполнения действий и порядка их выполнения. Таким образом, алгоритм не должен оставлять место произволу при его выполнении.

*Массовость* алгоритма означает возможность его применения для решения класса задач, предполагает его правильную работу при меняющихся в за­данных пределах значениях исходных данных.

*Результативность* алгоритма предполагает, что выполнение алгоритма долж­но приводить к получению определенного результата.

При структурном программировании важным компонентом является разбиение на модули.

*Модульность* – последовательность логических связанных операций, оформленных как отдельная часть программы.

Пример 1: Составить программу для нахождения суммы двух чисел.

Program primer1;

 Var a,b,sum:real;

Begin

 Write(‘Введите число А:’);readln(a);

 Write(‘Введите число В:’);readln(b);

 Sum:=a+b;

 Writeln(‘Сумма чисел =’, sum);

End.

**Задачи для самостоятельного решения:**

1. Даны длины ребер *a,b,c* прямоугольного параллелепипеда. Найти его объем *V=a\*b\*c* и площадь поверхности *S=2\*(a\*b+b\*c+a\*c).*
2. Найти длину окружности *L* и площадь круга *S* заданного радиуса *R: L=2\*3.14\*R, S=3.14\*R2*
3. Даны два числа *a* и *b*. Найти их среднее арифметическое: *(a+b)/2*
4. Даны два неотрицательных числа *a* и *b*. Найти их среднее геометрическое, то есть квадратный корень из их произведения 
5. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.
6. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их модулей.
7. Даны катеты прямоугольного треугольника *a* и *b*. Найти его гипотенузу *c* и периметр *P*.
8. Даны два круга с общим центром и радиусами *R1* и *R2 (R1>R2).* Найти площади этих кругов *S1* и *S2*, а также площадь кольца, внешний радиус которого равен *R1*, а внутренний радиус равен *R2*: *S1=3.14\*(R1)2, S2=3.14\*(R2)2, S3=S1-S2*
9. Дана длина окружности. Найти ее радиус *R* иплощадь *S* круга, ограниченного этой окружностью, учитывая, что *L=2\*3.14\*R, S=3.14\*R2*
10. Дана площадь S круга. Найти его диаметр D и длину L окружности, ограничивающей этот круг, учитывая, что *L=2\*3.14\*R, S=3.14\*R2.*

**Контрольные вопросы:**

1. Как описываются целочисленные, вещественные переменные?
2. При помощи какой инструкции можно вывести на экран текст или значение переменной?
3. При помощи какой инструкции можно запросить значение переменных во время работы программы??
4. При помощи какой инструкции можно присвоить переменной значение другой переменной, константы или формулы?
5. Как выглядит структура программы на языке Pascal в общем виде?

# Практическая работа № 2: Стандартные алгебраические функции

**Структура программы**

Программа на языке **Pascal ABC** имеет следующий вид:

**program** имя программы;
 раздел описаний
**begin**
  операторы
**end**.

Первая строка называется заголовком программы и не является обязательной. Раздел описаний может включать разделы описания переменных, констант, типов, процедур и функций, которые следуют друг за другом в произвольном порядке.Раздел подключения модулей и раздел описаний могут отсутствовать.Операторы отделяются один от другого символом "точка с запятой".

**Идентификаторы и служебные слова**

Идентификаторы служат в качестве имен программ, модулей, процедур, функций, типов, переменных и констант. Идентификатором считается любая последовательность латинских букв или цифр, начинающаяся с буквы. Буквой считается также символ подчеркивания "\_". Например, a1, \_h, b123 - идентификаторы, а 1a, ф2 - нет. Служебные слова служат для оформления конструкций языка и не могут быть использованы в качестве имен.

