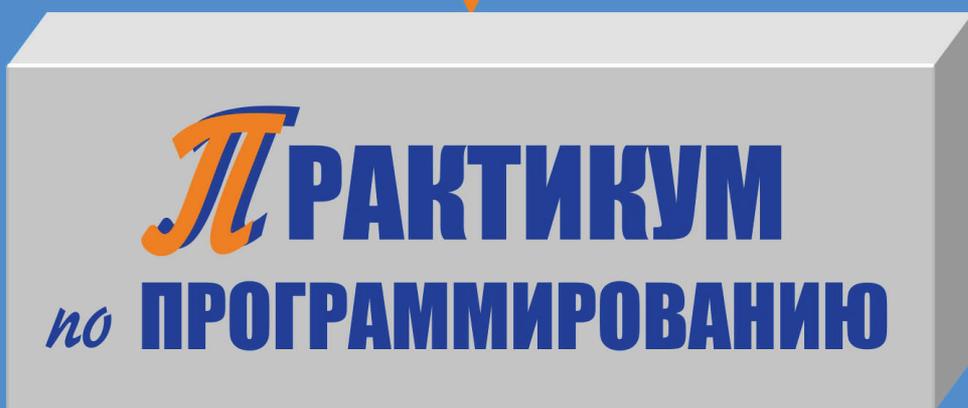


*И. Е. Белоцерковская
М. Ю. Втюрин
Э. В. Ефимова*



Государственное бюджетное образовательное учреждение
дополнительного профессионального образования
«НИЖЕГОРОДСКИЙ ИНСТИТУТ РАЗВИТИЯ ОБРАЗОВАНИЯ»
Кафедра теории и методики обучения информатике

**И. Е. БЕЛОЦЕРКОВСКАЯ, М. Ю. ВТЮРИН,
Э. В. ЕФИМОВА**

ПРАКТИКУМ
по ПРОГРАММИРОВАНИЮ

Нижний Новгород
Нижегородский институт развития образования
2020

УДК 62(075)
ББК 3я721
Б43

Авторы

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

И. Е. Белоцерковская, канд. физ.-мат. наук, доцент кафедры теории и методики обучения информатике ГБОУ ДПО НИРО;

Э. В. Ефимова, канд. пед. наук, доцент кафедры теории и методики обучения информатике ГБОУ ДПО НИРО

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

Белоцерковская, И. Е.

Б43 Практикум по программированию / И. Е. Белоцерковская, М. Ю. Втюрин, Э. В. Ефимова. — Н. Новгород : Нижегородский институт развития образования, 2020. — 77 с.

ISBN 978-5-7565-0862-8

Предлагаемое пособие предназначено для освоения основных алгоритмических структур языка программирования Паскаль, алгоритмов обработки массивов, работы со строковым типом данных. Алгоритмы представлены в виде фрагментов программ. Пособие содержит примеры использования основных алгоритмических структур языка Паскаль, алгоритмов обработки массивов, процедур и функций для работы со строками, а также упражнения для самостоятельной работы.

Издание адресовано слушателям курсов повышения квалификации и призвано оказать помощь учителям информатики в процессе изучения основ алгоритмизации и программирования с помощью языка Паскаль.

**УДК 62(075)
ББК 3я721**

© Белоцерковская И. Е., Втюрин М. Ю., Ефимова Э. В., 2020

© ГБОУ ДПО «Нижегородский институт развития образования», 2020

ISBN 978-5-7565-0862-8

ВВЕДЕНИЕ

П

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

Первая глава знакомит с основами работы в среде *Pascal ABC*, рассмотрены основные элементы интерфейса среды программирования.

Главы со второй по четвертую непосредственно посвящены вопросам структурного программирования. Рассмотрены линейные алгоритмы, алгоритмы ветвления и циклические алгоритмы.

В пятой главе пособия дано описание массива как структурированного типа данных и приведены программы, реализующие ввод и вывод элементов массива.

В главах с шестой по девятую рассказывается об алгоритмах обработки массивов. Рассмотрены алгоритмы суммирования, организации счетчиков, поиска минимального элемента, сортировки, построения массива без повторов и другие немаловажные алгоритмы.

В главе десятой данного пособия приведено описание строкового типа данных, основные процедуры и функции обработки строковых переменных, а также показаны примеры программ, реализующих их.

В одиннадцатой главе практикума рассматриваются два алгоритма выделения слов заданной строки. Первый алгоритм приведен для случая, когда слова строки разделены одним или несколькими пробелами. Второй алгоритм выделения слов строки показан для случая, когда слова разделены наперед заданным фиксированным набором символов.

Изложение материала ведется по принципу — сначала приводится краткая справка об основных операторах, необходимых для освоения соответствующей темы, описываются исходные данные, затем показывается требуемый результат выполнения алгоритма. Далее приводится алгоритм в виде фрагмента программы на языке программирования Паскаль и после этого рассматривается пример решения конкретной задачи с применением соответствующего алгоритма. В конце каждой главы приведены упражнения для самостоятельной работы.

Приступая к работе, сначала следует изучить программы, приведенные в соответствующем разделе, реализовать их самостоятельно на компьютере, отладить и протестировать с различными исходными данными. После этого рекомендуется самостоятельно приступать к решению задач из упражнений по изучаемой теме.

Пособие предназначено, в первую очередь, для работы на практических занятиях, а также для самостоятельной практической работы. Более полную информацию о языке программирования Паскаль, среде программирования и принципах структурного программирования читатель может получить из источников, представленных в разделе «Литература» данного издания.

С структура программы на языке Паскаль

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

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

Простая программа на языке Паскаль

Программа печатает приветствие на экран:

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

Порядок работы в среде Pascal ABC

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

- Запускаем **Pascal** с рабочего стола , используя ярлык, или из списка программ.

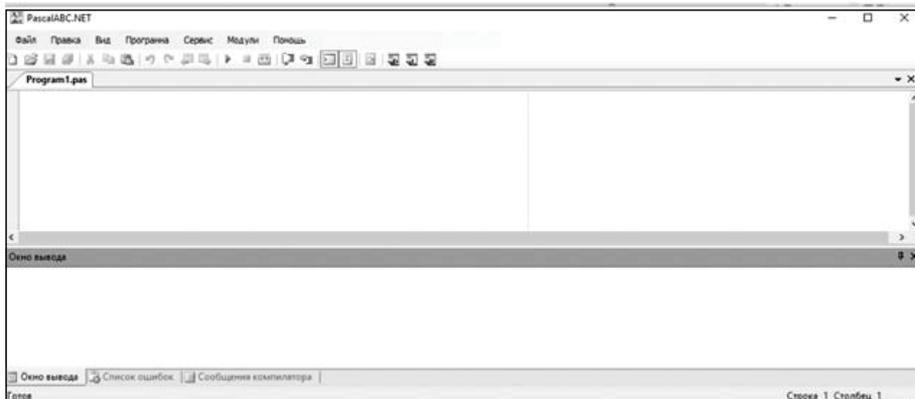


Рис. 1

- В меню **Файл** выбираем опцию **Новый**.

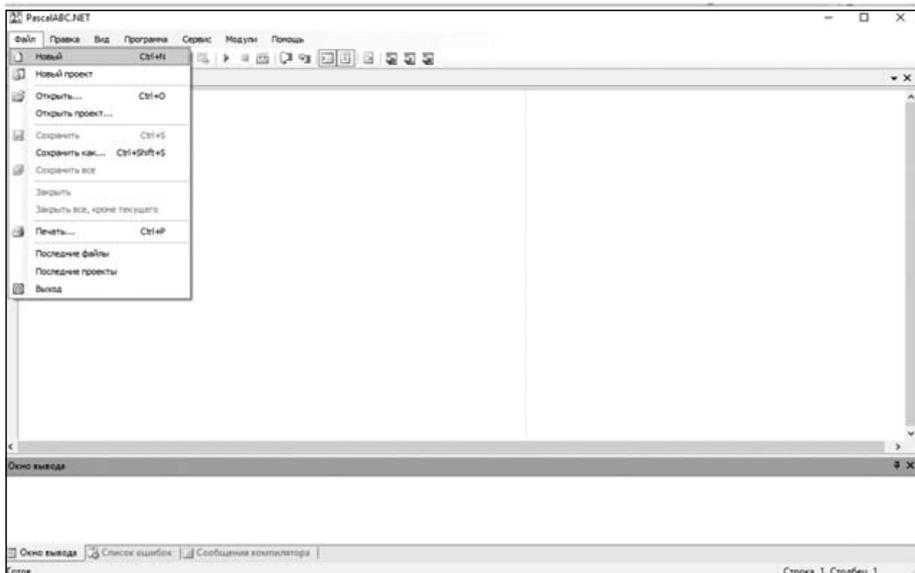


Рис. 2

- Открывается редактор для создания программы.

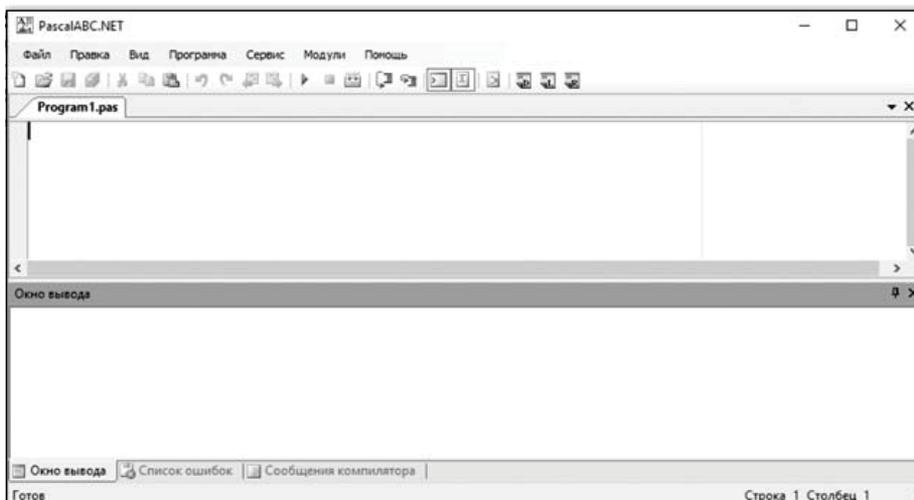


Рис. 3

- В меню **Файл** выбираем опцию **Сохранить**.

