

Реализация основных алгоритмических
структур на языке Pascal.
(пособие для учащихся)

Автор: Тюленев Евгений Анатольевич

Место работы: МБОУ "Боровихинская сош"

Должность: Учитель

Оглавление

1 Введение.....	3
1.1 Выражение — алгоритм.....	3
1.2 Определение переменных. Операция присваивания.....	4
1.3 Общий вид программы на языке Pascal.	6
1.3.1 Стандартный ввод и вывод.....	7
2 Управляющие операторы.....	8
2.1 Логические выражения.....	8
2.2 Условный оператор.....	10
2.3 Оператор выбора.....	13
2.4 Циклы.....	14
2.4.1 Цикл с предусловием.....	14
2.4.2 Цикл с постусловием.....	17
3 Массивы.....	17
3.1 Заполнение массивов.....	18
3.2 Обработка массива.....	21
4 Файлы.....	24
5 Приложения.....	26
5.1 Необходимое программное обеспечение.....	26
6 Список литературы.....	27

1 Введение.

Исполнитель — это объект способный выполнять определенный набор команд.

Алгоритм — это последовательность команд, выполнение которой ведет к решению задачи.

Требования к алгоритму:

1. *Понятность*: инструкции алгоритма должны быть понятны исполнителю.
2. *Замкнутость*: в результате выполнения алгоритма должна быть решена именно данная задача.
3. *Дискретность*: выполнение очередной инструкции не начинается, пока не закончится выполнение текущей.
4. *Конечность*: выполнение инструкций алгоритма должно когда-то закончиться.

1.1 Выражение — алгоритм.

Для того, чтобы вычислить значение выражения $3(12-6)$ нужно действовать по алгоритму:

1. из 12 вычесть 6;
2. результат первого действия умножить на 3;
3. сообщить результат второго действия.

Программирование можно рассматривать как процесс составления выражений.

Требования к записи выражений:

1. расставлять знаки всех операций.
2. дробная черта заменяется делением.
3. у десятичной дроби целая от дробной части отделяется точкой.
4. если необходимо, то числитель и знаменатель заключается в скобки.

Обозначение арифметических операций:

+	сложение
-	вычитание
*	умножение
/	деление
div	Деление на цело.
mod	Нахождение остатка

Например: $\frac{5,6 - 11(0,12 + 4)}{6,11 + 3,2^2}$

на языке программирования записывается так: $(5.6 - 11 * (0.12 + 4)) / (6.11 + 3.2 * 3.2)$.

Упражнения.

1. Запишите на языке программирования

а) $56,5(11 - 5,8)$

в) $(4,6 - 1)^4 - 87,3$

б) $\frac{23 - 1,123(3,5 + 6)}{5^3 - 6}$

г) $1,1 + \frac{89^2}{5,6} + 7$

2. Запишите на языке алгебраических формул:

а) $567.5 * 4 - 5 * 5$;

в) $(4 - 6.78) / (23 * 5)$;

б) $33.7 * (23 - 4.5 * 4.5)$;

г) $5.45 - 24 / 67 + 4.5$.

3. Вычислите.

а) $23/5 - 8*3$;

в) $37\%5 - 2+6$;

б) $6-78/(4+2)$;

г) $(56-23\%7)+76/4$;

4. Составьте выражения, и запишите их на языке программирования:

а) Найти площадь и периметр прямоугольника если его длина — 5м, а ширина 20,45см.

б) Автомобиль движется со скоростью 5.4 м/с. Сколько времени ему понадобится, чтобы проехать 55 км.

в) Сторона куба 3,4см. Сколько метров составляет объем куба.

г) С полуночи прошло 5 ч 34 м 20 с. Сколько секунд осталось до полудня.

1.2 Определение переменных. Операция присваивания.

Для хранения информации используют переменные.

Переменная — это область памяти.

Прежде чем использовать переменную, ее нужно определить. Переменные определяются в специальном блоке `var`.

