

Федеральное агентство по образованию

**Государственное образовательное учреждение  
высшего профессионального образования**

**«МАТИ» – Российский государственный  
технологический университет им. К.Э. Циолковского**

---

Кафедра «Высшая математика»

**Операторы цикла. Примеры программирования циклических  
алгоритмов**

Методические указания к лабораторной работе  
по курсу “Информатика”

Составители: **Сидоров Б. Н.**  
**Никулин А. М.**

Москва 2005

Данное руководство предназначено для студентов, изучающих предмет «Программирование на алгоритмических языках» и выполняющих лабораторные работы по курсу «Информатика». В методических указаниях рассмотрены операторы цикла языка Паскаль и приведены примеры их использования.

## Введение

Основу любой программы, кроме простейших, составляет цикл. Цикл позволяет, записав операторы (называемые телом цикла) один раз выполнять их много раз. Эти операторы называются телом цикла.

В Паскале реализовано три оператора цикла: цикл со счетчиком FOR, используется, когда заранее известно сколько раз должен выполняться цикл; и два условных цикла WHILE и REPEAT, используются, когда известно условие работы цикла. Рассмотрим последовательно эти операторы.

### Оператор цикла FOR :

#### Вид оператора

**FOR I:=** начальное(нач.) значение **TO** конечное(кон.)значение **DO**  
BEGIN

Оператор цикла

END;

I - параметр цикла.

#### Работа

1. Переменная I получает начальное значение
2. Проверка условия: если  $I >$  кон. значения, то выход из цикла  
если  $I \leq$  кон. значения, то 3
3. Выполняется оператор цикла
4. I увеличивается на 1
5. Переход к 2

#### Ограничения:

1. Параметр цикла, нач. значение, кон. значение должны быть целыми числами.
2. Параметр цикла, нач. значение, кон. значение нельзя менять внутри цикла.
3. Для работы цикла нач. значение должно быть меньше кон. значения, если надо, чтобы параметр цикла менялся от большего значения к меньшему с шагом равным -1, то заголовок цикла будет выглядеть так:

**FOR I:=** нач. значение **Downto** кон. значение **DO** (нач. значение больше кон. значение)

Пример.

Цикл распечатывает последовательность 1,2,3 ...10.

```
FOR I:=1 TO 10 DO  
  BEGIN  
    Write( i:4);  
  END;
```

Пример.

Цикл распечатывает последовательность 1,3,5 ...19. ( 10 чисел )

Эту задачу можно решить двумя способами:

1. выразив элемент последовательности через  $I : 2 \cdot I - 1$

```
FOR I:=1 TO n DO  
  BEGIN  
    Write( 2*i-1:4);  
  END;
```

2. Для члена последовательности, введя вспомогательную переменную  $a$ , каждый раз при выполнении цикла  $a$  увеличивать на 2 (шаг) :  $a:=a+2$ ; ( к старому значению  $a$  прибавить 2 и это станет новым значением  $a$ ). Переменная  $I$ , тогда используется только как счетчик, определяя сколько раз исполнится цикл.

```
  a:=1;  
FOR I:=1 TO n DO  
  BEGIN  
    a:=a+2;  
    Write( a:4);  
  END;
```

Этот способ, хоть чуть длиннее, но и более универсальный.

### Использование оператора FOR для вычисления сумм.

#### Пример 1 Вычисление простейших сумм и произведений.

Вычислить сумму  $S=1+2+3+\dots+n$ .

Решение.

Рассмотрим процесс суммирования. Введем переменную накапливающую сумму –  $S$ ; первоначально положим  $S:=0$ ; затем будем добавлять по слагаемому, тогда накопление суммы будет выглядеть так :

```
S0:=0;  
S1:=S0+1;  
S2:=S1+2  
S3:=S2+3
```

.....

Заметив, что в скобках указано предыдущее значение суммы, перепишем процесс суммирования;

```
S0:=0;  
S1:=S0+1;  
S2:=S1+2  
S3:=S2+3
```

.....

Теперь, вспомнив, как работает оператор присваивания (сначала вычисляется выражение, стоящее справа, а затем вычисленное значение присваивается переменной, стоящей слева, т. е. справа стоит предыдущее значение переменной  $S_{i-1}$ , а слева последующее  $S_i$ ), отметим, что индексы к переменной  $S$  можно опустить, тогда

```
S:=0;
S:=S+1;
S:=S+2;
S:=S+3;
.....
```

Если ввести обозначение для слагаемого  $a$ , ( $a=1, 2, 3, \dots$ ), то все операторы, кроме первого, запишутся так ;

```
S:=S+a;
```

Этот единственный оператор и есть оператор суммирования, который и будет размещен в цикле.

Текст программы:

```
Var s,a:integer;
    i,n: integer;
begin
  writeln(' Программа вычисления суммы S=1+2+3...n ');
  writeln(' Ввод количества слагаемых ');
  write('n -> ');
  readln(n);
  s:=0;
  for i:=1 to n do
    begin
      s:=s+i;
      writeln ('i= ',i, ' s= ',s);
    end;
  writeln (' s= ',s);
  readln;
end.
```

Заметим, так как в само слагаемое совпадает с его номером, то программе вместо  $a$  используется  $I$ , также в цикле добавлен отладочный оператор вывода;

Пример 2.