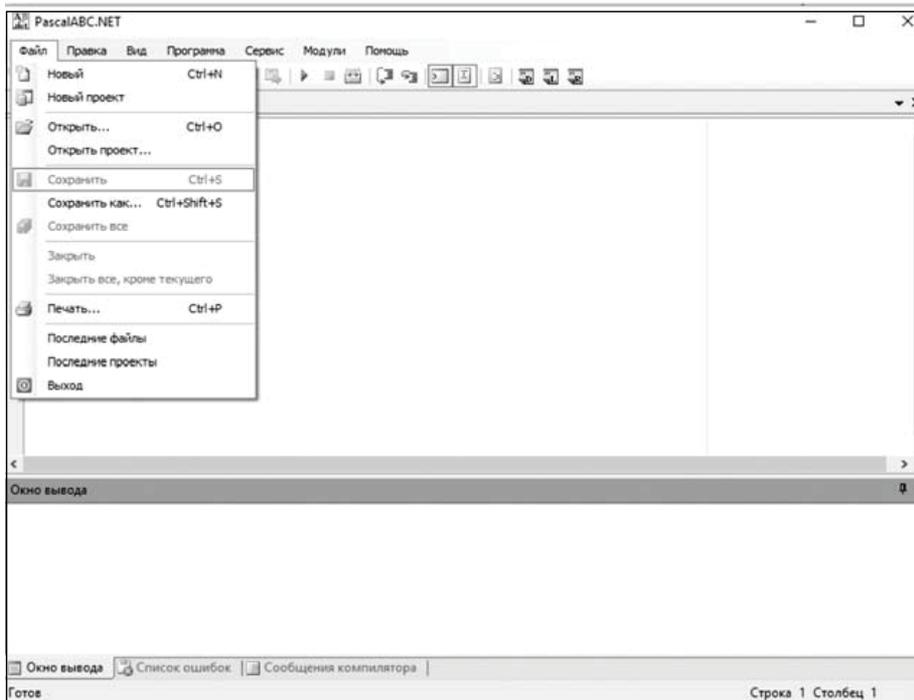


Рис. 4

- Задаем имя нашему файлу с текстом программы.

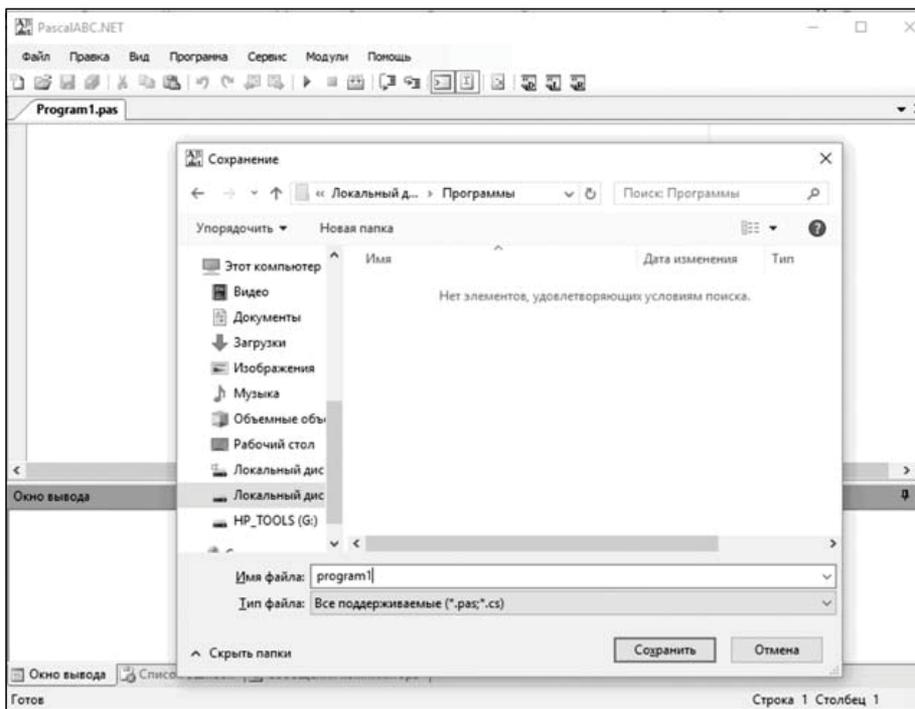


Рис. 5

- Осуществляем набор текста программы, затем в меню **Файл** выбираем опцию **Сохранить**.

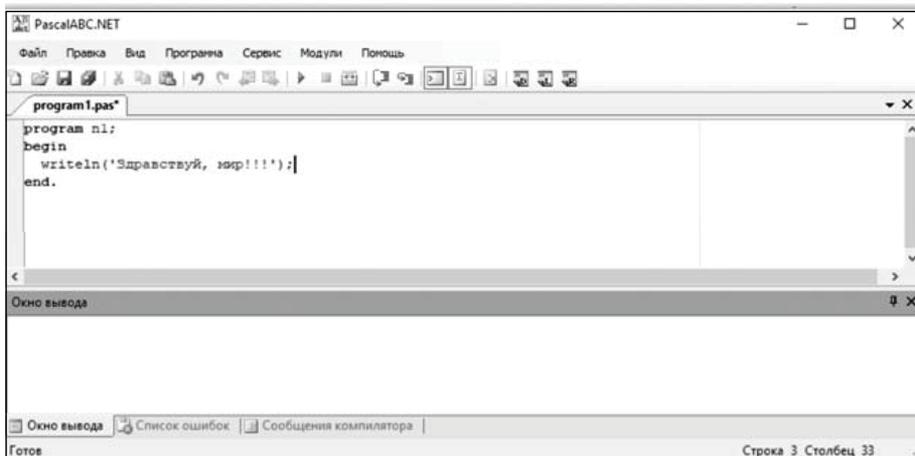


Рис. 6

- В меню **Программа** выбираем опцию **Выполнить**.

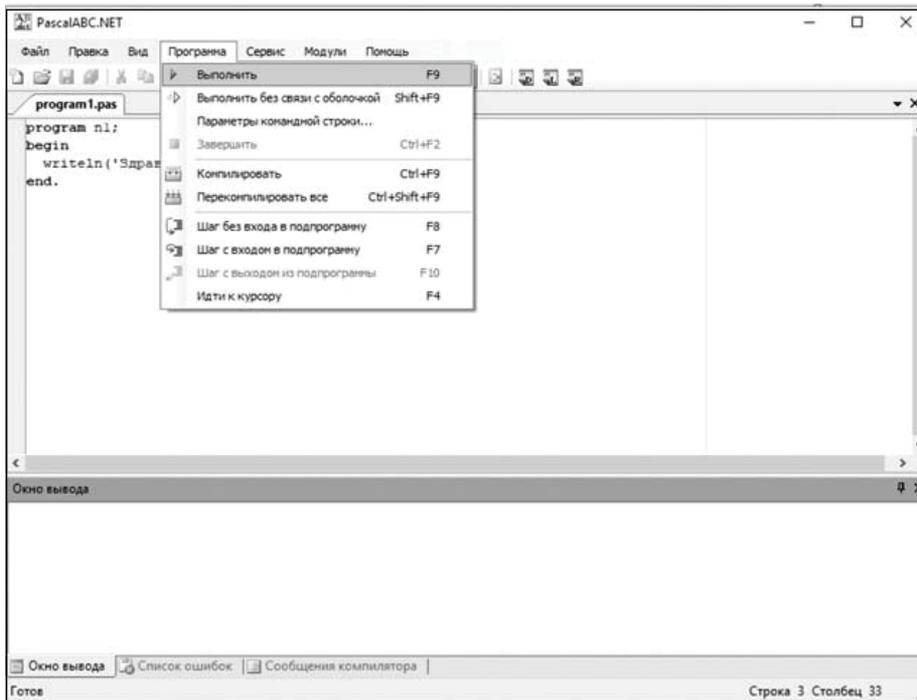


Рис. 7

- Результат программы вы можете увидеть в окне вывода.

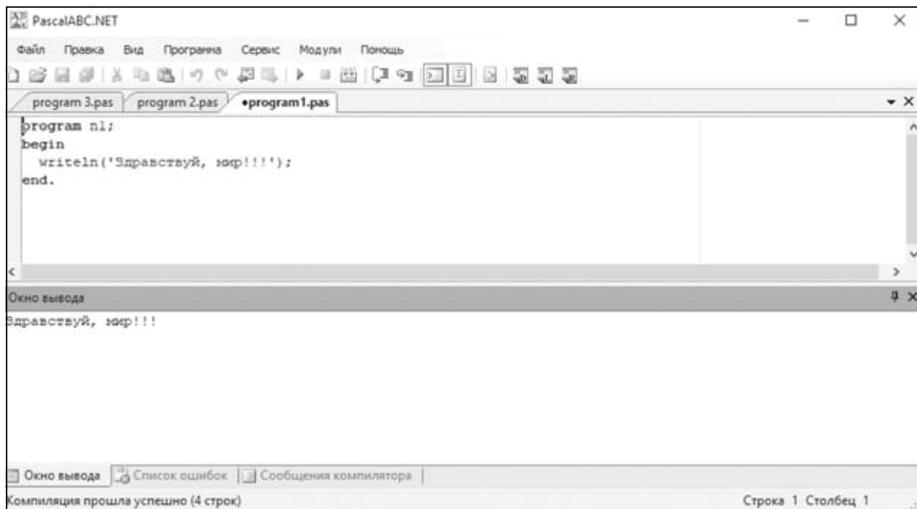


Рис. 8

■ Для просмотра результата выполнения программы без связи с оболочкой нажимаем клавиши **Shift** и **F9** одновременно.

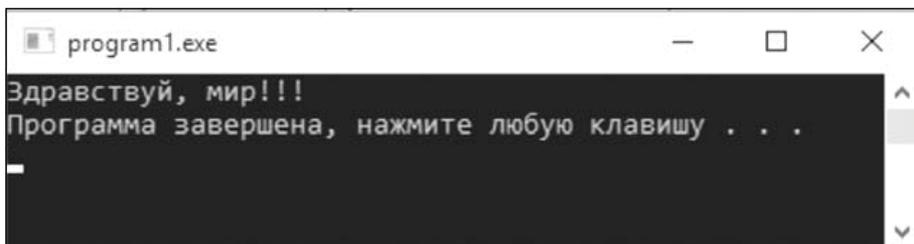


Рис. 9

■ Нажатие любой клавиши возвращает в окно редактора текста программы.

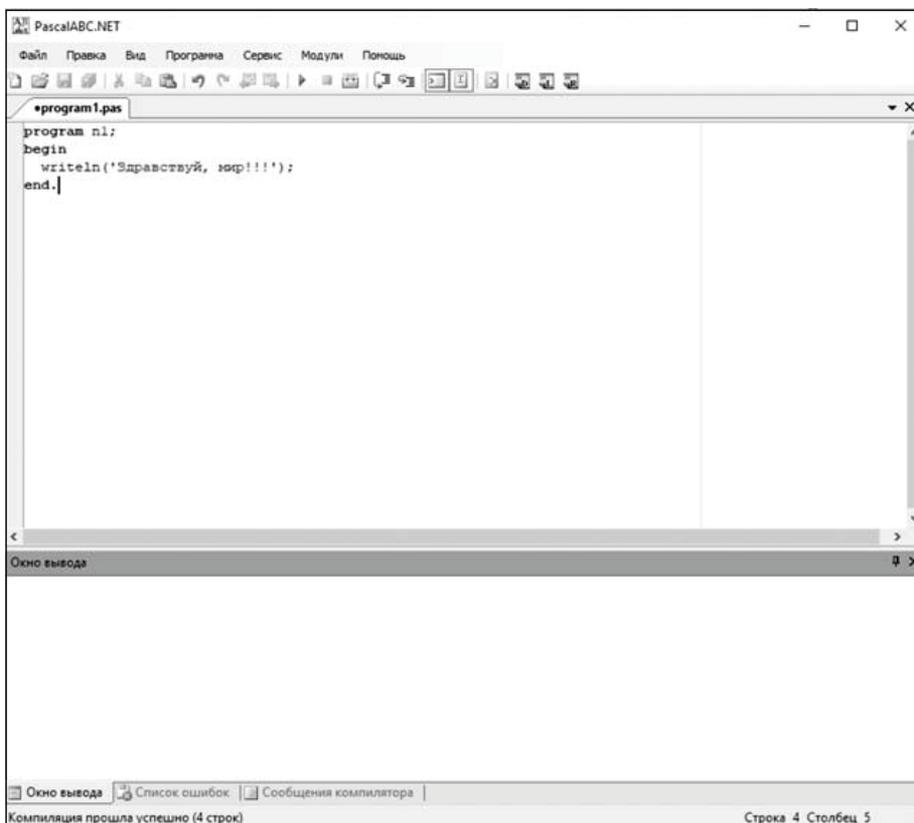


Рис. 10

■ Для выхода из *Pascal ABC* выбираем в меню **Файл** опцию **Выход**.