Общий вид определения переменной:

```
var имя1, имя2, ... : тип;  
...
```

Обозначения основных типов:

```
char — символ;  
byte — целое число из [0;255];  
integer — целое число;  
real — действительное число;  
string — последовательность(строка) символов.
```

Если нужно определить несколько переменных одного типа, то их имена* перечисляют

* Имя переменной не может начинаться с цифры, в именах нельзя использовать пробел, знаки препинания, скобки и

через запятую.

Примеры:

var

```
a,b:integer;//две переменных для целых чисел.  
x,y,radius:real;//три переменных для действительных чисел.  
asc,ch,c:char;//три переменных для символов.
```

Основной способ записи значения в переменную — *операция присваивания*.

Синтаксис операции присваивания:

```
имя := выражение;
```

Например:

```
result := 2 * ( 56.4 - 4 );//в result будет записано 104.8
```

Найти площадь круга S и длину окружности C , если радиус 4,67см.

Переменные S и C должны быть действительного типа. Расчетные формулы $S = \pi \cdot r^2$; $C = 2 \cdot \pi \cdot r$, таким образом:

var

```
s,c:real;
```

begin

```
s = 3.14 * 4.67 * 4.67;
```

```
p = 2 * 3.14 * 4.67;
```

end.

Блок операторов основной программы заключается в операторные скобки:

begin — скобка открывается

end. - скобка закрывается(точка обязательна)

Упражнения.

5. Определите:

- а) две переменных для целых чисел;
- б) три переменных для действительных чисел и одну для символа;
- в) пять переменных для символов и две для действительных чисел;
- г) две для целых чисел, одну для действительного числа и три для символов.

6. Для хранения какой информации определены переменные:

- а) var x,y: integer; a,b,c:real;

Многие другие символы. Чтобы не ошибиться, используйте в именах только буквы латинского алфавита.

б) var v,n,d:char; d,b: integer;

в) var x,t,b:real; m,n:integer; d,r,t,rr,yy:string;

г) var x,y,z:real; a,b,c,d:integer; f,l:char;

7. Что будет записано в переменные, и какого типа они должны быть:

а) $tt := 6 * 7 - 11$;

б) $result := 5 \bmod 2 + 7$;

в) $y := 5,67 - 8/7$;

г) $r := 34 \div 2 + 5$;

8. Составьте выражение, ведущее к решению задачи, запишите фрагмент программы с определением необходимых переменных:

а) Найти скорость V , если расстояние 2345 км, пройдено за 10 ч. (движение равномерное и прямолинейное);

б) Определить сколько минут MM содержится в 20 часах 125 секундах;

в) Найдите остаток r и неполное частное q при делении числа 23465 на 234;

г) За карандаш и тетрадь заплатили 15 рублей. Сколько стоит карандаш, если цена тетради составляет 70% стоимости покупки.

1.3 Общий вид программы на языке Pascal.

Общий вид программы:

```
program имя_программы;  
uses //подключение модулей  
type  
    //блок определения типов  
const  
    //блок определения глобальных постоянных  
var  
    //блок определения глобальных переменных  
  
//Блок определения глобальных процедур и функций.  
  
begin  
    //блок операторов  
end.
```

Если нет необходимости в использовании какого-либо блока его пропускают.

Например:

```

program Hello;
begin
    WriteLn('Привет, всем!!!');
end.

```

1.3.1 Стандартный ввод и вывод.

В пункте 1.3 в программе Hello используется процедура WriteLn. Она предназначена для вывода информации на консоль. Существуют два варианта этой процедуры:

```

write(выражение1, выражение2, ...); {выводить значения выражения друг за другом.}
writeln(выражение1, выражение2, ...); {выводить значения выражения друг за другом и начать новую строку}.

```

Для запроса ввода значений переменных используют процедуры *Read* или *ReadLn*.

Синтаксис вызова процедур:

```

Read(имя_переменной1, имя_переменной2, ...); {читаются данные, введенные через пробел, завершающий перевод строки не удаляется из буфера клавиатуры.}
ReadLn(имя_переменной1, имя_переменной2, ...); {читаются данные, введенные через пробел, завершающий перевод строки удаляется из буфера клавиатуры.}

```

Примеры.

