

Государственное бюджетное образовательное учреждение
дополнительного профессионального образования
«НИЖЕГОРОДСКИЙ ИНСТИТУТ РАЗВИТИЯ ОБРАЗОВАНИЯ»

М. Ю. ВТЮРИН, М. И. ГОЛУНОВА, Е. П. МОСТИПАН

ПРАКТИКУМ
по ПРОГРАММИРОВАНИЮ
на ЯЗЫКЕ ПАСКАЛЬ

Часть I

Нижний Новгород
Нижегородский институт развития образования
2014

УДК 372.016:004

ББК 74.263.2

В87

А в т о р ы

М. Ю. Втюрин, канд. физ.-мат. наук, доцент кафедры теории и методики обучения информатике ГБОУ ДПО НИРО;

М. И. Голунова, канд. пед. наук, зав. кафедрой теории и методики обучения информатике ГБОУ ДПО НИРО;

Е. П. Мостипан, старший преподаватель кафедры теории и методики обучения информатике ГБОУ ДПО НИРО

*Рекомендовано к изданию
научно-методическим экспертным советом
ГБОУ ДПО НИРО*

Втюрин, М. Ю.

В87 Практикум по программированию на языке Паскаль / М. Ю. Втюрин, М. И. Голунова, Е. П. Мостипан. — Н. Новгород : Нижегородский институт развития образования, 2014. — 37 с.

ISBN 978-5-7565-0607-5

Предлагаемое пособие предназначено для освоения основных алгоритмических структур языка программирования Паскаль.

Издание адресовано слушателям курсов повышения квалификации в Нижегородском институте развития образования и призвано оказать помощь учителям информатики в процессе изучения языка программирования Паскаль.

УДК 372.016:004

ББК 74.263.2

ISBN 978-5-7565-0607-5

© ГБОУ ДПО «Нижегородский институт развития образования», 2014

Предлагаемое пособие призвано помочь освоить основные алгоритмические структуры языка программирования Паскаль и ориентировано на учителей информатики, приступающих к изучению данного языка программирования.

Первая глава «Основы работы в среде Free Pascal» знакомит с основами работы в среде Free Pascal. В ней рассмотрены основные элементы интерфейса среды программирования. В главах 2, 3, 4 освещены вопросы структурного программирования. Рассмотрены линейные алгоритмы, алгоритмы ветвления и циклические алгоритмы.

Каждая глава организована следующим образом: сначала приводится краткая справка об основных операторах, необходимых для освоения соответствующей темы, затем следует раздел с примерами программ и дается практический материал — упражнения.

Рекомендуется, приступая к работе, сначала изучить описываемые в соответствующем разделе программы, реализовать их самостоятельно на компьютере: ввести исходные данные и получить результаты, приведенные в пособии. И уже после этого самостоятельно выполнять приведенные в пособии упражнения по изучаемой теме.

■ Основы работы в среде Free Pascal

Структура программы на языке Паскаль

Типовая программа на языке Паскаль состоит из следующих разделов:

- ▣ Заголовок программы
- ▣ Раздел описаний:
 - Раздел описания констант
 - Раздел описания типов
 - Раздел описания переменных
 - Раздел описания процедур и функций
- ▣ Тело программы.

На языке программирования структура выглядит следующим образом:

```
program <имя_программы>;  
const  
    <описание_констант>;  
var  
    <описание_переменных>;  
begin  
    <операторы_языка>;  
end.
```

Простая программа на языке Паскаль

Программа печатает приветствие на экране:

```
program n1;  
begin  
    writeln ('Здравствуй, МИР!!!');  
end.
```

Порядок работы в среде *Free Pascal*

1. Создаем на жестком диске рабочую папку, например: D:\Программы.


2. Запускаем *Free Pascal* с рабочего стола  (используя данный ярлык) или из списка программ.



Рис. 1

3. В меню *File* выбираем опцию *New*.

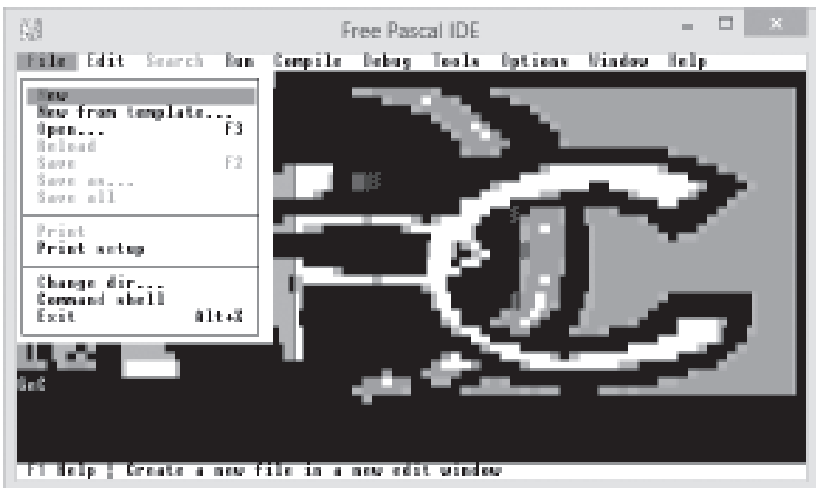


Рис. 2

4. Открывается редактор для создания программы.