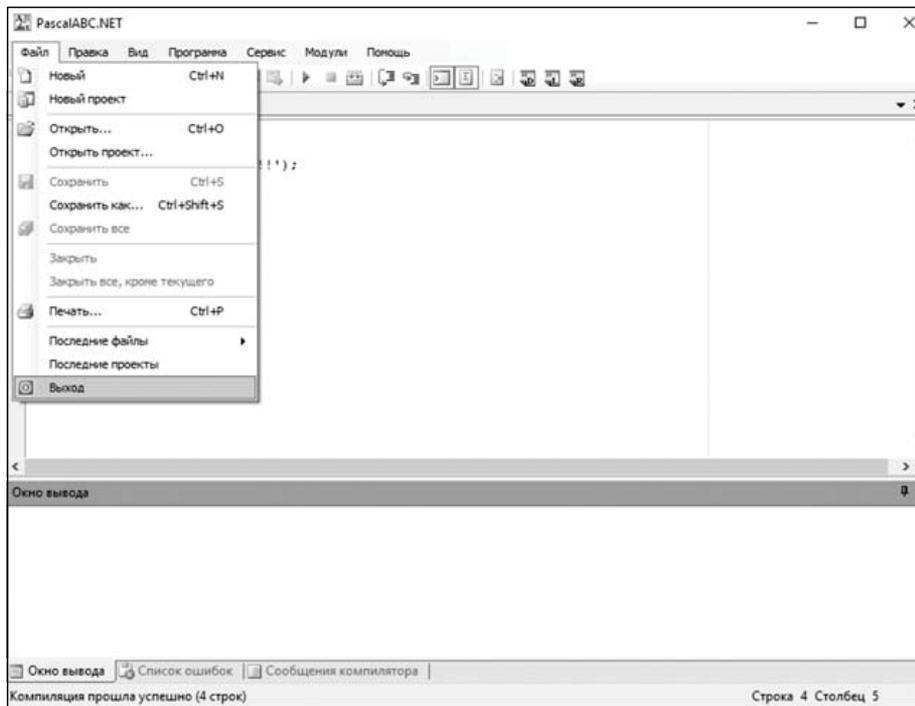


Рис. 11

Все действия следуют друг за другом и не повторяются (рис. 12).

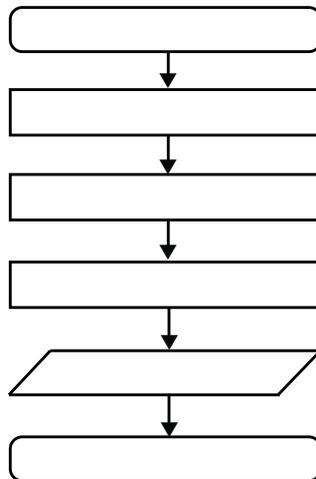


Рис. 12

Действия могут быть реализованы с помощью простейших операторов языка Паскаль.

Оператор присваивания :=

В операторе слева всегда стоит имя переменной, а справа значение.

Пример:

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. Результат: переменной a присваивается значение b.

Пример: var
a, b, c: real;
begin
a:=pi/2;
b:=5*ln(a);
c:=a+b;
end.

Операторы ввода-вывода

Ввод информации

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

где x1, x2, ..., xn — 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 **Enter**.

Вывод информации

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

где x1, x2, ..., xn — список выводимых переменных, констант, выражений.

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

Примеры программ

❶ Печать строки текста

- *Первый вариант программы*

```
program n1;  
begin
```

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

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

■ *Второй 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

■ *Первый 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:

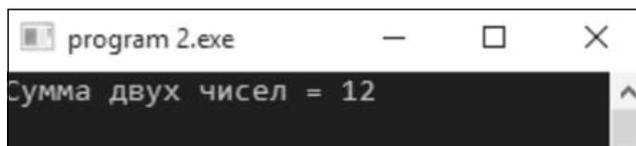


Рис. 13

■ *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:

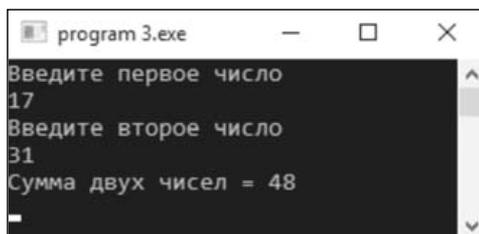


Рис. 14

3 Вычислить среднее арифметическое 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.
```

Программа выводит на экран:

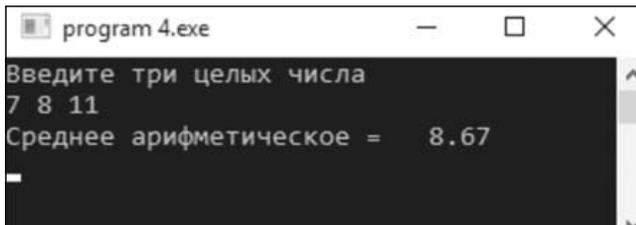


Рис. 15

4 Найти сумму 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}    {вспомогательные переменные}
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:

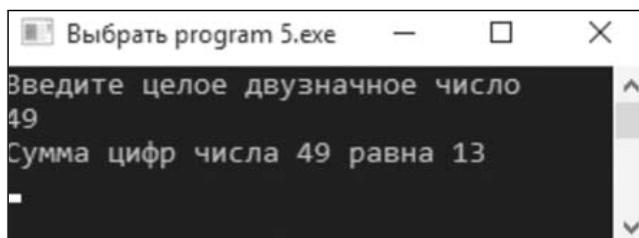


Рис. 16

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 этого 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.

16. Вычислить значение функции Y при заданном значении аргумента x :

а) $Y = \sqrt{x^2 + x}$ при $x = 0,12$;

б) $Y = \frac{\sqrt[3]{x}}{(x-2)^2} + \sin x$ при $x = 2,27$;

в) $Y = \cos(3,56(x^3 - 4x))$ при $x = -0,21$;

г) $Y = \arccos(e^{x^5})$ при $x = 0,23$;

д) $Y = \frac{5(\cos x)^3 - 1}{\sqrt[3]{x} + 2,5}$ при $x = 0,41$;

е) $Y = |\sin(x + x^2)|^4$ при $x = 0,152$.

17. Определить время падения камня на поверхность земли с высоты h .

18. Даны два числа a , b . Найти их среднее арифметическое.

19. Найти произведение значений двух переменных и вывести результат на экран.

20. Найти длину окружности p заданного радиуса r ($p = 2\pi r$).

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. 17).

If (условие) **then** <оператор_1> **else** <оператор_2>;

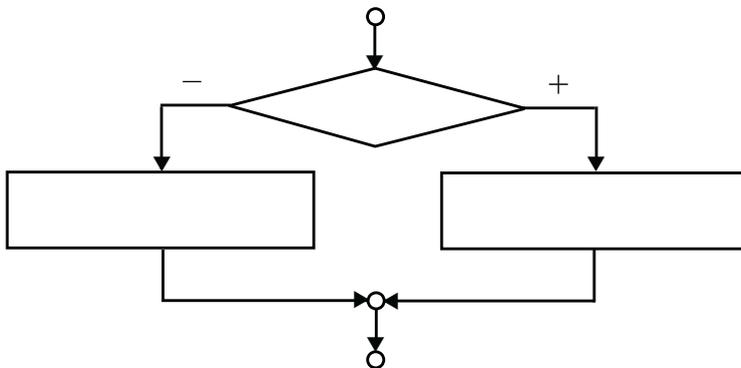


Рис. 17

■ 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. 18).

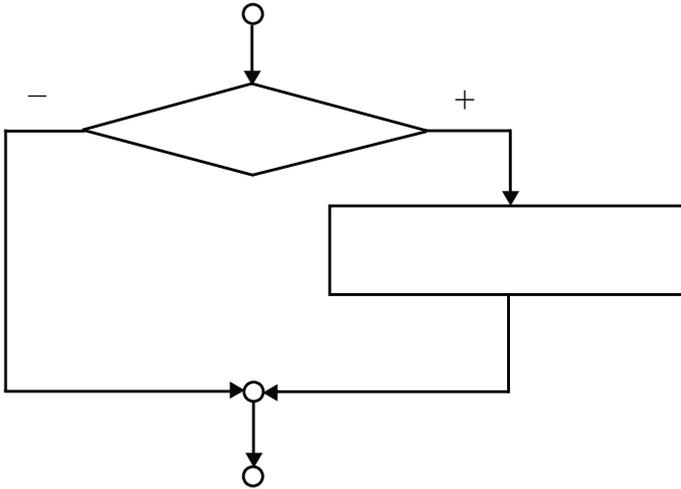


Рис. 18

Неполная форма оператора if (без ветви else)

■ Фрагмент программы

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

Оператор выбора case

Оператор необходим тогда, когда в зависимости от значений какой-либо переменной надо выполнить те или иные операторы (см. рис. 19).

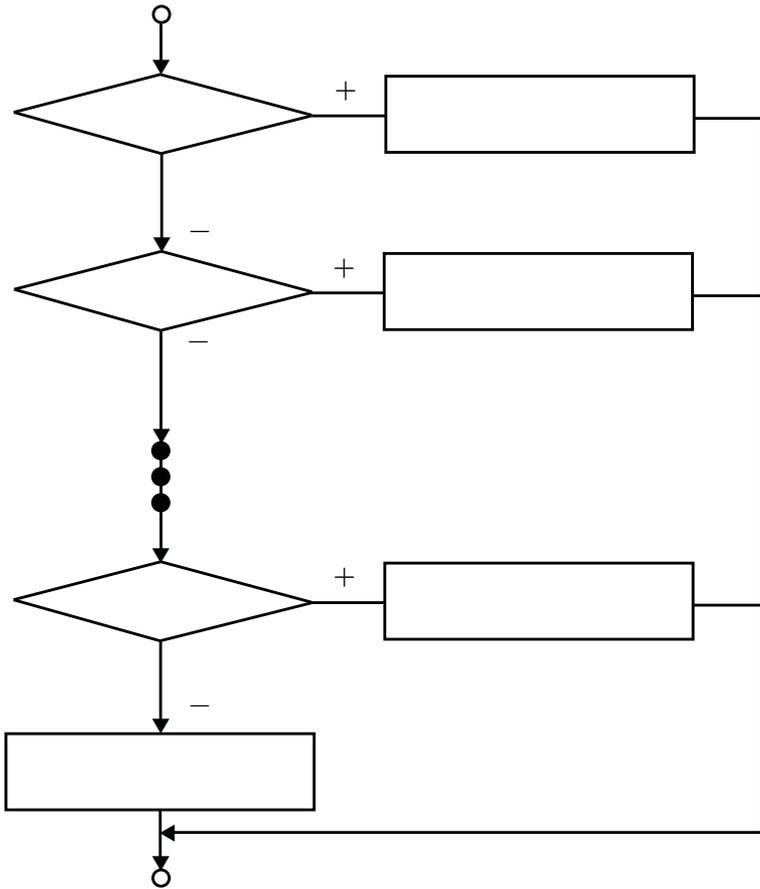


Рис. 19

```

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

Примечание

Тип управляющей переменной должен быть только перечислимым диапазоном или целочисленным.