1. Спросить имя и поздравиться.

```

program Hello1;
var
    s:string;
begin
    Write('Как Вас зовут?');
    ReadLn(s);
    WriteLn('Здравствуйте, ',s);
end.

```

2. Найти объем куба по его ребру.

```
program sq_round;
var
    a,V:real;
begin
    Write('Введите длину ребра:');
    ReadLn(a);
    V:=a*a*a;
    WriteLn('V=',V);
end.
```

Упражнения.

Напишите программы решения задач, организовав ввод и вывод:

9. Вычислить значения выражений (вывод осуществить в виде формул):

а) $5a+4b$

в) $4,6(x-y)^2+9$

б) x^3-4x+y

г) $\frac{4x-7}{5}$

10. Найти площадь треугольника:

а) по стороне и соответствующей высоте;

в) Найти последнюю цифру целого числа.

б) по трем сторонам;

г) Найти гипотенузу зная катеты.

11. Найти количество теплоты необходимое для нагревания 5 литров воды от 20 до 80 градусов Цельсия.

2 Управляющие операторы.

Для того, чтобы указать исполнителю на необходимость принятия решения о выполнении какого-либо действия или повторения действия служат *управляющие операторы*.

К основным управляющим операторам относятся:

1. Условный оператор.
2. Оператор выбора.
3. Оператор цикла.

В управляющих операторах используют *логические выражения*.

2.1 Логические выражения.

Логическое выражение состоит из *высказываний* и логических операций над ними.

Любое элементарное высказывание или логическое выражение может принимать одно из двух значений: *истина* или *ложь*. На языке Pascal истина — *true*, ложь — *false*.

Самые распространенные высказывания - это *равенства и неравенства, их системы и совокупности*.

Например:

$$5 > 6, 7 < 90, 2 * 3 < 6, 45 = 45$$

$$\begin{cases} 5 > x \\ y \geq 12 \end{cases}$$

$$\begin{cases} 3x < 4 \\ m \leq 2 \end{cases}$$

Основные логические операции и их *таблицы истинности*.

Каждая строка *таблицы истинности* показывает результат логической операции при указанных значения *операндов*.

1. Операция "не"

A	Not A
T	F
F	T

2. Операция "и"

A	B	A and B
F	F	F
T	F	F
F	T	F
T	T	1

3. Операция "или"

A	B	A or B
F	F	F
T	F	T
F	T	T
T	T	T

Примеры вычисления значений логических выражений:

$$\text{not } (4 > 5) = \text{not } F = T$$

$$\text{not } ((45 > 6) \text{ and } (9 < 11)) = \text{not } (T \text{ and } T) = !T = F$$

Логические выражения с *переменными*.

Если в равенстве или неравенстве один из операндов — *переменная*, то значение выражения нельзя найти до тех пор, пока не задано значение этой *переменной*.

Например:

$$(x > 2) \text{ or } (4 < x) = ?$$

при $x=5$, $(5>2)\text{or}(4<5)=T$ or $T=T$

Контрольные вопросы.

1. Что такое логическое выражение?
2. Что, чаще всего, выступает в роли высказывания? Назовите основные логические операции.
3. Запишите, по памяти, таблицы истинности основных логических операций.

Упражнения.

12. Запишите логические выражения на языке С:

а)
$$\begin{cases} 5-x \geq 1 \\ x < 0 \end{cases}$$

б)
$$\begin{cases} 6x-1 > 7 \\ x < 5 \end{cases}$$

в)
$$\begin{cases} 4,5a-1 \leq 4 \\ 6x > 11 \\ y < 56 \end{cases}$$

г)
$$\begin{cases} x^2-1 \geq 1 \\ y > 2x \\ x < 2 \end{cases}$$

13. Найдите значение логического выражения, при данных значениях переменных.

а) $\text{Not}(5x>11)$, при $x=1$; 34;

б) $(y+x<45)\text{and not}(y>x)$, при $x=2$; $y=17$;

в) $\text{not}((d>2)\text{and}(d+1<6))$, при $d=1$; 5; 0;

г) $(a+b+c \leq 0)$ or $\text{not}(a+b>c)$, при $a=-1$; $b=2$; -3.