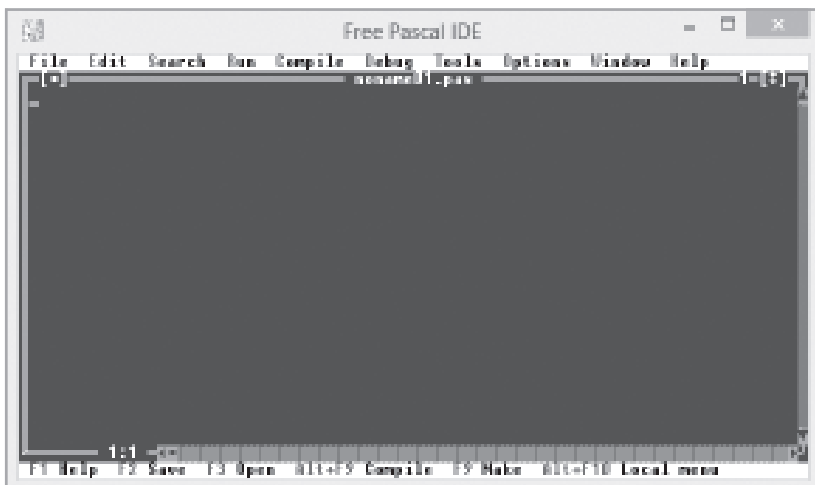


Рис. 3

5. В меню *File* выбираем опцию *Change dir...*

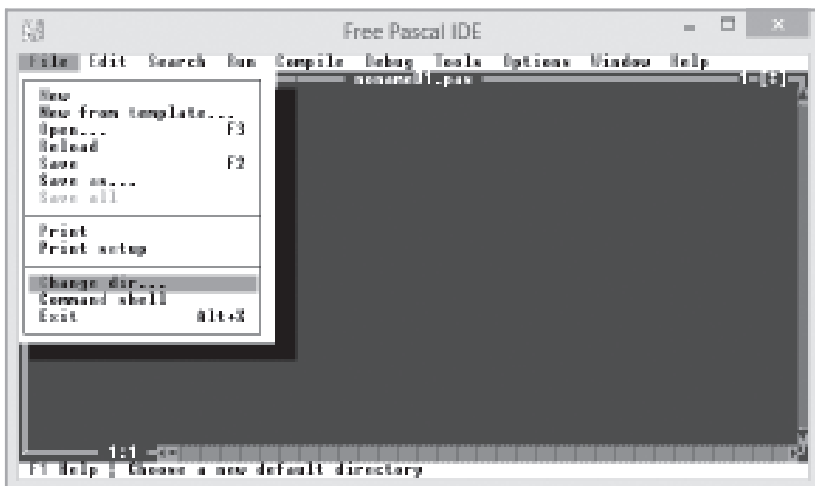


Рис. 4

6. Указываем путь к папке *Программы* на жестком диске.

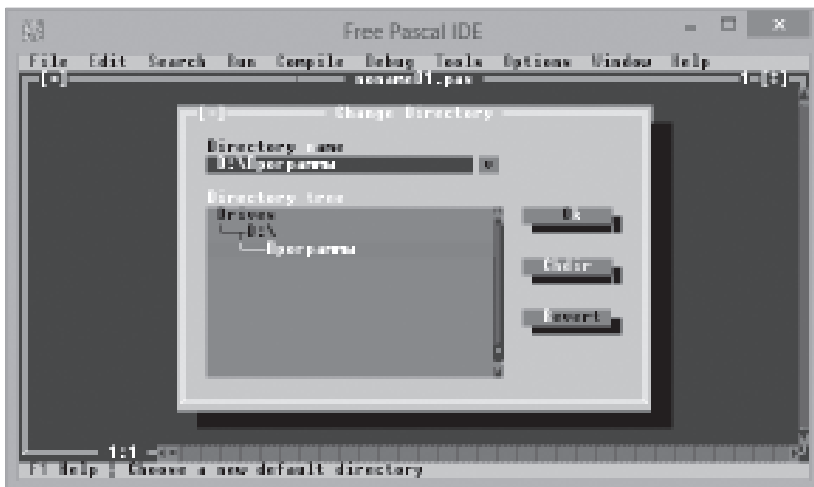


Рис. 5

7. В меню *File* выбираем опцию *Save*.

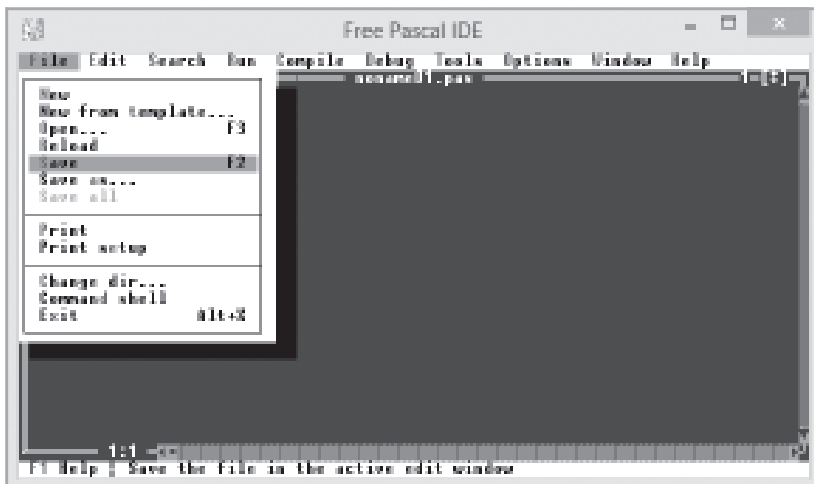


Рис. 6

8. Даем имя нашему файлу с текстом программы.



Рис. 7

9. Осуществляем набор текста программы, затем в меню *File* выбираем опцию *Save*.

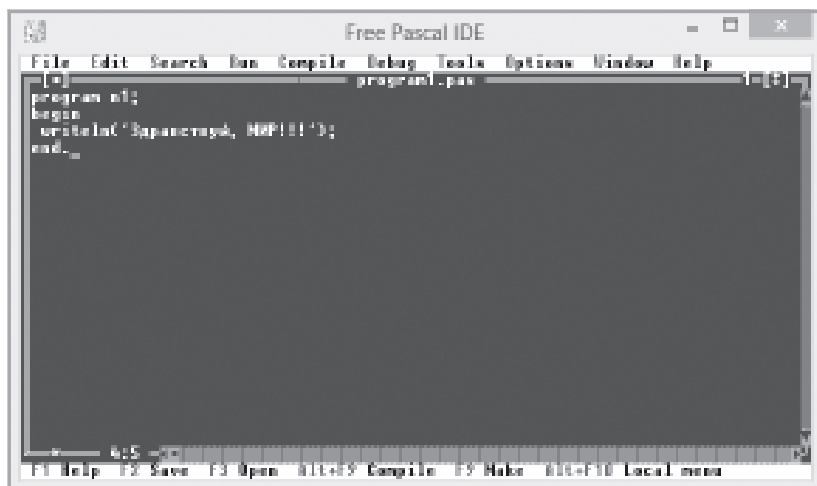


Рис. 8

10. В меню *Run* выбираем опцию *Run*.

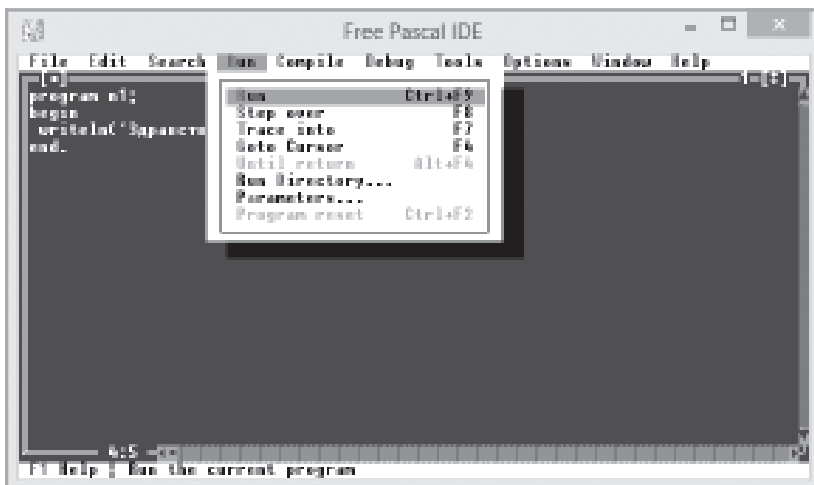


Рис. 9

11. Для просмотра результата выполнения программы одновременно нажимаем клавиши *Alt* и *F5*.