Примеры программ

- 1 Определить принадлежность точки с координатами x, y заштрихованной части плоскости (рис. 20) (граница принадлежит заштрихованной фигуре)

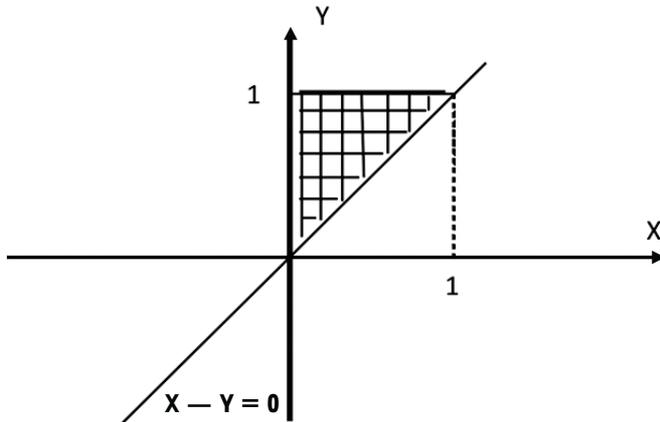
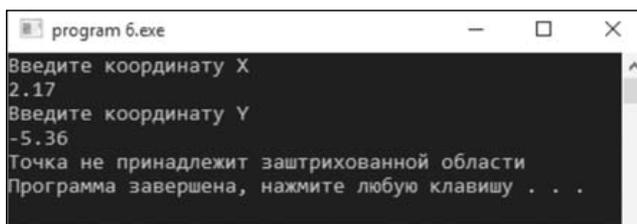


Рис. 20

```
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 об-  
ласти')  
  else  
    writeln ('Точка не принадлежит заштрихованной  
области');  
end.
```

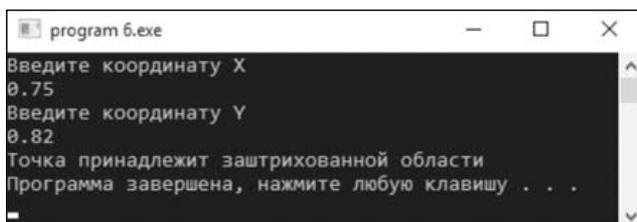
Программа выводит на экран:



```
program 6.exe
Введите координату X
2.17
Введите координату Y
-5.36
Точка не принадлежит заштрихованной области
Программа завершена, нажмите любую клавишу . . .
```

Рис. 21

При других исходных данных:



```
program 6.exe
Введите координату X
0.75
Введите координату Y
0.82
Точка принадлежит заштрихованной области
Программа завершена, нажмите любую клавишу . . .
```

Рис. 22

2 Определить, 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:

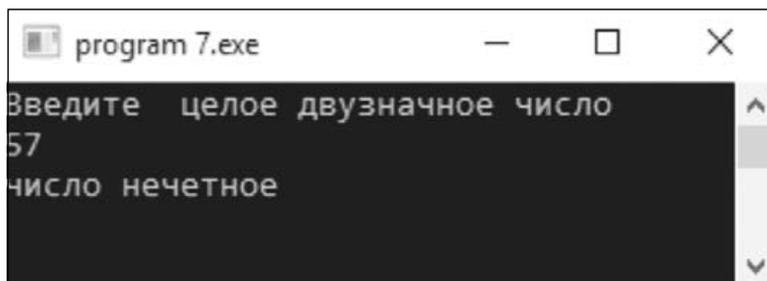


Рис. 23

3 Определить название 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:

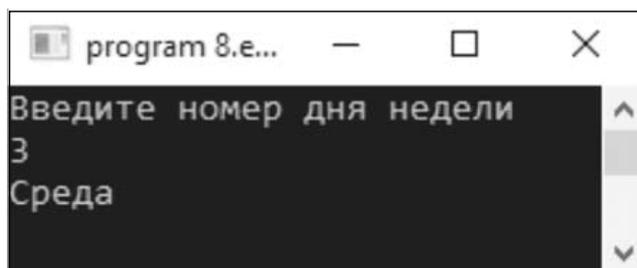


Рис. 24

○
Упражнения

1. Ввести два числа. Вывести наименьшее из них.
2. Ввести три числа. Вывести наибольшее из них.
3. Дано линейное уравнение $ax + b = 0$. Определить решение этого уравнения в зависимости от параметров 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 размерами $a \times b \times c$ в дверь размером $x \times y$.

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

21. Вычислить значение функции Z при x, y , считываемых с клавиатуры:

$$\text{а) } Z = \begin{cases} \tan\left(\sqrt[3]{xy^2} - \frac{1}{\sqrt[3]{x}}\right), & \text{если } \frac{x}{y^2} < 1; \\ \sin(xy - x^2 - y^2), & \text{если } \frac{x}{y^2} \geq 1; \end{cases}$$

$$\text{б) } Z = \begin{cases} (1 - \sin(x^2 + y^3)), & \text{если } xy^2 < 2; \\ \sqrt{x^2y + 8}, & \text{если } xy^2 \geq 2. \end{cases}$$

22. Вычислить значение функции Z при x, y , считываемых с клавиатуры:

$$\text{а) } Z = \min\left\{\sqrt{\sin\frac{y}{x}}; \frac{\sin y}{x}; \sqrt[3]{y \arcsin x}\right\}$$

$$\text{б) } Z = \max\{|\sin x|^y; \sqrt[5]{y + x}\}.$$

0 ператор цикла с предусловием **while... do**
(пока... делать)

while <условие> **do** <оператор>;

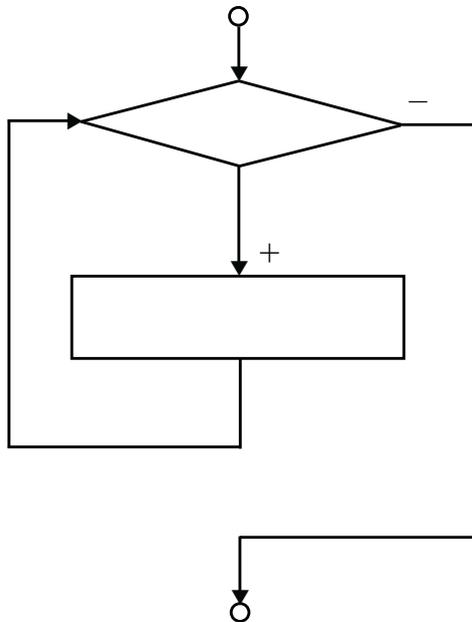


Рис. 25

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

Назначение: 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.

0 оператор цикла с постусловием `repeat... until`

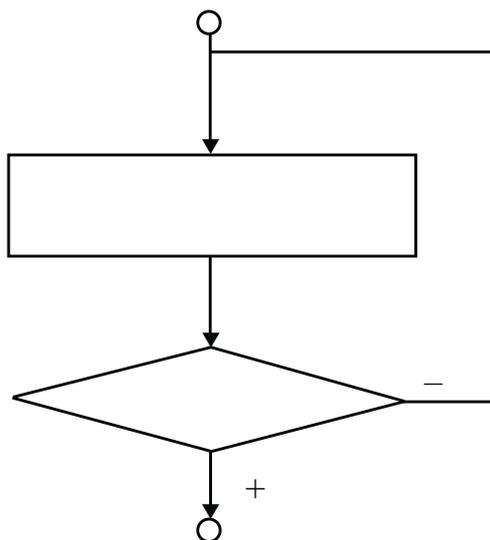


Рис. 26

repeat

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

Назначение

Реализация циклов с постусловием. Выполнять тело цикла, пока не станет истинным условие, то есть пока условие ложно, выполняется цикл.

0 ператор цикла с параметром **for... do**

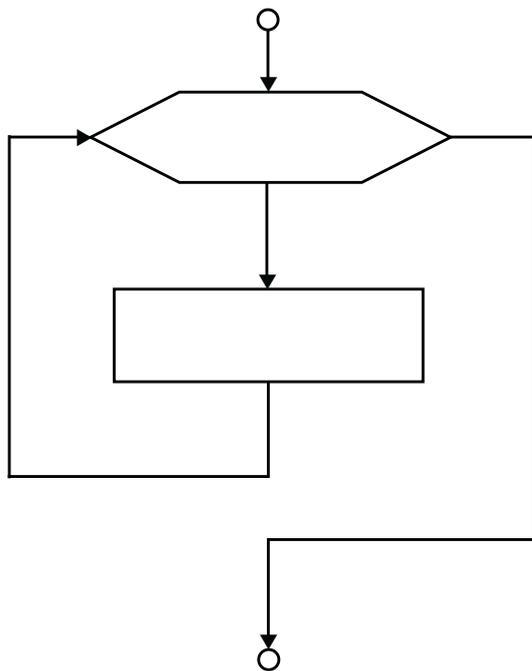


Рис. 27

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

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

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 программ

1 Найти сумму 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:

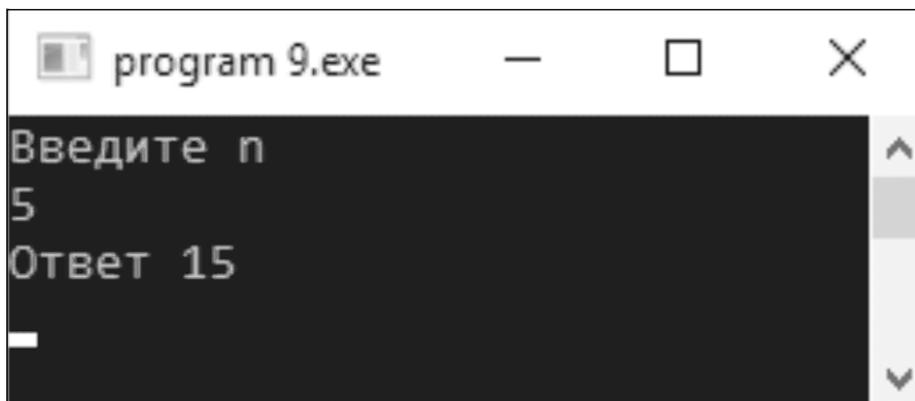


Рис. 28

- ② Ввести целое число, вывести его удвоенное значение. Повторять действия, 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:

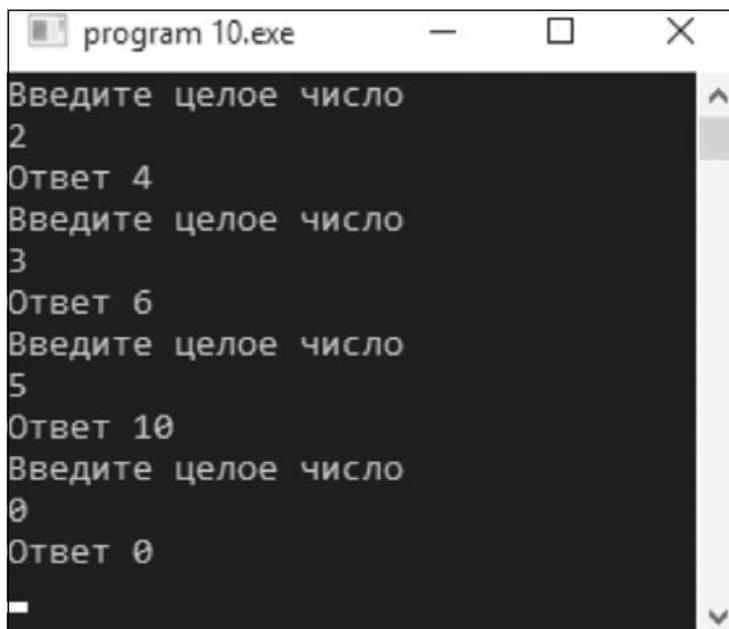


Рис. 29

3 Цикл for

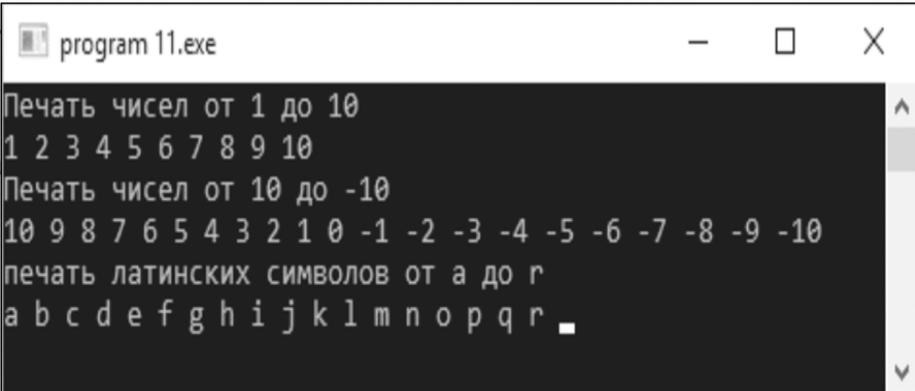
```
program n11;
uses crt;
var
  i: integer;
  c: char;
begin
  clrscr;
  for i:=1 to 10 do           {шаг цикла равен 1}
    write(i);                {печать чисел от 1 до 10}
  for i:=10 downto -10 do   {шаг цикла равен -1}
    write(i);                {печать чисел от 10 до
                              -10}

  for c:='a' to 'r' do      {параметр цикла изменя-
                              ется от а до г в алфавит-
                              ном порядке}

    write(c);                {печать латинских симво-
                              лов от а до г}

  readln;
end.
```

Программа выводит на экран:



```
program 11.exe
Печать чисел от 1 до 10
1 2 3 4 5 6 7 8 9 10
Печать чисел от 10 до -10
10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
печать латинских символов от а до г
a b c d e f g h i j k l m n o p q r .
```

Рис. 30

④ Вычислить факториал числа N ($N!=1*2*3*...*N$)

```
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:

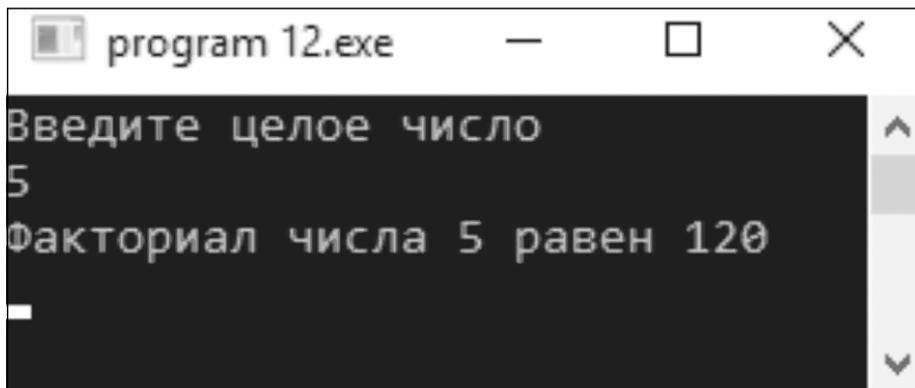


Рис. 31



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. Для данного натурального числа N определить: содержит ли число одинаковые цифры и 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 данная 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. 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;
 - при увеличении на 4 делятся на 5.

- 19.** В трехзначном числе зачеркнули первую цифру слева. Когда полученное двузначное число умножили на 7, то получили исходное число. Найти исходное число.
- 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 число счастливых билетов в рулоне с номерами билетов от 000000 до 999999. Счастливым считается билет, у которого сумма первых трех цифр равна сумме последних трех цифр.

Массив — структурированный тип данных, состоящий из фиксированного числа элементов одного типа.

Описание массивов

```
var <переменная> : array [<тип_индекса>] of <тип_переменных>;
```

Примеры программ

Пример:

```
var  
  x, y: array [1..25] of integer;  
  z: array [-2..8] of real;
```

Описание массивов с помощью констант

Пример:

```
const  
  n=10; m=7;  
var  
  a: array [1..n] of byte;  
  b: array [1..m] of real;
```

Ввод-вывод элементов массива (ввод с помощью цикла for)

```
program n1;
var
  i: integer;
  a: array [1..10] of real;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {ВЫВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    writeln (a[i]:6:2);
  readln;
end.
```

Ввод-вывод элементов массива (ввод с помощью цикла while)

```
program n2;
var
  i: integer;
  a: array[1..10] of real;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  i:=1;
  while (i<=10) do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
      i:=i+1;
    end;
```

```
{вывод 10 элементов массива}  
for i:=1 to 10 do  
  writeln (a[i]:6:2);  
  readln;  
end.
```



Упражнения

1. Ввести массив из 5 элементов с помощью цикла **for** и вывести данный массив на экран.
2. Ввести массив из 8 элементов с помощью цикла **while** и вывести данный массив на экран.

А лгоритм нахождения суммы элементов массива■ **Исходные данные**

n — длина исходного массива;
 $a(n)$ — исходный массив чисел.

■ **Требуемый результат**

s — сумма чисел исходного массива.

■ **Фрагмент программы**

```
s:=0;  
for i:=1 to n do  
  s:=s+a[i];
```

Пример:

Задан массив из 10 целых чисел.

Вычислить сумму элементов массива

```
program n3;  
var  
  i, s: integer;  
  a: array [1..10] of integer;  
begin  
  {Ввод 10 элементов массива}  
  for i:=1 to 10 do  
    begin  
      write ('a(', i, ')=');  
      readln (a[i]);
```

```

end;
{подсчет суммы элементов массива}
s:=0;
for i:=1 to 10 do
  s:=s+a[i];
{вывод результата}
writeln ('Сумма элементов = ', s);
readln;
end.

```

А лгоритм нахождения произведения элементов массива

■ Исходные данные

n — длина исходного массива;

a(n) — исходный массив чисел.

■ Требуемый результат

p — произведение чисел исходного массива.

■ Фрагмент программы

```

p:=1;
for i:=1 to n do
  p:=p*a[i];

```

Пример:

Задан массив из 10 целых чисел.

Вычислить произведение элементов массива.

```

program n4;
var
  i: integer;
  p: longint;
  a: array [1..10] of integer;
begin
  {ввод 10 элементов массива}
  for i:=1 to 10 do
    begin
      write ('a(', i, '=');

```

```

        readln (a[i]);
    end;
    {подсчет произведения}
    p:=1;
    for i:=1 to 10 do
        p:=p*a[i];
    {вывод результата}
    writeln ('Произведение элементов = ', p);
    readln;
end.

```

Подсчет количества элементов массива, удовлетворяющих некоторому условию

■ Исходные данные

n — длина исходного массива;
a(n) — исходный массив чисел.

■ Требуемый результат

k — количество элементов массива, удовлетворяющих некоторому условию.

■ Фрагмент программы

```

k:=0;
for i:=1 to n do
    if (a[i] ?) then {проверка условия, например: a[i]>0}
        k:=k+1;

```

Пример:

Задан массив из 10 целых чисел.

Подсчитать количество положительных элементов.

```

program n5;
var
    i, k: integer;
    a: array [1..10] of integer;
begin
    {ввод 10 элементов массива}

```

```

for i:=1 to 10 do
  begin
    write ('a( ', i, ')=');
    readln (a[i]);
  end;
{подсчет положительных элементов}
k:=0;
for i:=1 to 10 do
  if (a[i]>0) then
    k:=k+1;
{вывод результата}
writeln ('Количество положительных элемен-
        тов = ', k);
readln;
end.

```

Суммирование элементов массива, удовлетворяющих некоторому условию

■ Исходные данные

n — длина исходного массива;
a(n) — исходный массив чисел.

■ Требуемый результат

s — сумма чисел исходного массива, удовлетворяющих некоторому условию.

■ Фрагмент программы

```

s:=0;
for i:=1 to n do
  if (a[i] ?) then {проверка условия, напри-
                    мер: (a[i] mod 2=0)}
    s:=s+a[i];

```

Пример:

**Задан массив из 10 целых чисел.
Вычислить сумму четных элементов массива.**