При разработке программ можно использовать стандартные, т. е. предостав­ляемые Turbo Pascal, функции. Ниже, в таблице, перечислены некоторые из них.

|  |  |
| --- | --- |
| Функция | Значение функции |
| Abs(n)  | Абсолютное значение n   |
| Sqrt(n)  | Квадратный корень n |
| Sqr(n) | Квадрат n |
| Sin (n)  | Синус *п* |
| Cos(n)  | Косинус  n  |
| Arctan(n)  | Арктангенс n |
| Ехр(n)  | Экспонента n |
| Ln(n)  | Натуральный логарифм n  |
| Round(n)  | Ближайшее к n целое  |
| Trunc(n)  | Целая часть n |
| Frac(n) | Дробная часть вещественного n, представленная как целое число |
| Frac(n) | Целая часть вещественного n, представленная как целое число |
| Exp(n\*ln(x)) | xn  число Х возведено в степень n  |

**Задачи для самостоятельного решения:**

1. Найти расстояние между двумя точками с заданными координатами x1 и x2 на числовой оси: [x2-x1].
2. Даны три точки А, В, С на числовой оси. Найти длины отрезков АС и ВС и их сумму.
3. Даны три точки на числовой оси. Точка С расположена между точками А и В. Найти произведение длин отрезков АС и ВС.
4. Даны координаты двух противоположных вершин прямоугольника: (x1,y1), (x2,y2). Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.
5. Найти расстояние между двумя точками с заданными координатами (x1,y1) и (x2,y2) на плоскости. Расстояние вычисляется по формуле 
6. Даны координаты трех вершин треугольника: (x1,y1), (x2,y2), (x3,y3). Найти его периметр, площадь, используя формулу для расстояния между двумя точками из предыдущего задания и формулу Герона для нахождения площади:, где p=(a+b+c)/2 – полупериметр.
7. Поменять местами содержимое переменных A и B и вывести новые значения A и B.
8. Даны переменные A, B, C. Изменить их значения, переместив содержимое A в В, В в С, С в А. Вывести новые значения переменных А, В, С.
9. Найти значение функции y=3\*x6-6\*x2-7 при данном значении x.
10. Найти значение функции y=4\*(x-3)6-7\*(x-3)2+2 при данном значении х.

**Контрольные вопросы:**

1. Воспроизведите по памяти таблицу стандартных функций.
2. В чем отличие типа integer от real?
3. Запишите структуру программы.
4. Алфавит языка Pascal, служебные слова и идентификаторы.
5. Комментарии в программе.

# Практическая работа № 3: Работа с целыми числами

**Обзор типов**

В Pascal ABC имеются следующие типы:

Типы integer, byte, char, перечислимый и диапазонный называются *порядковыми*. Только значения этих типов могут быть индексами массивов и фигурировать в качестве выражения-переключателя в операторе case. Переменная-параметр цикла for также должна иметь перечислимый тип.

Ко всем значениям порядкового типа применимы следующие функции:

Pred(x) возвращает значение, предшествующее x (к наименьшему значению не применяется);
Succ(x) возвращает значение, следующее за x (к наибольшему значению не применяется);
Ord(x) возвращает порядковое целое значение, соответствующее x. Для целых x возвращает само значение x, для символов char возвращает их код, а для элементов перечислимого типа - их номер (нумерация начинается с нуля).

Все порядковые типы, а также типы boolean, real и complex называются *простыми* типами.

**Оператор присваивания**

Оператор присваивания имеет вид:

переменная:= выражение

В качестве переменной может быть простая переменная, разыменованный указатель, переменная с индексами или компонент переменной типа запись. Выражение должно иметь тип, либо совпадающий с типом переменной, либо неявно к нему приводящийся

**Описание переменных и констант**

*Раздел описания переменных* начинается со служебного слова var, после которого следуют строки вида

список имен переменных: тип;

Имена в списке перечисляются через запятую. Например:

Var   a,b,c: integer;
   d: real;
  e,f: integer;
  s,s1: string;
  ch: char;

**Составной и пустой операторы**

Составной оператор предназначен для объединения нескольких операторов в один. Он имеет вид:

**begin** операторы **end**

Операторы отделяются один от другого символом ";". Служебные слова **begin** и **end**, окаймляющие операторы, называются *операторными скобками*.