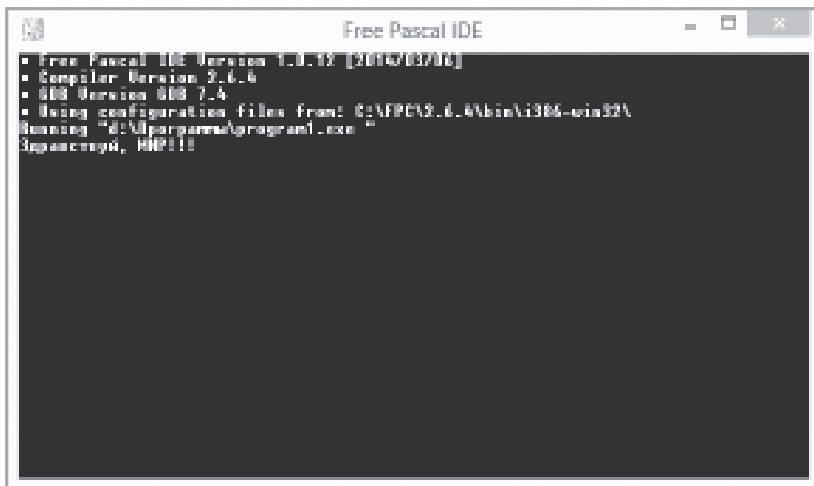


Рис. 10

12. Нажав любую клавишу, возвращаемся в окно редактора текста программы.

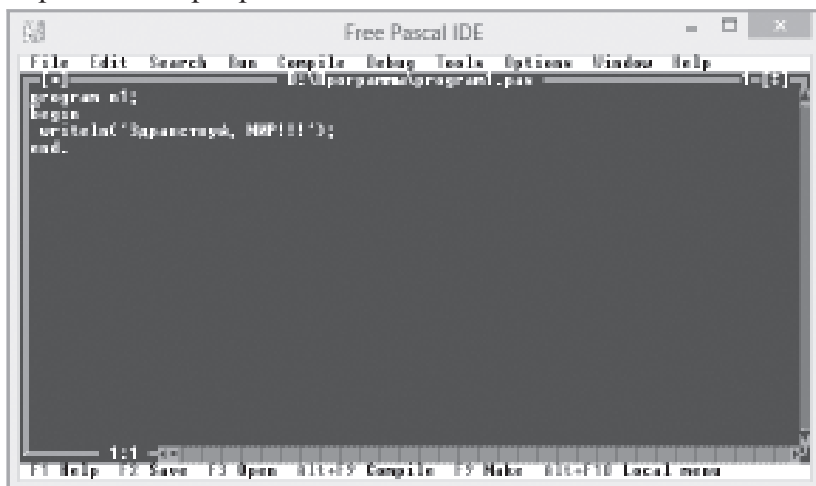


Рис. 11

13. Для того чтобы выйти из *Free Pascal*, выбираем в меню *File* опцию *Exit*.

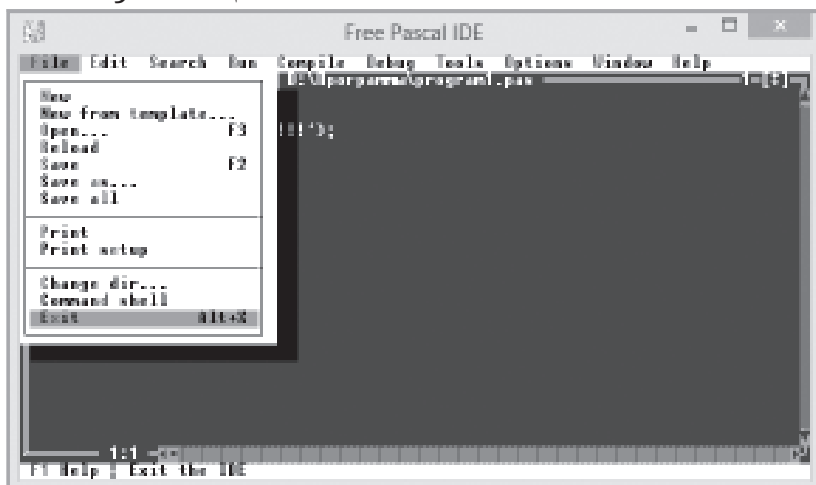


Рис. 12

Примечание: информацию по установке и настройке среды Free Pascal можно получить на сайте разработчиков [8], а также пользуясь литературой [7] и интернетом [9].

■ *Линейные алгоритмы*

При выполнении линейного алгоритма все действия следуют друг за другом и не повторяются (рис. 13).

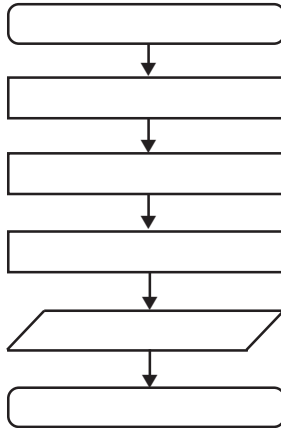


Рис. 13

Действия могут быть реализованы с помощью простейших операторов языка Паскаль.

Оператор присваивания :=

В операторе слева всегда стоит имя переменной, а справа — значение:

$$a:=b;$$

где a — имя переменной, b — значение, выражение, переменная, константа или функция. Типы переменных a и b должны совпадать или быть совместимыми для присваивания. В результате выполнения оператора переменной a присваивается значение b .

Пример:

```
var
  a, b, c: real;
begin
  a:=pi/2;    {pi — число π (3,14)}
  b:=5*ln(a); {ln — натуральный логарифм}
  c:=a+b;
end.
```

Операторы ввода-вывода

Ввод информации

read (x1, x2... xn); или **readln (x1, x2... xn);**

где $x_1, x_2... x_n$ — список вводимых переменных.

Когда в программе встречается оператор *read*, ее действие приостанавливается до тех пор, пока не будут введены исходные данные.

Вводимые переменные отделяются друг от друга пробелом или клавишей Enter.

Вывод информации

write (x1, x2... xn); или **writeln (x1, x2... xn);**

где $x_1, x_2... x_n$ — список выводимых переменных, констант, выражений.

Если используется оператор *writeln*, то после вывода информации курсор перемещается на новую строку.

Примеры программ

1. Печать строки текста

Первый вариант программы:

```
program n1;
begin
  writeln ('Здравствуй, МИР!!!');
end.
```

Программа печатает на экране приветствие и завершает свою работу. Для просмотра результата работы следует одновременно нажать клавиши *Alt* и *F5*. После просмотра нажать любую клавишу.

В т о р о й вариант программы:

```
program n1_1;
uses crt; {модуль процедуры очистки экрана}
begin
  clrscr; {процедура очистки экрана}
  writeln ('Здравствуй, МИР!!!');
  readln; {задержка выполнения программы}
end.
```

Программа выводит на экран приветствие и не завершается, пока не будет нажата любая клавиша.