```

program n6;
var
  i, s: integer;
  a: array [1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {ПОДСЧЕТ СУММЫ ЧЕТНЫХ ЭЛЕМЕНТОВ}
  s:=0;
  for i:=1 to 10 do
    if (a[i] mod 2 =0) then {проверка на чет-
      ность}
      s:=s+a[i];
  {ВЫВОД РЕЗУЛЬТАТА}
  writeln ('Сумма четных элементов = ', s);
  readln;
end.

```

○

Упражнения

- 1. Найти сумму элементов массива.
2. Найти произведение элементов массива.
3. Школьный класс писал контрольную по информатике. Подсчитать среднюю оценку по контрольной.
4. Школьный класс писал контрольную по информатике. Подсчитать количество пятерок и количество двоек, полученных за контрольную.
5. Школьный класс писал контрольную по информатике. Подсчитать количество удовлетворительных и количество неудовлетворительных оценок.

6. Известно количество каждого товара и стоимость единицы каждого товара. Определить суммарную стоимость всех товаров.

7. Задан массив целых чисел. Найти сумму четных чисел.

8. Задан массив целых чисел. Найти произведение нечетных чисел.

9. Задан массив целых чисел. Найти сумму элементов массива, кратных трем.

10. Задан массив целых чисел. Подсчитать количество чисел, кратных семи.

11. Задан массив целых чисел. Подсчитать количество трехзначных чисел.

12. Задан массив целых чисел. Подсчитать количество чисел, оканчивающихся на цифру три.

13. Задан массив целых чисел. Подсчитать сумму двузначных чисел, оканчивающихся на цифру девять.

13. Задан массив действительных чисел. Подсчитать количество целых чисел.

14. Задан массив действительных чисел. Подсчитать сумму нецелых чисел.

15. Задан массив действительных чисел. Подсчитать сумму четных чисел.

16. Задан массив действительных чисел. Подсчитать произведение нечетных чисел.

17. Задан массив 10 действительных чисел. Найти максимальный из отрицательных элементов и поменять его местами с последним.

18. Задан массив 10 действительных чисел. Найти сумму отрицательных элементов, количество положительных и произведение ненулевых элементов.

**ПОИСК МИНИМАЛЬНОГО
(МАКСИМАЛЬНОГО) ЭЛЕМЕНТА
МАССИВА**

0 определение минимального (максимального) элемента и его порядкового номера в массиве (среди всех элементов)

■ **Исходные данные**

n — длина исходного массива;

$a(n)$ — исходный массив чисел.

■ **Требуемый результат**

\min — минимальный элемент массива $a(n)$;

nom — номер минимального элемента в массиве $a(n)$.

■ **Фрагмент программы**

```
min:=a[1];
nom:=1;
for i:=2 to n do
  if a[i]<min then
    begin
      min:=a[i];
      nom:=i;
    end;
```

Пример:

Задан массив из 10 целых чисел.

Найти минимальный элемент массива и его номер.

```

program n7;
var
  i, min, nom: integer;
  a: array [1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {ПОИСК МИНИМАЛЬНОГО ЭЛЕМЕНТА}
  min:=a[1];
  nom:=1;
  for i:=2 to 10 do
    if (a[i]<min) then
      begin
        min:=a[i];
        nom:=i;
      end;
  {ВЫВОД РЕЗУЛЬТАТА}
  writeln ('Минимальный элемент = ', min,
          ' Его номер ', nom);
  readln;
end.

```

0 определение минимального (максимального) элемента и его порядкового номера в массиве (с условием отбора)

■ Исходные данные

n — длина исходного массива;

a(n) — исходный массив чисел.

■ **Требуемый результат**

min — минимальный элемент массива a(n);

nom — номер минимального элемента в массиве a(n).

■ **Фрагмент программы**

```
min:=MAXIMUM; {MAXIMUM — величина,
                заведомо большая, чем
                любой из элементов мас-
                сива}
```

```
nom:=0;
```

```
for i:=1 to n do
```

```
  if (a[i] ?) then {условие отбора}
```

```
    if a[i] < min then
```

```
      begin
```

```
        min:=a[i];
```

```
        nom:=i;
```

```
      end;
```

Пример:

Задан массив из 10 целых чисел.

Найти минимальный нечетный элемент массива и его номер.

```
program n8;
```

```
var
```

```
  i, min, nom: integer;
```

```
  a: array[1..10] of integer;
```

```
begin
```

```
  {ввод 10 элементов массива}
```

```
  for i:=1 to 10 do
```

```
    begin
```

```
      write ('a(', i, ')=');
```

```
      readln (a[i]);
```

```
    end;
```

```
  {поиск минимального элемента}
```

```
  min:=32767; {для типа integer}
```

```
  nom:=0;
```

```
  for i:=1 to 10 do
```

```

if (a[i] mod 2<>0) then
  if (a[i]<min) then
    begin
      min:=a[i];
      nom:=i;
    end;
  {Вывод результата}
  if nom<>0 then
    writeln ('Минимальный нечетный элемент = ', min, ' Его номер ', nom)
  else
    writeln ('Таких элементов нет');
  readln;
end.

```

Упражнения

1. Определить максимальный элемент массива и его номер. Если таких элементов в массиве несколько, указать тот, который в массиве встретится первым.
2. Определить максимальный отрицательный элемент массива и его номер. Если таких элементов в массиве несколько, указать тот, который в массиве встретится последним.
3. Определить минимальный положительный элемент массива и его номер. Если таких элементов в массиве несколько, указать тот, который в массиве встретится первым.
4. Поменять местами максимальный отрицательный и минимальный положительный элементы массива. В заданном массиве один максимальный отрицательный элемент и один минимальный положительный элемент.
5. Найти среднее арифметическое элементов массива, не равных ни наименьшему, ни наибольшему элементам данного массива.
6. Вывести на экран элементы массива целых чисел, имеющих максимальное количество делителей.

7. Вывести на экран элементы массива целых чисел, имеющих максимальную сумму цифр.

8. Задан массив действительных чисел. Среди всех нецелых элементов этого массива найти наименьший и определить его номер. Если таких элементов в массиве несколько, указать тот, который в массиве встретится первым.

9. Задан массив 10 действительных чисел. Найти минимальный из элементов, больших 2, и поменять его местами с первым.

10. Задан массив 10 действительных чисел. Найти минимальный из элементов и максимальный из элементов и поменять их местами.

Сортировка элементов массива в порядке возрастания (убывания)

■ Исходные данные

n — длина исходного массива;
 $a(n)$ — исходный массив чисел.

■ Требуемый результат

$a(n)$ — отсортированный массив (по возрастанию или убыванию).

■ Рабочие переменные

r — переменная для перестановки элементов массива.

■ Фрагмент программы

```

for i:=1 to n-1 do
  for j:=i+1 to n do
    if a[i]>a[j] then {по возрастанию}
      begin
        r:=a[i];
        a[i]:=a[j];
        a[j]:=r;
      end;

```

Пример:

Задан массив из 10 целых чисел.

Упорядочить элементы массива по возрастанию.

```

program n9;
var

```

```

i, j, r: integer;
a: array[1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {СОРТИРОВКА}
  for i:=1 to 9 do
    for j:=i+1 to 10 do
      if a[i]>a[j] then
        begin
          r:=a[i];
          a[i]:=a[j];
          a[j]:=r;
        end;
  {ПЕЧАТЬ РЕЗУЛЬТАТА}
  writeln ('После сортировки');
  for i:=1 to 10 do
    writeln (a[i]);
  readln;
end.

```



1. Упорядочить элементы массива в порядке возрастания.
2. Упорядочить элементы массива в порядке убывания.
3. Вывести на экран элементы массива целых чисел в порядке возрастания их числа делителей.
4. Вывести на экран элементы массива целых чисел в порядке убывания их суммы цифр.

5. Дана ведомость зарплаты сотрудников, в которой указаны табельный номер сотрудника и зарплата каждого. Вывести на экран список табельных номеров сотрудников в порядке увеличения их зарплаты.

6. Дан список студентов группы с указанием фамилии, роста и веса каждого студента. Вывести список студентов, упорядоченный по убыванию роста, при равном росте — по убыванию веса. Вместо фамилии можно использовать номер студенческого билета.

7. В деканате составлена ведомость, в которой указаны фамилии студентов, название предметов и количество прогулов по каждому предмету. Вывести на экран фамилии студентов в порядке увеличения их суммарного числа прогулов по всем предметам.

8. Дана таблица стран — участниц Олимпийских игр с указанием для каждой из них количества завоеванных золотых, серебряных и бронзовых медалей. Упорядочить все страны по убыванию количества золотых медалей. В случае равенства золотых медалей — по убыванию серебряных медалей. Если и здесь равенство, то по убыванию бронзовых медалей.

9. Дан массив из 10 элементов. Первые 4 упорядочить по возрастанию, последние 4 — по убыванию.

10. Дан массив из 10 элементов. Упорядочить по возрастанию только четных элементов массива (нечетные элементы остаются на своих местах).

Отбор элементов, удовлетворяющих некоторому условию, в новый массив

■ Исходные данные

n — длина исходного массива;
 $a(n)$ — исходный массив чисел.

■ Требуемый результат

k — количество элементов формируемого массива;
 $b(k)$ — новый формируемый массив.

■ Фрагмент программы

```

k:=0;
for i:=1 to n do
  if (a[i]?) then {условие отбора в новый
                  массив}
    begin
      k:=k+1;
      b[k]:=a[i];
    end;

```

Пример:

Задан массив из 10 целых чисел.

Построить новый массив, в который поместить все элементы исходного массива, оканчивающиеся на цифру 3.

```

program n10;
var

```

```

i, k: integer;
a, b: array [1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {ФОРМИРОВАНИЕ НОВОГО МАССИВА}
  k:=0;
  for i:=1 to 10 do
    if (a[i] mod 10=3) then {отбор по условию}
      begin
        k:=k+1;
        b[k]:=a[i];
      end;
  {ВЫВОД РЕЗУЛЬТАТА}
  if k<>0 then
    for i:=1 to k do
      writeln (b[i])
    else
      writeln ('Массив пуст');
  readln;
end.

```

П оиск заданного элемента в массиве

■ Исходные данные

n — длина исходного массива;

a(n) — исходный массив чисел;

b — элемент, который надо найти в массиве a(n).

■ Требуемый результат

nom — номер элемента b в массиве a(n).



■ **Фрагмент программы**

```
nom:=0;
i:=1;
while (i<=n) and (nom=0) do
if a[i]=b then
  nom:=i
else
  i:=i+1;
```

Пример:

Задан массив из 10 целых чисел.

Определить, под каким номером в массиве находится заданное число.