14. Петя и Коля выше Васи, но Коля ниже Пети. Каково значение логического выражения: $(K>V)$ or $(V>\Pi)$, где первой буквой имени обозначен рост.

2.2 Условный оператор.

Если необходимо принять решение о выполнении того или иного оператора, то уместно использовать условный оператор.

Например, перед вычислением значения выражения $\frac{1}{x}$ нужно убедиться в том, что значение x не равно 0.

Синтаксис условного оператора:

```
if высказывание then
    begin
        //выполняется если истина
    end
else
    begin
        //выполняется если ложь
    end;
```

Пример:

Найти значение выражения $\frac{5a-b}{a-b}$.

Очевидно, что значение выражения можно вычислить только при условии, что $a-b \neq 0$.

```
program calc;
var
    a,b,r:real;
begin
    Write('Введите a,b:');
    Read(a,b);
    if a <> b then
        begin
            r:=(5*a-b)/(a-b);
            WriteLn('результат: ',r);
        end
    else
        begin
            WriteLn('Деление на ноль.');
```

Если в качестве оператора1 или оператора2 необходимо вызвать несколько операторов, то они берутся в операторные скобки:

begin - скобка отрывается

end - скобка закрывается

Когда при неверном условии требуется просто перейти к выполнению операторов после условного, то else часть можно упустить:

```
if высказывание then
    begin
        //выполнять если истина.
    end
```

Этот оператор называют *сокращенным условным оператором*.

Например, найти наименьшее среди трех чисел.

```

program min_3;
var
    a,b,c,min:real;
begin
    Write('Введите три числа:');
    min:=a;
    if min>b then min:=b;
    if min>c then min:=c;
    WriteLn('min=',min);
end.

```

Для организации вычислений полезно использовать библиотеку математических функций:

Синтаксис функции	Результат.
sqrt(x)	\sqrt{x}
sin(x)	sin(x)
cos(x)	cos(x)
tan(x)	tg(x)
exp(x)	e^x
abs(x)	x

Например, выражение $\frac{\sin x}{\sqrt{x+4}}$ будет записано так: sin(x)/sqrt(x+4).

Контрольные вопросы.

1. Запишите общий вид сокращенного и полного условного оператора.
 2. Объясните: как компьютер выполняет условный оператор?
 3. Перечислите основные математические функции.
-

Упражнения.

15. Что будет храниться в переменной x после выполнения кода:

- a) x := 7; if x>3 then x := x-1 else x := x+1;
- б) x := 34; x := x+5; if x<=40 then x := 0 else x := x+1;
- в) x:=11;y := 5; if x>2*y then x := x+y else x := 3;
- г) * x:=5; if(x+1>=6) x:=0;

16. Найти значение выражения:

- | | |
|--|---|
| <ol style="list-style-type: none"> a) $\frac{r-m}{r+1}$ б) $\sqrt{4x-5}$ | <ol style="list-style-type: none"> в) $\frac{2}{x} - \sqrt{x-1}$ г) $r + \sqrt{\frac{4x-2}{x}}$ |
|--|---|

- | | |
|----------|----------|
| a) [a;b] | в) (a;b] |
| б) [a;b) | г) (a;b) |

Иногда необходимо выполнять операторы в зависимости от нескольких различных значений выражения. В этом случае удобнее использовать *оператор выбора*.

```
case выражение of  
    множество значений1: оператор 1;  
    множество значений2: оператор 2;  
    ...  
    множество значенийn: оператор n;  
else  
    оператор;  
end;
```

Пример: Вывести название целого числа из отрезка [1;5].

```

program num_name;
var
    a:integer;
begin
    Write('Введите число:');
    case a of
        1:WriteLn('один');
        2:WriteLn('два');
        3:WriteLn('три');
        4:WriteLn('четыре');
        5:WriteLn('пять');
        else WriteLn('Не знакомое число!');
    end;{case}
end.

```

2.4 Циклы.

Цикл — это управляющая конструкция, организующая повторение выполнения блока операторов, пока верно условие.