2. Сложение двух чисел

П е р в ы й вариант программы:

```
program n2;
uses crt;
var
  a:integer; {первое слагаемое}
  b:integer; {второе слагаемое}
  s:integer; {сумма}
begin
  clrscr;
  a:=5;      {присвоение переменной a значения}
  b:=7;      {присвоение переменной b значения}
  s:=a+b;    {вычисление суммы}
  writeln ('Сумма двух чисел = ',s); {вывод суммы}
  readln;
end.
```

Программа выводит на экран:

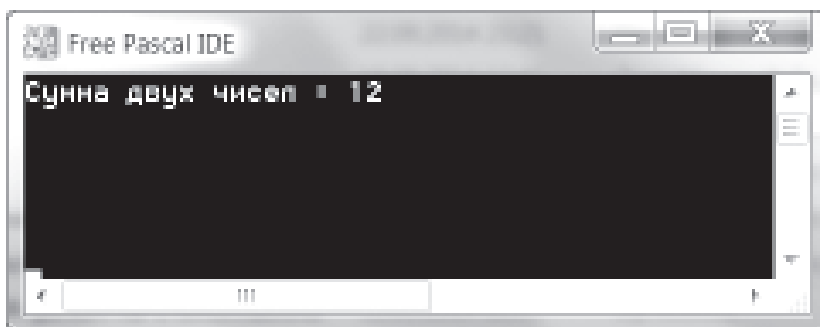


Рис. 14

Второй вариант программы.

Программа запрашивает значения слагаемых у пользователя:

```
program n3;
uses crt;
var
  a, b, s:integer;
begin
  clrscr;
  {Ввод исходных данных}
  writeln ('Введите первое число');
  readln(a);
  writeln ('Введите второе число');
  readln(b);
  {Решение задачи}
  s:=a+b;
  {Вывод результатов}
  writeln ('Сумма двух чисел = ',s);
  readln;
end.
```

Программа выводит на экран:

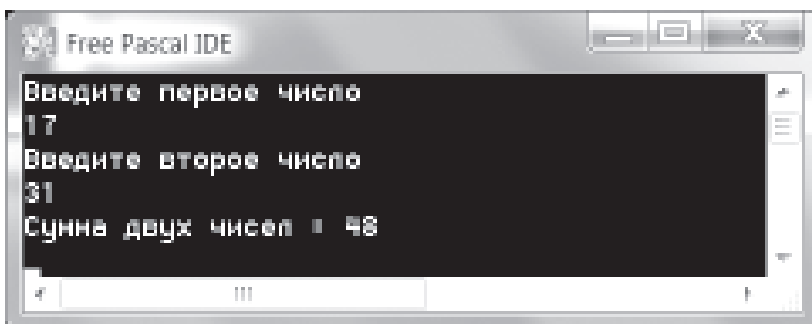


Рис. 15

3. Вычисление среднего арифметического трех целых чисел

```
program n4;  
uses crt;  
var  
    a, b, c:integer;  
    sr:real; {среднее арифметическое}  
begin  
    clrscr;  
    writeln ('Введите три целых числа');  
    readln (a, b, c);  
    sr:=(a+b+c)/3;  
    writeln ('Среднее арифметическое = ',sr:6:2);  
    readln;  
end.
```

Примечание: в данной программе используется форматный вывод результата.

Программа выводит на экран:

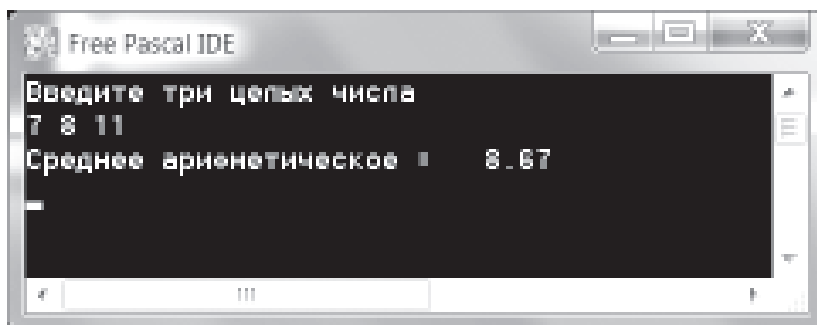


Рис. 16

4. Нахождение суммы цифр двузначного числа

```
program n5;  
uses crt;  
var  
    a:integer; {вводимое число} {исходные данные}  
    s:integer; {сумма цифр}      {требуемый результат}  
    c1,c2:integer; {цифры числа}  
begin  
    clrscr;  
    {Ввод исходных данных}  
    writeln ('Введите целое двузначное число');  
    readln(a);  
    {Решение задачи}  
    c1:=a div 10;      {первая цифра}  
    c2:=a mod 10;     {вторая цифра}  
    s:=c1+c2;         {сумма цифр}  
    {Вывод результата}  
    writeln ('Сумма цифр числа ',a,' равна ',s);  
    readln;  
end.
```


Программа выводит на экран:

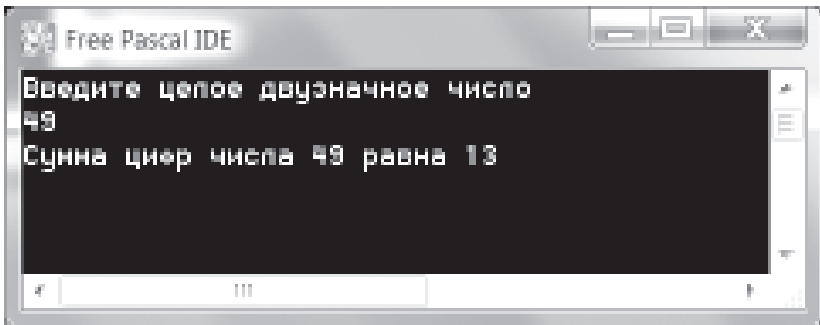


Рис. 17

Упражнения

1. Дана окружность радиуса R . Вычислить ее длину.
2. Дан круг радиуса R . Найти его площадь.
3. Дан угол в градусах. Найти его величину в радианах.
4. Дан угол в радианах. Найти его величину в градусах.
5. Дан шар радиуса R . Найти его объем и площадь поверхности.
6. Вычислить площадь треугольника по трем сторонам. Использовать формулу Герона.
7. Вычислить гипотенузу прямоугольного треугольника по значениям двух катетов. Использовать теорему Пифагора.
8. Вычислить периметр и площадь прямоугольного треугольника по значениям двух катетов.
9. Даны два числа. Найти среднее арифметическое квадратов этих чисел.
10. Даны два действительных числа — x и y . Вычислить их сумму, разность, произведение и частное.

11. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.

12. Момент времени задан в часах, минутах и секундах. Определить количество секунд, прошедших от полуночи до указанного момента.