```
program n11;
var
  i, b, nom: integer;
  a: array [1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, '=');
      readln (a[i]);
    end;
  writeln ('Введите число ');
  readln (b);
  {ПОИСК В МАССИВЕ}
  nom:=0;
  i:=1;
  while (i<=10) and (nom=0) do
    if a[i]=b then
      nom:=i
    else
      i:=i+1;
  {ВЫВОД РЕЗУЛЬТАТА}
  if nom<>0 then
    writeln (b, ' в массиве под номером ',
```

```

nom)
else
writeln (b, ' в массиве нет');
readln;
end.

```

Формирование массива без повторений из массива, элементы в котором могут повторяться

■ Исходные данные

n — длина исходного массива;

$a(n)$ — исходный массив чисел (с повторениями).

■ Требуемый результат

k — длина нового массива без повторений;

$b(k)$ — новый массив без повторений.

■ Рабочие переменные:

nom — номер элемента из массива $a(n)$ в массиве $b(k)$.

■ Фрагмент программы

```

k:=0;
for i:=1 to n do
begin
nom:=0;
j:=1;
while (j<=k) and (nom=0) do
if a[i]=b[j] then
nom:=j
else
j:=j+1;
if nom=0 then
begin
k:=k+1;
b[k]:=a[i];
end;
end;
end;

```

Пример:

**Задан массив из 10 целых чисел с повторениями.
Создать массив без повторений.**

```
program n12;
var
  i, j, k, nom: integer;
  a, b:array [1..10] of integer;
begin
  {ВВОД 10 ЭЛЕМЕНТОВ МАССИВА}
  for i:=1 to 10 do
    begin
      write ('a(', i, ')=');
      readln (a[i]);
    end;
  {СТРОИМ МАССИВ БЕЗ ПОВТОРОВ}
  k:=0;
  for i:=1 to 10 do
    begin
      nom:=0;
      j:=1;
      while (j<=k) and (nom=0) do
        if a[i]=b[j] then
          nom:=j
        else
          j:=j+1;
      if nom=0 then
        begin
          k:=k+1;
          b[k]:=a[i];
        end;
    end;
  {ВЫВОД РЕЗУЛЬТАТОВ}
  writeln ('Массив без повторений');
  for i:=1 to k do
```

```
writeln(b[i]);  
readln;  
end.
```



Упражнения

1. Определить, содержит ли числовой массив заданное число.
2. Дан числовой массив с повторениями (может содержать одинаковые элементы). Сформировать из него массив без повторений.
3. Имеется два массива элементов без повторений. Сформировать третий массив без повторений из элементов, содержащихся хотя бы в одном из исходных массивов.
4. Имеется два массива элементов без повторений. Сформировать третий массив из элементов первого, которых нет во втором.
5. Имеется два массива элементов без повторений. Сформировать третий массив из элементов, которые есть в обоих массивах.
6. Дан массив целых чисел, который может содержать повторяющиеся элементы. Найти число различных элементов массива.
7. Дан массив целых чисел, который может содержать повторяющиеся элементы. Выяснить, сколько чисел входит в массив по одному разу.
8. Дан массив целых чисел, который может содержать повторяющиеся элементы. Выяснить, сколько чисел входит в массив более чем по одному разу.
9. Дан массив целых чисел, который может содержать повторяющиеся элементы. Выяснить, имеется ли в массиве хотя бы одна пара совпадающих чисел.
10. Дан массив целых чисел, который может содержать повторяющиеся элементы. Найти число, повторяющееся в массиве максимальное количество раз.
11. Дан массив целых чисел, который может содержать повторяющиеся элементы. Найти число вхождений каждого элемента массива.
12. Дан массив целых чисел, который может содер-

жать повторяющиеся элементы. Удалить из массива все отрицательные элементы.

13. Дан массив целых чисел, который может содержать повторяющиеся элементы. Удалить из массива все элементы, значения которых совпадают с максимальным.

14. Дан массив целых чисел, который может содержать повторяющиеся элементы. Расположить все элементы массива в обратном порядке.

15. Найти упорядоченную по возрастанию последовательность подряд расположенных элементов массива максимальной длины.

16. Найти симметричную последовательность подряд расположенных элементов массива максимальной длины.

17. Даны два массива целых чисел a и b , каждый из которых не содержит повторяющихся элементов. Построить пересечение массивов a и b .

18. Даны два массива целых чисел a и b , каждый из которых не содержит повторяющихся элементов. Построить объединение массивов a и b .

Строка — последовательность символов. При использовании в выражениях строка заключается в апострофы. Длина строки до 255 символов.

<имя_переменной>: string[n];

n — длина строковой переменной, если *n* не указано, то длина строки 255 символов.

Пример:

```
var
  x: string[7];
  st: string;
```

В языке Паскаль строку можно рассматривать как массив символов и получить доступ к отдельному символу по его номеру в строке. Например, для строки **st** запись **st[i]** означает символ этой строки, расположенный на *i*-й позиции.

Строки, не содержащие ни одного символа, в языке Паскаль обозначаются двумя подряд идущими апострофами.

Пример:

```
st:=''; {пустая строка (аналог "числового нуля")}
```

Строковые величины можно сравнивать между собой с помощью операций отношения **<**, **>**, **<=**, **>=**, **=**, **<>**.

Процедуры и функции для обработки строк

❶ Определение длины L строки S:

Пример:

```
S:='Мама мыла раму';
```

```
L:=Length(S);
```

Результат: переменной L присвоится значение 14.

❷ Выделение подстроки S1 строки S, длиной L (или до конца) начиная с K-того символа

```
S1:=Copy(S,K,L);
```

Пример:

```
S:='Мама мыла раму';
```

```
S1:=Copy(S,5,6);
```

Результат: переменной S1 присвоится значение ' мыла '.

❸ Поиск подстроки S1 в строке S

```
P:=Pos(S1,S);
```

Пример:

```
S:='Мама мыла раму';
```

```
S1:='мыла';
```

```
P:=Pos(S1,S);
```

Результат: переменной P присвоится значение 6.

❹ Удаление фрагмента строки S длиной L начиная с позиции K

```
Delete(S,K,L);
```

Пример:

```
S:='Мама мыла раму';
```

```
Delete(S,6,7);
```

Результат: переменной S присвоится значение 'Мама му'.

П римеры программ

1 Подсчитать количество цифровых символов в заданной строке текста

```
program n1;
var
  S: string; {исходная строка}
  i:integer; {параметр}
  k:integer; {количество цифровых символов}
begin
  {ввод исходных данных}
  writeln ('Исходная строка: ');
  readln(S);
  {решение}
  k:=0;
  for i:=1 to length(S) do
    if (S[i]='0') or (S[i]='1') or (S[i]='2') or
      (S[i]='3') or (S[i]='4') or (S[i]='5') or
      (S[i]='6') or (S[i]='7') or (S[i]='8') or
      (S[i]='9') then
      k:=k+1;
  {вывод результатов}
  writeln ('Результат: ');
  if k>0 then
    writeln ('Количество цифр ',k)
  else
    writeln ('Цифр в строке нет');
end.
```

2 Удалить из заданной строки все запятые

```
program n2;
var
  s1:string; {исходная строка}
  i:integer; {параметр}
  s2:string; {строка — решение}
begin
```

```

{ВВОД ИСХОДНЫХ ДАННЫХ}
writeln ('Исходная строка: ');
readln (s1);
{решение задачи}
{Формирование новой строки без запятых}
s2:=""; {пустая строка}
for i:=1 to length(s1) do
  if s1[i]<>',' then
    s2:=s2+s1[i];
{Вывод результатов}
writeln ('Строка — решение: ',s2);
end.

```

**3 Задан массив слов.
Объединить все слова в единую строку,
разделяя соседние слова пробелом**

```

program n3;
const
  nmax=50; {максимальный размер массива}
var
  n:integer; {количество слов в массиве}
  i:integer; {параметр}
  w:array [1..nmax] of string; {массив слов}
  s: string; {строка слов}
begin
  {ВВОД ИСХОДНЫХ ДАННЫХ}
  writeln ('Количество слов: ');
  readln (n);
  for i:=1 to n do
    begin
      writeln ('Слово: ');
      readln (w[i]);
    end;
  {решение задачи}
  s:=""; {пустая строка}

```

```

for i:=1 to n do
  s:=s+w[i]+' ';
{Вывод результатов}
writeln ('Строка слов: ',s);
end.

```



Упражнения

1. Найти позицию первого появления заданного символа в заданной строке.
2. Найти позицию третьего появления заданного символа в заданной строке.
3. Удалить все символы пробелов в строке.
4. Подсчитать, сколько раз среди символов строки встречается символ *x*.
5. Подсчитать наибольшее число подряд идущих пробелов в строке.
6. Заменить все восклицательные знаки в строке точками.
7. Удалить символы строки, стоящие перед символом ***.
8. Распечатать все символы строки, расположенные между двумя запятыми.
9. Удалить все символы строки, не являющиеся буквами.
10. Исключить все символы строки между круглыми скобками. Рассмотреть случай вложенных скобок.
11. Подсчитать число вхождений в строку группы символов *abc*.
12. Удалить из каждой группы идущих подряд цифр строки нули.
13. Удалить из строки все запятые, предшествующие первой точке.
14. Проверить, является ли строка палиндромом.
15. Преобразовать строку так, чтобы сначала в ней шли все цифры, а затем все буквы исходной строки.
16. Подсчитать частоту встречаемости каждой цифры в строке.

- 17.** Подсчитать наибольшее число букв А, идущих подряд в строке.
- 18.** Все запятые заданной строки окаймить пробелами.
- 19.** Преобразовать строку, удалив из нее каждый символ * и повторив каждый символ, отличный от *.
- 20.** Заменить все разделители «.,-!?:;» пробелами.
- 21.** Найти номер позиции первого появления частицы «не» в строке. Слева и справа от частицы должны быть пробелы.
- 22.** Инвертировать строку (расположить символы строки в обратном порядке).
- 23.** Разделить исходную строку на несколько строк длины К.
- 24.** Заменить все русские буквы в строке их порядковым номером в русском алфавите.
- 25.** Определить, содержит ли строка дату в виде xx.xx.xx.

Разделителем слов является один или несколько пробелов, любое количество пробелов может стоять в начале или в конце строки

■ **Исходные данные:**

Str — строка слов, разделенных пробелами.

■ **Требуемый результат:**

kw — количество слов в строке Str;

w(kw) — массив слов строки Str.

■ **Рабочая переменная:**

p — позиция очередного пробела в строке.

■ **Фрагмент программы**

```
Str:=Str+' '; {добавляем завершающий пробел}
```

```
kw:=0;
```

```
while Str<>"" do {пока строка не пустая}
```

```
begin
```

```
  p:=Pos(' ',Str); {определяем позицию пробела}
```

```
  if p>1 then {если пробел не первый}
```

```
  begin
```

```
    kw:=kw+1;
```

```
    w[kw]:=Copy(Str, 1, p-1); {копируем все до пробела (слово)}
```

```
  end;
```

```

Delete(Str, l, p); {удаляем все до пробела
                  и пробел}
end;

```

П римеры программ

- ❶ **Задана строка, состоящая из слов, между которыми, а также в начале и конце строки может быть любое количество пробелов. Выделить все слова, начинающиеся на гласную латинскую букву, и поместить их в массив**

```

program n4;
const
  nmax=30; {максимальный размер массива}
var
  s:string;           {исходная строка}
  kw:integer;        {количество слов
                    в строке}
  p:integer;         {позиция пробела}
  w:array[1..nmax] of string; {массив слов стро-
                              ки}
  i:integer;         {параметр}
  k:integer;         {количество иско-
                    мых слов}
  wa:array[1..nmax] of string; {слова на гласную
                              букву}
begin
  {ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (s);
  {решение задачи}
  s:=s+' ';
  kw:=0;             {количество слов
                    в строке}
  while s<>" do
  begin
    p:=pos(' ',s); {номер позиции, на которой пер-
                  вый раз встречается пробел}

```

```

if p>1 then
  begin
    kw:=kw+1;
    w[kw]:=copy(s,1,p-1);
  end;
delete(s, 1, p);
end;
{выделяем слова, начинающиеся на глас-
ную латинскую букву}
k:=0;
for i:=1 to kw do
  if (copy(w[i],1,1)='a') or (copy(w[i],1,1)='e')
  or (copy(w[i], 1,1)='y') or (copy(w[i],1,1)='u')
  or (copy(w[i],1,1)='o') or (copy(w[i],1,1)='i')
  then
    begin
      k:=k+1;
      wa[k]:=w[i];
    end;
{Вывод результатов}
writeln ('Результат: ');
if k=0 then
  writeln ('Слов на гласную букву в строке
нет')
else
  for i:=1 to k do
    writeln (wa[i]);
end.

```

- ② Для слов заданной строки, которые содержат заданное ключевое слово, найти самое длинное слово. Между словами, а также в начале и конце строки может быть любое количество пробелов

■ Контрольный пример

Строка: мирный род помирить родина народ миры.

Ключ: мир.



Результат: помирить.