Различают циклы *с предусловием* и *постусловием*.

Чаще всего используют циклы с предусловием.

2.4.1 Цикл с предусловием.

Основной оператор:

```

while высказывание do
    begin
        //повторять пока истинно
    end;

```

Пример:

Вывести на консоль все числа последовательности Фибоначчи, не превосходящие заданное число.

Последовательность Фибоначчи:

1 1 2 3 5 7 12 ...

Начиная с третьего каждый элемент равен сумме двух предыдущих: $a_i = a_{i-1} + a_{i-2}$.

```

program fib;
var
    a,b,c,d:integer;
begin
    Write('Введите натуральное число?');
    ReadLn(c);
    a:=1; b:=1;
    Write(a, ' ',b);
    while b<c do
        begin
            d:=a+b;
            a:=b; b:=d;
            Write(' ',b);
        end;
    end.

```

Оператор *for/to(downto)*. Когда заранее известно количество повторений удобнее использовать оператор *for/to(downto)*.

Первый вариант (счет ведется в возрастающем порядке).

```

for ипс:=нз to кз do
    begin
        //повторять на каждый счет.
    end;

```

где:

ипс — имя переменной счетчика

нз — начальное значение

кз — конечное значение

Пример:

Вывести на экран n первых неотрицательных чисел кратных трем.

Последовательность: 0 3 6 9 ...

```

program three;
var
    i,n:integer;
begin
    Write('Введите число:');
    ReadLn(n);
    for i:=1 to n do
        begin
            WriteLn(i, '-е число:', 3*i);
        end;
    end.

```

Второй вариант(счет ведется в убывающем порядке).

```

for ипс:=нз downto кз do
    begin
        //повторять на каждый счет.
    end;

```

Пример.

Вывести на экран первые n нечетных натуральных чисел в обратном порядке.

```

program odd;
var
    n,i:integer;
begin
    Write('Введите количество:');
    ReadLn(n);
    for i:=n downto 1 do
        begin
            WriteLn(i, '-е число:', 2*i-1);
        end;
    end.

```

Упражнения.

22. Найти значение выражения

а) $1*2*3*...*n$;

б) $1+2+3+...+n$;

в) $x+2*x+3*x+...+n*x$;

г) $\frac{1}{2x} + \frac{2}{3x} + \dots + \frac{n}{(n+1)x}$

23. Вывести на консоль первые n нечетных чисел.

24. Вывести на консоль значения функции из данного интервала, с указанным шагом:

- а) $y=3x-1$ $[1;5]$, $шаг=1$
 б) $y=2\sin(x)$ $(-\pi;\pi)$, $шаг=0,1$
 25. Вывести все цифры целого числа.
 26. Записать цифры числа в обратном порядке.
 27. Найти сумму первых n элементов последовательности, если $a_i = \begin{cases} 2i-1 & i \text{ четное} \\ i-2 & i \text{ нечетное} \end{cases}$
 28. Для последовательности Фибоначчи найти
 а) сумму элементов не превосходящих в) сумму первых n элементов
 данного целого положительного числа; г) количество четных элементов меньших
 б) первые n элементов; данного целого положительного числа.

2.4.2 Цикл с постусловием.

Синтаксис оператора:

```
repeat
    //повторять пока не выполнится высказывание.
until высказывание;
```

Пример: Добиться ввода четного числа

```
program in_2n;
var n:integer;
begin
    repeat
        Write('Введите число:');
        Read(n);
    until n mod 2 = 0;
    WriteLn(n, ' - четное число');
end.
```

Упражнения.

29. Добиться ввода задуманного числа.
 30. Трехзначное число — пароль. Добиться ввода пароля не более чем с 4 попыток.
 31. *Реализовать игру "Угадай число".

3 Массивы.

Для хранения многих однотипных данных удобно использовать составную переменную — массив.

Массив — это последовательность однотипных элементарных переменных.

Вся последовательность имеет имя, а конкретная элементарная переменная имеет номер.

В языке C нумерация элементов массива ведется с 0.