13. Момент времени задан в секундах от полуночи. Определить время в часах, минутах и секундах.

14. Заданы моменты начала и конца некоторого промежутка времени в часах, минутах и секундах (в пределах одних суток). Найти продолжительность этого промежутка в тех же единицах измерения.

15. Вводится трехзначное число. Вывести на экран число, которое получается из исходного числа путем записи его цифр в обратном порядке.

■ Разветвляющиеся алгоритмы

Разветвляющиеся алгоритмы в языке Паскаль реализуются с помощью условного оператора *if* и оператора выбора *case*.

Условный оператор if... then... else

Условный оператор *if* служит для организации процесса вычислений в зависимости от какого-либо логического условия (рис. 18).

if <условие> *then* <оператор_1> *else* <оператор_2>;

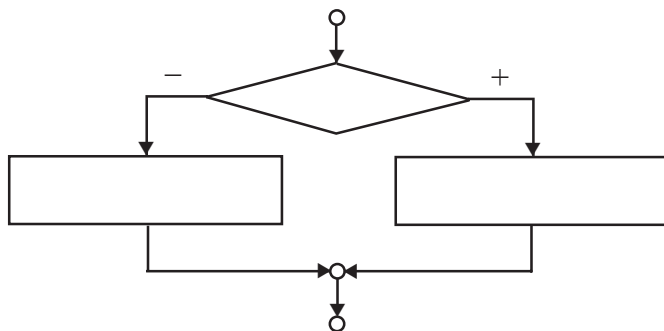


Рис. 18

Фрагмент программы

```
a:=3;  
b:=7;  
if a>b then  
  writeln ('a больше b')
```

```
else  
    writeln ('a меньше b');
```

Примечание: альтернативная ветвь *else* может отсутствовать, если в ней нет необходимости (рис. 19).

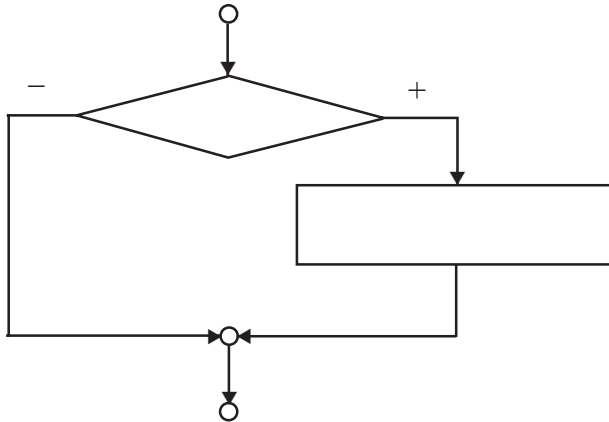


Рис. 19

Чеполная форма оператора if (без ветви else)

Фрагмент программы

```
a:=3;  
b:=7;  
if a>b then  
    writeln ('a больше b');
```

Оператор выбора case

Данный оператор необходим тогда, когда в зависимости от значений какой-либо переменной надо выполнить те или иные операторы (см. рис. 20).

```
case <управляющая переменная> of
  <набор_значений_1>: <оператор_1>;
  <набор_значений_2>: <оператор_2>;
  <...>
  <набор_значений_n>: <оператор_n>
else
  <альтернативный_оператор>
end.
```

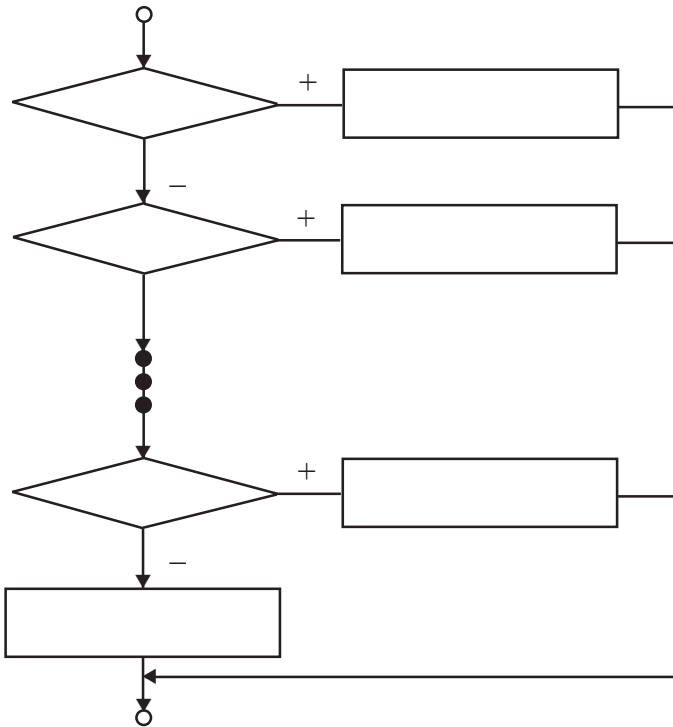
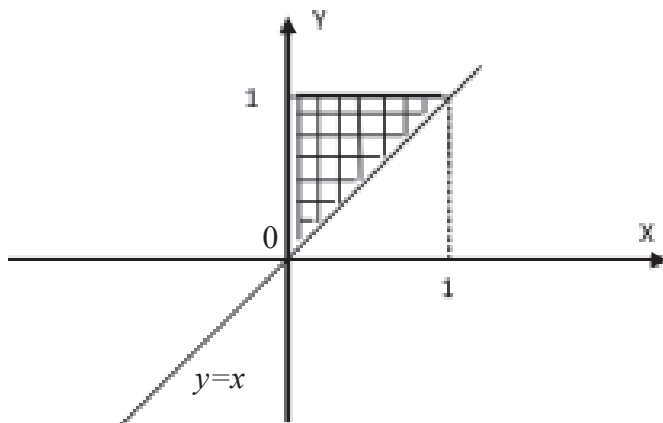


Рис. 20

Примечание: тип управляющей переменной должен быть только перечислимым или целочисленным.

Примеры программы

1. Определение принадлежности точки с координатами x, y заштрихованной части плоскости



```
program n6;
uses crt;
var
  x, y:real;
begin
  clrscr;
  writeln ('Введите координату X');
  readln(x);
  writeln ('Введите координату Y');
  readln(y);
  if (y>x) and (y<1) and (x>0) then
    writeln ('Точка принадлежит заштрихованной области')
  else
    writeln ('Точка не принадлежит заштрихованной области');
  end.
```

Примечание: граница не принадлежит заштрихованной фигуре.