Например:

s:=0; p:=1;
for i:=1 to 10 do
**begin** p:=p\*i;
  s:=s+p **end**

Перед **end** также может ставиться ";". В этом случае считается, что последним оператором перед **end** является *пустой оператор*, не выполняющий никаких действий.

**Задачи для самостоятельного решения:**

1. Дано двузначное число. Вывести вначале его левую цифру (десятки), затем правую цифру (единицы). Используйте операции целочисленного деления div и mod.
2. Дано двузначное число. Найти сумму и произведение его цифр.
3. Дано двузначное число, вывести число, полученное при перестановке цифр исходного числа.
4. Дано трехзначное число. Запишите его наоборот.
5. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.
6. Дано четырехзначное число. Две средние цифры убрали и приписали их в конец числа. Вывести результат.
7. Найти сумму цифр шестизначного числа.
8. Дано шестизначное число. Записать его наоборот.
9. Переставить в шестизначном числе цифры, стоящие в разряде десятков и единиц.
10. Дано трехзначное число. Вывести следующую информацию: количество сотен, количество десятков, количество единиц.

**Контрольные вопросы:**

1. Составить таблицу (тип, диапазон, принимаемые значения) для следующих типов переменных: integer, longint, shortint, byte, real.
2. Составить таблицу: арифметическая операция – назначение операции-тип переменных, поддерживающих операцию. Включите в таблицу функции div и mod.

# Практическая работа № 4: Команда ветвления и логические выражения. Условный оператор

Условный оператор имеет *полную* и *краткую* формы.

*Полная форма условного оператора* выглядит следующим образом:

**if** условие **then** оператор1
**else** оператор2

В качестве условия указывается некоторое логическое выражение. Если условие оказывается истинным, то выполняется оператор1, в противном случае выполняется оператор2.

*Краткая* форма условного оператора имеет вид:

**if** условие **then** оператор

Если условие оказывается истинным, то выполняется оператор, в противном случае происходит переход к следующему оператору программы.

В случае конструкции вида

**if** условие1 **then**  **if** условие2 **then** оператор1
**else** оператор2

**else** всегда относится к ближайшему предыдущему оператору **if**, для которого ветка **else** еще не указана. Если в предыдущем примере требуется, чтобы **else** относилась к первому оператору **if**, то необходимо использовать составной оператор:

**if** условие1 **then
begin**  **if** условие2 **then** оператор1
**end**
**else** оператор2

**Задачи для самостоятельного решения:**

1. Выяснить, является ли целое число, введенное с клавиатуры положительным.
2. Выяснить, является ли целое число, введенное с клавиатуры четным. (Остаток от деления на 2 равен нулю)
3. Выяснить, является ли целое число, введенное с клавиатуры кратным семи. (Остаток от деления на 7 равен нулю)
4. Даны координаты трех точек А, В, С, лежащие на одной прямой. Выяснить: лежит ли точка С между точками А и В? Известно, что А > В. Должно выполниться следующее сложное условие (С > В) и (С>А).
5. Даны координаты центра окружности О(x,y), координата точки С(хс,yc) и радиус окружности. Выяснить: принадлежит ли точка С множеству точек плоскости, расположенных внутри окружности, лежащих на окружности, лежащих за пределами окружности.
6. Проверить истинность высказывания: «Числа А и В имеют одинаковую четность». Ответ вывести в виде: «Высказывание истинно», «Высказывание ложно».
7. Найти большее из 4 чисел, введенных с клавиатуры.
8. Найти произведение минимального и максимального из четырех чисел, введенных с клавиатуры.
9. Написать программу, сообщающую: «Вы можете снять указанную сумму со счета», «Вы не можете снять указанную сумму со счета». Сумма вклада вводится в тексте программы случайным образом в заданном диапазоне от 0 до 10000.
10. Решить квадратное уравнение вида *ax2+bx+c=0.* Напоминаем, что возможны следующие варианты решений: дискриминант отрицательный – решений нет, дискриминант равен нулю – один корень x=(-b)/(2\*a), дискриминант больше нуля – два корня

x1=(-b+)/(2\*a),

x=(-b-)/(2\*a),

где D=b2-4\*a\*c.