Пример:

массив **a** из 11 целых чисел

0	1	2	3	4	5	6	7	8	9	10
345	12	56	98	11	5	98	4	11	8	6

$a[7] = 4$

Определение массива

имя: **array** [ни..ки] **of** тип;

Например:

```
a:array[1..50] of integer; //массив из 50-ти целочисленных переменных
x:array[1..10] of real; //массив из 10-ти действительных переменных
s:array[1..300] of char; //массив из 300 символов (строка).
```

Виды задач на массивы:

1. Заполнение массива;
2. Обработка массива.

3.1 Заполнение массивов.

По формуле n-го элемента.

Заполнить массив из 12 целых чисел первыми четными. (формула четного числа $2*n$, где n — номер четного числа)

```

program ex1;
var
    a:array[1..12] of integer;
    i:integer;
begin
    for i:=1 to 12 do
        begin
            a[i]:=2*i;
        end;
    //Вывод массива.
    for i:=1 to 12 do
        begin
            Write(' ',a[i]);
        end;
    end.

```

По рекуррентному соотношению

Рекуррентное соотношение — это зависимость очередного элемента от предыдущих.

Заполнить массив из 8 действительных чисел, если $a[0]=1$, $a[i]=2*a[i-1]+1$

```

program ex2;
var
    a:array[1..8] of real;
    i:integer;
begin
    a[1]:=1.1;
    for i:=2 to 8 do
        begin
            a[i]:=2*a[i-1]+1;
        end;
    for i:=1 to 8 do
        begin
            Write(' ',a[i]);
        end;
    end.

```

Из потока ввода.

Пример

Заполнить массив из 5 целых чисел со стандартного потока ввода (клавиатура).

```
program ex3;
var
    a:array[1..5] of integer;
    i:integer;
begin
    WriteLn('Введите 5 целых чисел. ');
    for i:=1 to 5 do
        begin
            ReadLn(a[i]);
        end;
    for i:=1 to 5 do
        begin
            Write(' ',a[i]);
        end;
    end.
```

Случайными числами.

Случайное число получается в результате функции `rand()` из библиотеки *stdlib.h*. Это число очень большое.

Чтобы получить число из отрезка $[a;b]$ можно использовать формулу $rand() \% (b-a) + a$;

Пример.

Заполнить массив из десяти целых чисел случайными двузначными числами.

```

program ex4;
var
    a:array[1..10] of integer;
    i:integer;
begin
    for i:=1 to 10 do
        begin
            a[i]:=Random(90)+10;
        end;
    for i:=1 to 10 do
        begin
            write(' ',a[i]);
        end;
    end.

```

Упражнения.

32. Заполнить массив:

а) int a[17] числами кратными 5;

б) int b[9] первыми элементами последовательности Фибоначчи.

в) float x[7] с клавиатуры.

г) int m[10] случайными трехзначными числами.

33. Заполнить массив int a[20] элементами последовательности $a_i = \begin{cases} i+1 & i < 3 \\ 2*a_{i-1} & i \geq 3 \end{cases}$

34. Заполнить массив int a[14] элементами последовательности:

а) 1 0 1 0 1 0 ...

б) 1 2 3 1 2 3 1 2 3 ...

в) 1 2 3 2 3 4 3 4 5 ...

г) * 1 0 1 1 0 1 1 1 ...

3.2 Обработка массива

Поиск элементов.

Пример

Найти минимальный элемент в массиве из 12 двузначных чисел.

```

program Search;
var
    a:array[1..12] of integer;
    i,Min:integer;
begin
    for i:=1 to 12 do
        begin
            a[i]:=Random(90)+10;
            Write(a[i], ' ');
        end;
    Min:=a[1];
    for i:=2 to 12 do
        begin
            if Min>a[i] then Min:=a[i];
        end;
    WriteLn;
    WriteLn('минимальный равен:',min);
end.

```

Поиск и замена.

Пример.