```
program n5;
const
  nmax=40;           {максимальный раз-
                    мер массива}

var
  s:string;         {заданная строка}
  sk:string;       {ключ}
  kw:integer;      {количество слов
                    в строке}
  p:integer;       {позиция пробела}
  w:array[1..nmax] of string; {массив слов стро-
                              ки}
  i:integer;       {параметр}
  lmax:integer;    {длина искомого
                    слова}
  nom:integer;     {номер искомого
                    слова в массиве
                    слов}

begin
  {ввод исходных данных}
  writeln ('Исходная строка: ');
  readln (s);
  writeln ('Ключ: ');
  readln (sk);
  {алгоритм выделения слов строки}
  s:=s+' ';
  kw:=0;
  while s<>'' do
    begin
      p:=pos(' ',s);
      if p>1 then
        begin
          kw:=kw+1;
```

```

        w[kw]:=copy(s, 1, p-1);
    end;
    delete(s, 1, p);
end;
{алгоритм поиска максимального элемента
(условный)}
lmax:=0;
nom:=0;
for i:=1 to kw do
    if pos(sk,w[i])>0 then {если ключ в слове}
        if length(w[i])>lmax then {слово максималь-
            ной длины}
            begin
                lmax:=length(w[i]);
                nom:=i;
            end;
        {вывод результатов}
        writeln ('Результат: ');
        if nom=0 then
            writeln ('В строке нет слова с заданным ключ-
                ком')
        else
            writeln ('Искомое слово: ',w[nom]);
        end.

```

Разделителем слов являются знаки препинания, заданные в виде набора символов

■ Исходные данные:

Str — строка, состоящая из слов, разделенных заданным набором символов-разделителей;

Dlm — строка символов-разделителей (« .,!?-:;»).

■ Требуемый результат:

kw — количество слов в строке Str;

w(kw) — массив слов строки Str;

d(kw) — массив разделителей слов строки Str.

■ Фрагмент программы

```
kw:=0;
while (Str<>") do
begin
  kw:=kw+1;
  w[kw]:=""; {сумматор символов очередно-
              го слова}
  d[kw]:=""; {сумматор разделителей после
              очередного слова}
  while (pos(Str[1],Dlm)=0) and (Str<>") do
  {пока первый символ строки Str – не
  разделитель}
  begin
    w[kw]:=w[kw]+Str[1];
    delete(Str, 1, 1);
  end;
  while (pos(Str[1],Dlm)>0) and (Str<>") do
  begin
    d[kw]:=d[kw]+Str[1];
    delete(Str, 1, 1);
  end;
end;
```

П ример программы

- ❶ Задана строка, состоящая из слов. Разделителями слов являются знаки препинания, заданные в виде набора символов (.,!?-:;), или набор знаков препинания (например, несколько пробелов или несколько точек). Выделить все слова и поместить их в массив слов. Выделить разделители слов и поместить их в массив разделителей

```
program n6;
const
  nmax=30; {максимальный размер массива}
  Dlm='.,!?-:;'; {строка символов-разделителей}
var
```

```

Str:string;           {исходная строка}
kw:integer;          {количество слов в строке}
w:array[1..nmax] of string; {массив слов строки}
d:array[1..nmax] of string; {массив разделителей
                             слов}

i:integer;
begin
  {ВВОД ИСХОДНЫХ ДАННЫХ}
  writeln ('Исходная строка: ');
  readln (Str);
  {решение задачи}
  kw:=0;
  while (Str<>") do
    begin
      kw:=kw+1;
      w[kw]:="; {сумматор символов очередного слова}
      d[kw]:="; {сумматор разделителей после очеред-
                ного слова}
      while (pos(Str[1],Dlm)=0) and (Str<>") do
        {пока первый символ строки Str — не раздели-
          тель}
        begin
          w[kw]:=w[kw]+Str[1];
          delete (Str,1,1);
        end;
      while (pos(Str[1],Dlm)>0) and (Str<>") do
        begin
          d[kw]:=d[kw]+Str[1];
          delete(Str, 1, 1);
        end;
      end;
    {ВЫВОД РЕЗУЛЬТАТА}
    {ВЫВОД МАССИВА СЛОВ}
  for i:=1 to kw do
    writeln(w[i]);

```

```
{Вывод массива разделителей}  
for i:=1 to kw do  
  writeln(d[i]);  
end.
```



Упражнения

1. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Сформировать массив слов строки.
2. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Найти самое длинное слово строки.
3. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Найти самое короткое слово строки.
4. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Вывести на экран слова строки, упорядочив их по алфавиту.
5. Строка состоит из слов, разделенных одним или несколькими пробелами. Слово — последовательность любых символов, не содержащих пробелов. Вывести на экран слова строки, упорядочив их по возрастанию количества символов.
6. Дан словарь в виде списка (массива) слов и строка, слова в которой разделены пробелами. Определить, какие слова из словаря встречаются в строке.
7. Дан словарь в виде списка (массива) слов и строка, слова в которой разделены пробелами. Вывести те слова строки, которых нет в словаре.
8. Слова в строке разделены символами «пробел,-!?:;». Сформировать массив слов строки.
9. Слова в строке разделены разделителями «пробел,-!?:;». Сформировать массив слов текста.
10. Задан текст в виде массива строк. Предложения в тексте завершаются одним из символов «.!?». Сформировать массив предложений.

11. Задан текст в виде массива строк. Слова в строке разделены символами «пробел,.-!?:;». Сформировать частотный словарь текста — список слов текста в алфавитном порядке с указанием частоты встречаемости каждого слова в тексте.

12. Дан текст в виде массива строк. Отформатировать текст на заданную длину строки с выравниванием по левому краю.

13. Дан текст в виде массива строк. Отформатировать текст на заданную длину строки с выравниванием по правому краю.

14. Дан текст в виде массива строк. Абзацы в тексте начинаются со строки, в начале которой не менее пяти пробелов. Распечатать абзацы в порядке убывания строк в абзацах.

15. Дан текст на некотором языке и алфавит этого языка. Распечатать слова текста в алфавитном порядке (по алфавиту заданного языка).

16. Дана строка, состоящая из русских слов, набранных заглавными буквами и разделенных пробелами (одним или несколькими). Найти количество слов, которые содержат хотя бы одну букву «А».

ЛИТЕРАТУРА

1. *Алексеев, Е. Р.* Турбо Паскаль 7.0 / Е. Р. Алексеев, О. В. Чеснокова, В. Н. Павлыш, Л. В. Славинская. — М. : АСТ : НТ-Пресс, 2006. — 314 с. — (Самоучитель).
2. *Громницкий, В. С.* Сборник задач по программированию / В. С. Громницкий, В. А. Гусев, Н. А. Паршин. — Н. Новгород : Изд-во ННГУ, 2005.
3. *Дагене, В. А.* 100 задач по программированию : книга для учащихся / В. А. Дагене, Г. К. Григас, К. Ф. Аугутис. — М. : Просвещение, 1993. — 255 с.
4. *Кетков, Ю. Л.* Свободное программное обеспечение : FREE PASCAL для студентов и школьников / Ю. Л. Кетков, А. Ю. Кетков. — СПб. : БХВ — Санкт-Петербург, 2011. — 376 с. + CD-ROM. — (Информатика и информационно-коммуникационные технологии + CD).
5. *Культин, Н. Б.* Turbo Pascal в задачах и примерах / Н. Б. Культин. — СПб. : БХВ — Санкт-Петербург, 2007. — 256 с.
6. *Павловская, Т. А.* Паскаль. Программирование на языке высокого уровня : практикум / Т. А. Павловская. — СПб. : Питер, 2007. — 317 с.
7. *Пильщиков, В. Н.* Сборник упражнений по языку Паскаль : учеб. пособие для вузов / В. Н. Пильщиков. — М. : Наука, 1989. — 153 с.
8. <http://pascalabc.net/ssyilki-dlya-skachivaniya>.



СОДЕРЖАНИЕ

	Введение	3
1	ОСНОВЫ РАБОТЫ В СРЕДЕ PASCAL ABC	5
2	ЛИНЕЙНЫЕ АЛГОРИТМЫ	12
	Примеры программ	13
	<i>Упражнения</i>	17
3	РАЗВЕТВЛЯЮЩИЕСЯ АЛГОРИТМЫ	19
	Примеры программ	22
	<i>Упражнения</i>	25
4	ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ	27
	Примеры программ	30
	<i>Упражнения</i>	33
5	ОПИСАНИЕ МАССИВОВ. ВВОД-ВЫВОД ЭЛЕМЕНТОВ МАССИВА	36
	Примеры программ	36
	<i>Упражнения</i>	38
6	ПОДСЧЕТ СУММ И ПРОИЗВЕДЕНИЙ. ОРГАНИЗАЦИЯ СЧЕТЧИКОВ	39
	<i>Упражнения</i>	43
7	ПОИСК МИНИМАЛЬНОГО (МАКСИМАЛЬНОГО) ЭЛЕМЕНТА МАССИВА	45
	<i>Упражнения</i>	48

8	СОРТИРОВКА МАССИВА	50
	<i>Упражнения</i>	51
9	ПОСТРОЕНИЕ МАССИВА БЕЗ ПОВТОРЕНИЙ	53
	<i>Упражнения</i>	58
10	СТРОКИ В ЯЗЫКЕ ПАСКАЛЬ	60
	Примеры программ	62
	<i>Упражнения</i>	64
11	АЛГОРИТМЫ ВЫДЕЛЕНИЯ СЛОВ СТРОКИ	66
	Примеры программ	67
	<i>Упражнения</i>	73
	Литература	75

Белоцерковская Ирина Ефимовна
Втюрин Максим Юрьевич
Ефимова Элеонора Владимировна

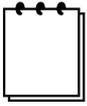
ПРАКТИКУМ
по ПРОГРАММИРОВАНИЮ

Редактор | **Ю. В. Платыгина**
Компьютерная |
верстка | **П. И. Половинкиной**

Оригинал-макет подписан в печать 04.03.2020 г.
Формат 60 × 84 ¹/₁₆. Бумага офсетная. Гарнитура «TimesET».
Печать офсетная. Усл.-печ. л. 4,65. Тираж 100 экз. Заказ 2593.

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



*И. Е. Белоцерковская
М. Ю. Втюрин
Э. В. Ефимова*

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