Программа выводит на экран:

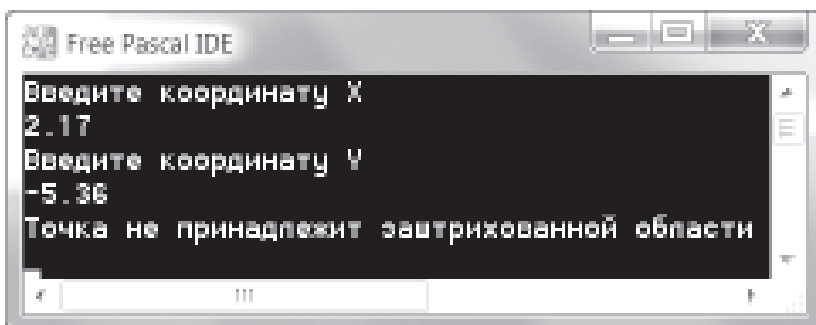


Рис. 22

При введении других исходных данных на мониторе видим:

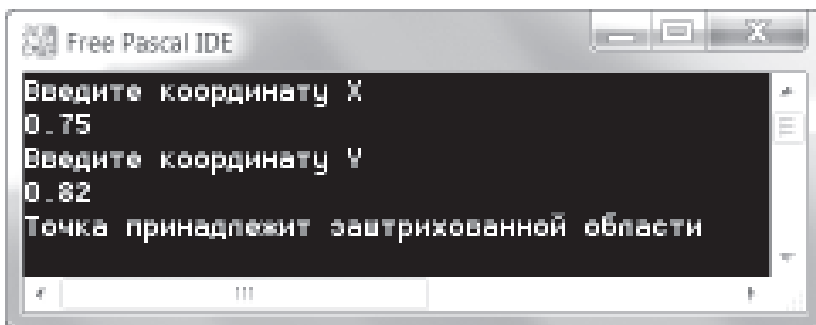


Рис. 23

2. Определение, является ли целое двузначное число четным

```
program n7;  
uses crt;  
var  
    n:integer; {вводимое число}  
begin  
    clrscr;  
    writeln ('Введите целое двузначное число');
```

```

readln(n);
if (n mod 2 = 0) then
    writeln ('Число четное')
else
    writeln ('Число нечетное');
readln;
end.

```

Программа выводит на экран:

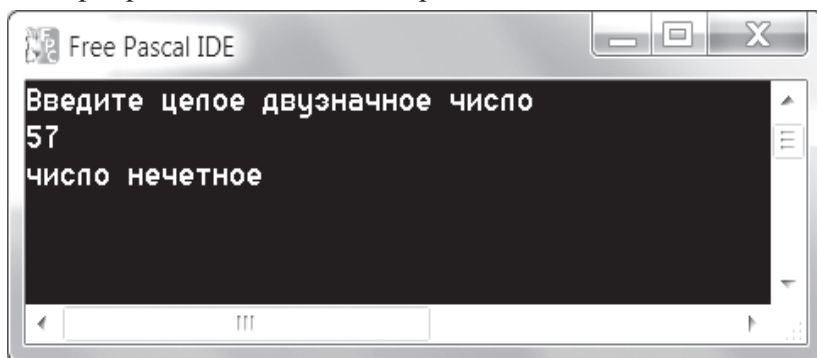


Рис. 24

3. Определение названия дня недели по введенному номеру дня

```

program n8;
uses crt;
var
    m:byte;
begin
    clrscr;
    writeln ('Введите номер дня недели');
    readln(m);
    case m of
        1: writeln ('Понедельник');
        2: writeln ('Вторник');
        3: writeln ('Среда');

```



```

4: writeln ('Четверг');
5: writeln ('Пятница');
6: writeln ('Суббота');
7: writeln ('Воскресенье')
else
  writeln ('Такого дня не существует')
end;
readln;
end.

```

Программа выводит на экран:

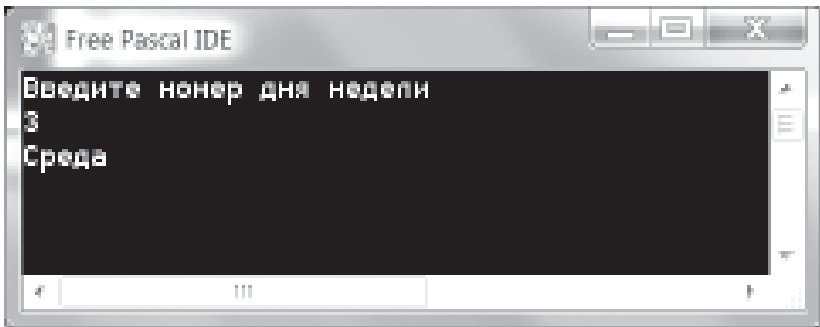


Рис. 25

Упражнения

1. Ввести два числа. Вывести наименьшее из них.
2. Ввести три числа. Вывести наибольшее из них.
3. Дано линейное уравнение $ax+b=0$. Решить это уравнение в соответствии с параметрами a и b .
4. Найти вещественные корни квадратного уравнения $ax^2+bx+c=0$.
5. Дано трехзначное число N . Проверить, будет ли сумма его цифр четным числом.
6. Определить, является ли целое число N четным двузначным числом.
7. Даны длины трех отрезков. Определить, могут ли отрезки быть сторонами треугольника.

8. Определить, является ли треугольник со сторонами a, b, c равносторонним.

9. Определить, является ли треугольник со сторонами a, b, c равнобедренным.

10. Определить, является ли треугольник со сторонами a, b, c разносторонним.

11. Определить, является ли треугольник со сторонами a, b, c прямоугольным.

12. Подсчитать количество отрицательных чисел среди чисел a, b, c .

13. Подсчитать количество положительных чисел среди чисел a, b, c .

14. Подсчитать количество целых чисел среди чисел a, b, c .

15. По результатам сессии, состоящей из трех экзаменов, определить тип студента: отличник, хорошист, троечник, двоечник.

16. По номеру месяца определить время года (зима, весна, лето, осень).

17. Перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y — меньшее.

18. Написать программу нахождения суммы большего и меньшего из трех чисел.

19. Определить, пройдет ли шкаф с размерами $a \times b \times c$ в дверь размером $x \times y$.

20. Определить, можно ли коробку размером $a \times b \times c$ упаковать в посылку размером $u \times v \times w$.

■ Циклические алгоритмы

Оператор цикла с предусловием
while... do (пока... делать)

while <условие> *do* <оператор>;

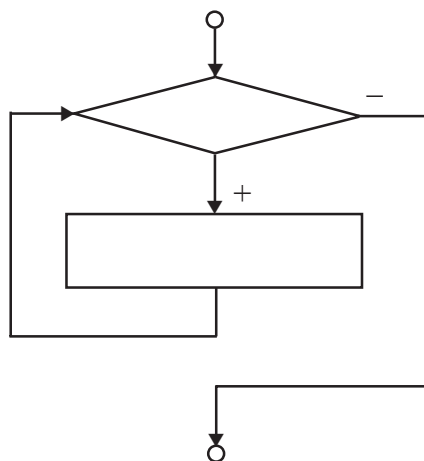


Рис. 26

Оператор после слова *do* называется **телом цикла** и выполняется циклически, пока логическое условие истинно.