В массиве из 15-ти трехзначных чисел заменить все четные числа нулями.

```

program SearchReplace;
var
    a:array[1..15] of integer;
    i:integer;
begin
    for i:=1 to 15 do
        begin
            a[i]:=Random(900)+100;
            Write(a[i], ' ');
        end;
    Min:=a[1];
    for i:=1 to 15 do
        begin
            if a[i] mod 2 = 0 then a[i]=0;
        end;
    WriteLn;
    for i:=1 to 15 do
        begin
            Write(a[i], ' ');
        end;
    end.

```

Изменение порядка элементов.

Пример.

Упорядочить массив целых неотрицательных чисел меньших 50 `int a[10]` в порядке возрастания.

Упражнения.

35. Дан массив двузначных чисел `int a[10]`. Найти

- а) Номера четных элементов;
- б) Элементы большие 50;
- в) Элементы, оканчивающиеся на 5.
- г) Элементы, сумма цифр которых больше 10.

36. Дан массив трехзначных чисел `int a[15]`. Определить какие из них не могут входить в `ip` адрес и заменить их нулями.

37. В массиве целых неотрицательных чисел меньших 20 `int a[30]` найти:

- а) минимальный.
- б) максимальный

в) сумму минимального и максимального.

г) * первый элемент больший минимального.

4 Файлы.

Файл — это совокупность взаимосвязанных значений.

Работа с файлом состоит из:

1. Открытия файла.
2. Чтение или запись информации.
3. Закрытие файла.

Для работы с файлом понадобится файловая переменная, в которой хранится служебная информация. Эта переменная будет использоваться в функциях работы с файлом.

Определение файловой переменной¹:

```
имя:text; - текстовый файл
```

Например:

```
var  
    myf:text;
```

Открытие файла:

`Assign(f, 'имя файла');` - связывание файловой переменной с файлом

`Reset(f);` - открытие файла. Если файл существует то указатель помещается в конец файла

`Rewrite(f);` - открытие файла. Если файл существует то он очищается.

Например:

```
Assign(myf, 'data.txt');  
Rewrite(myf);
```

Запись в файл:

```
Write(f, выражение1, выражение2, ...);
```

```
WriteLn(f, выражение1, выражение2, ...);
```

Чтение из файла:

```
Read(f, имя_переменной1, имя_переменной2, ...);
```

```
ReadLn(f, имя_переменной1, имя_переменной2, ...);
```

Процедуры аналогичны процедурам чтения и записи из стандартных файлов INPUT (связан с устройством ввода) OUTPUT (связан с устройством вывода). Первый параметр имя файловой переменной.

¹ В школьном курсе достаточно рассмотреть текстовые файлы.

Заккрытие файла:

```
close(f);
```

Примеры:

В файл data.txt записать 10 первых нечетных чисел.

```
program file1;
var
    myf:text;
    i:integer;
begin
    Assign(myf, 'data.txt');
    Rewrite(myf);
    for i:=1 to 10 do
        begin
            Write(myf, 2*i, ' ');
        end;
    Close(myf);
end.
```

Из файла data.txt прочитать 7 целых чисел, увеличить их на 5 и записать в файл data1.txt

```
program file1;
var
    myf, f1:text;
    i, a:integer;
begin
    Assign(myf, 'data.txt');
    Assign(f1, 'data1.txt');
    Reset(myf);
    Rewrite(f1);
    for i:=1 to 7 do
        begin
            Read(myf, a);
            Write(f1, a+5, ' ');
        end;
    Close(myf);
    Close(f1);
end.
```

Упражнения.

1. В файл data.txt активного каталога записать 20 случайных трехзначных чисел через пробел.
2. Найти наименьшее трехзначное число из файла data.txt
3. Числа из файла data.txt умножить на 4 и записать в файл data1.txt
4. Объединить файлы data.txt и data1.txt в один файл result.txt
5. Загрузить числа из result.txt в массив целых чисел. Полученный массив упорядочить и вывести в файл result.txt

5 Приложения.

5.1 Программное обеспечение.

Для работы можно использовать любую IDE поддерживающую язык программирования Pascal.

Рекомендую использовать Geany+FPC. В AltLinux мастер установлены по умолчанию. Для Windows отдельно скачать и установить Geany [отсюда](#), Free Pascal [отсюда](#)