**Контрольные вопросы:**

Полная форма команды ветвления

Неполная форма команды ветвления.

# Практическая работа № 5: Команда выбора

**Оператор выбора**

*Оператор выбора* выполняет одно действие из нескольких в зависимости от значения некоторого выражения, называемого *переключателем*. Он имеет следующий вид:

**case** переключатель **of**  список выбора 1: оператор1;
  ...
  список выбора N: операторN;
**else** оператор0
**end**;

Переключатель представляет собой выражение порядкового типа (целого, символьного, перечислимого или интервального), а списки выбора содержат константы совместимого типа. Как и в операторе **if**, ветка **else** может отсутствовать.

Оператор **case** работает следующим образом. Если в одном из списков выбора найдено текущее значение переключателя, то выполняется оператор, соответствующий данному списку. Если же значение переключателя не найдено ни в одном списке, то выполняется оператор по ветке **else** или, если ветка **else** отсутствует, оператор **case** не выполняет никаких действий.

Список выбора состоит либо из одной константы, либо из диапазона значений вида a..b (константа a должна быть меньше константы b); можно также перечислить несколько констант или диапазонов через запятую:

**case** DayOfWeek **of**
  1..5: writeln('Будний день');
  6,7: writeln('Выходной день');
**end**;

Списки выбора не должны пересекаться. Например, следующий фрагмент

**case** i **of**
  2,5: write(1);
  4..6: write(2);
**end**;

приведет к ошибке компиляции "Пересечение диапазонов меток в операторе **case**".

**Задачи для самостоятельного решения:**

1. Составить программу, которая по номеру дня недели выдает его название.
2. Составить программу, которая по номеру дня недели сообщает «Рабочий день», «Выходной день».
3. Составить программу, которая по введенному числу от 1 до 5 выдает название оценки: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «такой оценки нет».
4. Составить программу, которая по введенному возрасту в диапазоне от 1 года до 99 лет подписывает «год», «года», лет. При составлении программы пользуйтесь следующим алгоритмом: слово определяется по последней цифре возраста. Например: 25 лет (последняя цифра - 5). Исключение составляют числа: 11,12,13, 14,15. Все остальные числа можно определить при помощи алгоритма:

*Год 1*

*Года 2, 3, 4,*

*Лет 5, 6, 7, 8, 9, 0*

1. С клавиатуры вводят номер числа и номер месяца. Программа заменяет номер месяца его названием.

*Например:*

*Введите день: 25*

*Введите месяц: 5*

*25 мая*

1. Программа, в зависимости от номера месяца сообщает время года:

Например:

*Введите номер месяца: 12*

*Зима*

1. Программа после введенного числа, в диапазоне от 1 до 99 подписывает «копейка», «копейки», «копеек».
2. По цифре программа выдает название геометрической фигуры: 1-точка, 2-отрезок, 3-треугольник, 4-четырехугольник, 5-пятиугольник.
3. Найти значение функции y, которая принимает различные значения в зависимости от значения аргумента x:



1. В восточных календарях принят 12-летний цикл, в котором каждый год называется годом животного. Составить программу, по номеру года выдающую символ года. Информация для составления программы: крыса, бык, тигр, кролик, дракон, змея, лошадь, коза, обезьяна, петух, собака, свинья. Крыса – 1984 год.

**Контрольные вопросы:**

Принцип работы оператора выбора.