Назначение данного оператора: *реализация циклов с предусловием.*

Примечание:

■ Содержимое тела цикла должно обязательно влиять на условие цикла.

■ Условие должно состоять из корректных выражений и значений, определенных до первого выполнения тела цикла.

Оператор цикла с постусловием *repeat... until*

```
repeat  
  <оператор_1>;  
  <оператор_2>;  
  <...>;  
  <оператор_n>;  
until <условие>;
```

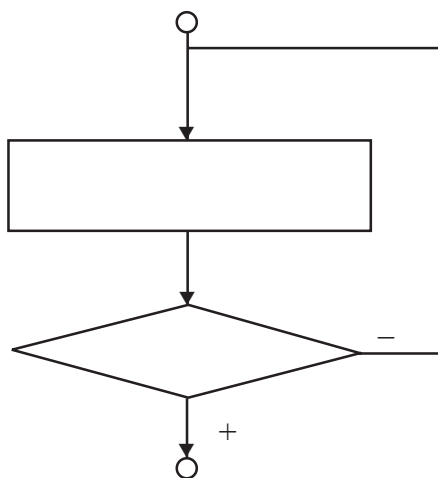


Рис. 27

Назначение данного оператора: *реализация циклов с постусловием.*

Примечание: необходимо выполнять тело цикла, пока не станет истинным условие, то есть цикл выполняется, пока условие ложно.

Оператор цикла с параметром *for... do*

for <параметр_цикла>:=<начальное_значение> *to*
<конечное_значение> *do* <оператор>;
for <параметр_цикла>:=<конечное_значение> *downto*
<начальное_значение> *do* <оператор>;

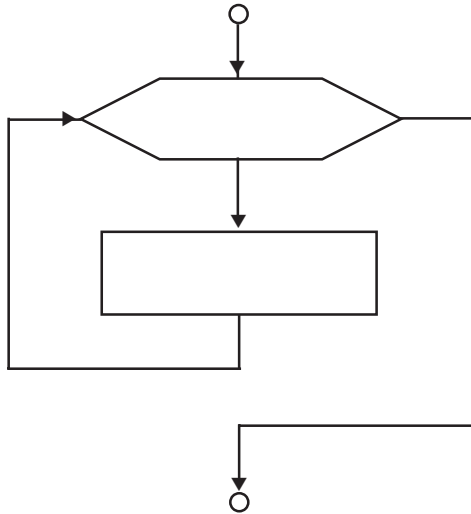


Рис. 28

Примечание:

- ▣ Оператор может быть простым или составным.
- ▣ Параметр цикла только целочисленный или перечислимого типа.
- ▣ Параметр описывается совместно с другими переменными.
- ▣ Шаг цикла всегда постоянный и равен интервалу между двумя ближайшими значениями типа параметра цикла.

Примеры программы

1. Нахождение суммы натуральных чисел от 1 до n.

```
program n9;
uses crt;
var
  n,i:byte;
  s:integer;
begin
  clrscr;
  writeln ('Введите n');
  readln(n);
  s:=0; {обнуление сумматора}
  i:=1;
  while (i<=n) do
    begin
      s:=s+i;
      i:=i+1;
    end;
  writeln ('Ответ ',s);
  readln;
end.
```

Программа выводит на экран:

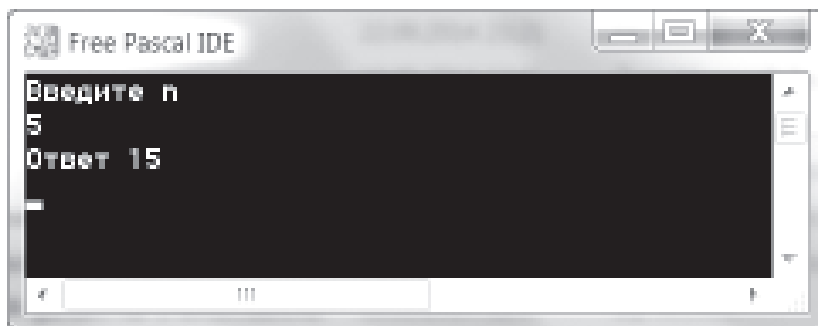


Рис. 29

2. Введение целого числа, выводение его удвоенного значения. Повторение действий, до тех пор пока не будет введен 0

```
program n10;  
uses crt;  
var  
    n:integer;  
begin  
    clrscr;  
    repeat  
        writeln ('Введите целое число');  
        readln (n);  
        writeln ('Ответ ',n*2);  
    until n=0;  
    readln;  
end.
```

Программа выводит на экран:

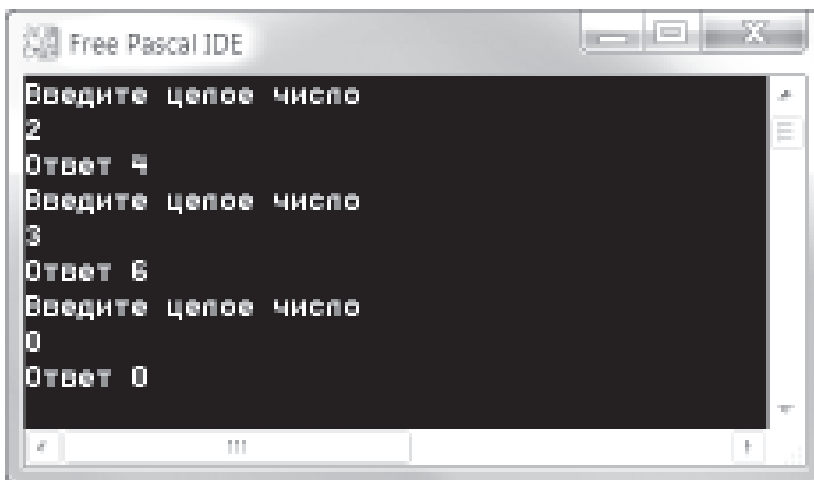


Рис. 30

3. Цикл `for`

```
program n11;
uses crt;
var
  i: integer;
  c: char;
begin
  clrscr;
  writeln ('Печать чисел от 1 до 10');
  for i:=1 to 10 do {Шаг цикла равен 1}
    write(i, ' '); {Печать чисел от 1 до 10}
  writeln;
  writeln ('Печать чисел от 10 до —10');
  for i:=10 downto —10 do {шаг цикла равен —1}
    write(i, ' '); {Печать чисел от 10 до —10}
  writeln;
  writeln ('Печать латинских символов от а до г');
  for c:='a' to 'r' do {Параметр цикла изменяет-
    ся от а до г в алфавитном порядке}
    write(c, ' '); {Печать символов от а до г}
  readln;
end.
```

Программа выводит на экран:

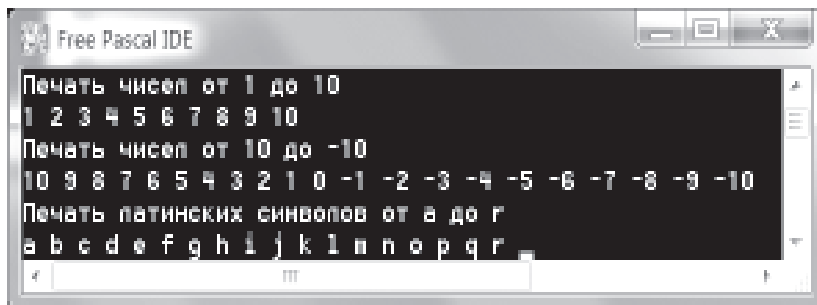


Рис. 31

4. Вычисление факториала числа N ($N!=1\cdot 2\cdot 3\cdot \dots\cdot N$)

```
program n12;  
uses crt;  
var  
    n,i:integer;  
    fact:integer;  
begin  
    clrscr;  
    writeln ('Введите целое число');  
    readln(n);  
    fact:=1; {Начальное значение факториала}  
    for i:=1 to n do  
        fact:=fact*i;  
    writeln ('Факториал числа ',n,' равен ',fact);  
    readln;  
end.
```

Программа выводит на экран:

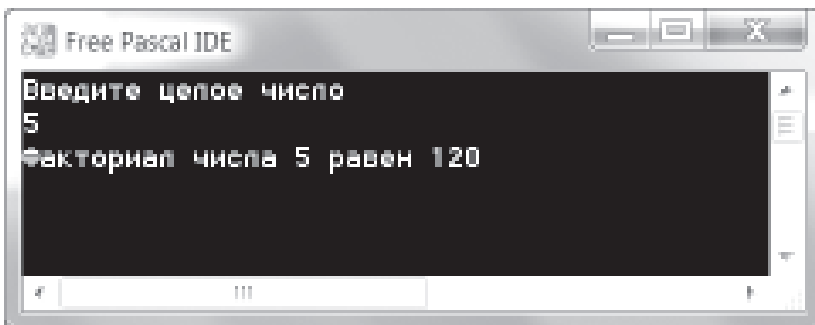


Рис. 32

Упражнения

1. Вывести на экран 50 раз текст «Я хочу учиться».
2. Найти сумму натуральных чисел от 1 до n .
3. Найти сумму чисел от 1 до 100.
4. Найти сумму нечетных чисел от 1 до 100.

5. Найти сумму четных чисел в интервале от A до B (A и B — натуральные числа, $A < B$).

6. Определить количество натуральных трехзначных чисел, сумма цифр которых равна заданному числу N .

7. Среди четырехзначных чисел выбрать те, у которых все четыре цифры различны.

8. Среди двузначных чисел найти те, сумма квадратов цифр которых делится на 13.

9. Найти двузначные числа, обладающие следующими свойствами: если к сумме цифр числа прибавить квадрат этой суммы, то получится снова данное число.

10. Квадраты некоторых трехзначных чисел оканчиваются тремя цифрами, которые как раз и составляют исходные числа. Написать программу поиска таких чисел.

11. Найти количество четных цифр заданного натурального числа N .

12. Определить, содержит ли натуральное число N одинаковые цифры, и если да, то найти их количество.

13. Найти все натуральные числа, не превосходящие заданного N и делящиеся на каждую из своих цифр.

14. Найти сумму четных цифр заданного натурального числа.

15. Определить, сколько раз цифра K встречается в целом числе N .

16. Найти все симметричные натуральные числа (палиндромы) из интервала от K до L .

17. Найти все натуральные числа, не превосходящие заданного N , десятичная запись которых есть строго возрастающая или строго убывающая последовательность цифр.

18. В трехзначном числе зачеркнули первую цифру слева, образовавшееся двузначное число умножили на 7 и получили исходное число. Найти исходное число.

19. Найти все трехзначные числа, которые при увеличении на 1 делятся на 2, при увеличении на 2 делятся на 3, при увеличении на 3 делятся на 4, а при увеличении на 4 делятся на 5.

20. Переставить цифры данного натурального числа таким образом, чтобы образовалось наименьшее число, записанное этими цифрами.

21. Найти сумму четных делителей натурального числа N .

22. Найти сумму целых чисел из промежутка от 1 до 200, у которых ровно 5 делителей.

23. Дано целое число N . Получить все простые делители этого числа.

24. Определить количество счастливых билетов в рулоне с номерами билетов от 000001 до 999999. (Счастливым считается билет, у которого сумма трех первых цифр равна сумме трех последних цифр.)

Литература

1. *Пильщиков, В. Н.* Сборник упражнений по языку Паскаль : учебное пособие для вузов. — М. : Наука, 1989.

2. *Дагене, В. А.* 100 задач по программированию : книга для учащихся / В. А. Дагене, Г. К. Григас, К. Ф. Аугутис. — М. : Просвещение, 1993.

3. *Алексеев, Е. Р.* Турбо Паскаль 7.0. / Е. Р. Алексеев, О. В. Чеснокова, В. Н. Павлыш. — М. : АСТ : НТ Пресс, 2004.

4. *Громницкий, В. С.* Сборник задач по программированию / В. С. Громницкий, В. А. Гусев, Н. А. Паршин. — Н. Новгород : Издательство ННГУ, 2005.

5. *Культин, Н. Б.* Turbo Pascal в задачах и примерах / Н. Б. Культин. — СПб. : БХВ-Петербург, 2007.

6. *Павловская, Т. А.* Паскаль. Программирование на языке высокого уровня : практикум. — СПб. : Питер, 2006.

7. *Кетков, Ю. Л.* Свободное программное обеспечение: FREE PASCAL для студентов и школьников / Ю. Л. Кетков, А. Ю. Кетков. — СПб. : БХВ-Петербург, 2011.

8. www.freepascal.org.

9. www.freepascal.ru.

Содержание _____

Введение

▣▣▣ 3

Основы работы в среде *Free Pascal*

▣▣▣ 4

Линейные алгоритмы

▣▣▣ 11

Разветвляющиеся алгоритмы

▣▣▣ 19

Циклические алгоритмы

▣▣▣ 27

Литература

▣▣▣ 36

Учебное издание

***ВТЮРИН Максим Юрьевич, ГОЛУНОВА Марина Ивановна,
МОСТИПАН Елена Павловна***

П Р А К Т И К У М
по программированию на языке Паскаль

Часть I

Редактор *И. М. Морева*
Компьютерная верстка *О. В. Кондрашиной*

Оригинал-макет подписан в печать 03.10.2014 г.
Формат 60×84^{1/16}. Бумага офсетная. Гарнитура «Times».
Печать офсетная. Усл.-печ. л. 2,33. Тираж 100 экз. Заказ 2193.
ГБОУ ДПО «Нижегородский институт развития образования»
603122, Н. Новгород, ул. Ванеева, 203.
www.niro.nnov.ru

Отпечатано в издательском центре учебной
и учебно-методической литературы ГБОУ ДПО НИРО