Вычислить произведение  $P=1*2*3 \dots n$ .

Решение:

Этот пример аналогичен предыдущему, только вместо  $S:=S+I$ , используем  $P:=P+I$ ; первоначально же  $P:=0$ ;

Текст программы:

```
Var p:integer;
    i,n: integer;
begin
  writeln(' Программа вычисления произведения P=1*2*3...n.
'); writeln(' Ввод количества сомножителей ');
  write('n -> ');
  readln(n);
  p:=1;
  for i:=1 to n do
    begin
      p:=p*i ;
      writeln ('i= ',i, ' p= ',p);
    end;
  writeln (' p= ',p);
  readln;
```

#### Вычисление суммы арифметической прогрессии.

Арифметическая прогрессия задается формулой:  $a_n := a_{n-1} + d$ , где  $a_n$  – n-ый член прогрессии,  $a_{n-1}$  – предыдущий член,  $d$  – разность прогрессии. Как было разобрано в Пример 1 в Паскале формулу определения арифметической прогрессии можно переписать в виде оператора присваивания, опустив индексы:

**a:= a+d;**

тогда вычисление суммы первых  $n$  членов арифметической прогрессии будет реализовано таким образом:

S:=0; a:= нач. значение; d:= значение;

FOR I:=1 to N DO

  Begin

    S:=S+a;

    a:=a+d;

  End;

Пример 3.

$$S = 2 + 5 + 8 + 11 \dots$$

{ здесь сумма является арифметической прогрессией с  $a_1=2$ ;  $d:=3$  }

Текст программы:

```
Var s,a,d :integer;
    i,n: integer;
begin
  writeln(' Программа вычисления суммы      S= 2+5+8+11.... ');
  writeln(' Ввод количества слагаемых ');
  write('n -> ');

  readln(n);
  s:=0; a:=2; d:=3;
  for i:=1 to n do
    begin
      s:=s+a;
      writeln ('a= ',a, ' s= ',s);
      a:=a+d;
    end;
  writeln (' s= ',s);
  readln;
end.
```

#### Вычисление суммы геометрической прогрессии.

Геометрическая прогрессия задается формулой:  $a_n := a_{n-1} * q$ , где  $a_n$  – n-ый член прогрессии,  $a_{n-1}$  – предыдущий член,  $q$  – знаменатель прогрессии. В Паскале же формулу определения прогрессии можно переписать в виде оператора присваивания, опустив индексы:

$$a := a * q;$$

тогда вычисление суммы первых  $n$  членов геометрической прогрессии будет реализовано таким образом:

S:=0; a:= нач. значение; q:= значение;

FOR I:=1 to N DO

Begin

S:=S+a;

a:=a\*q;

End;

Пример 4.

$$S = 1 + 2 + 4 + 8 + 16 + \dots$$

{ здесь, S – сумма есть геометрической прогрессии с  $a_1=1$ ;  $q:=2$  }

Текст программы:

```
Var s,a,d :integer;
    i,n: integer;
begin
  writeln(' Программа вычисления суммы S=1+2+4+8+16+... ');
  writeln(' Ввод количества слагаемых ');
  write('n -> ');

  readln(n);
  s:=0; a:=2; d:=3;
  for i:=1 to n do
    begin
      s:=s+a;
      writeln ('a= ',a, ' s= ',s);
      a:=a+d;
    end;
  writeln (' s= ',s);
  readln;
end.
```

Пример 5.

Вычислить сумму  $S=1+3/2+5/4+7/8+9/16+11/32+\dots$

Решение:

Здесь слагаемое  $a$  можно представить в виде дроби  $a=c/z$ ; где  $c$  ( $c=1,3,5,\dots$ )– числитель, представляющий из себя арифметическую прогрессию ( $a_1:=1; d:=2$ );  $z$  ( $z=1,2,4,8,\dots$ )– знаменатель, представляющий из себя геометрическую прогрессию ( $a_1:=1; q:=2$ ). Тогда в одном цикле, отдельно вычисляя числитель и знаменатель, вычисление суммы будет выглядеть так:

S:=0;

c:=1; { нач. значение числителя }; d:=2; { разность ар. прогрессии } ;

z:=1; { нач. значение знаменателя }; q:=2 ; { знаменатель геом. прогрессии } ;

FOR I:=1 to N DO

Begin

a:=c/z;

S:=S+a;

c:=c+d;

z:=z\*q;

End;



## Оператор цикла WHILE

Вид оператора

**WHILE** (условие) **DO** оператор;

Если в цикле необходимо выполнять несколько операторов, то их необходимо объединять в один составной оператор, заключая в операторные скобки **begin ... end**.

Работа:

- 1 вычисляется условие.
- 2 если условие принимает значение «истинна», то а) выполняется оператор цикла б) переход к 1.
- 3 если условие принимает значение «ложь», то выполняется оператор, следующий за оператором цикла.

То есть оператор цикла повторяется, пока истинно условие; перед каждым повторением оператора условие вычисляется заново. Для правильной работы цикла необходимо

- перед циклом должны быть определены значения всех переменных, входящих в условие цикла ( инициализация цикла)
- оператор цикла должен менять значения этих переменных, иначе условие всегда будет истинно и цикл будет исполняться бесконечно.

Таким образом работа с циклом сводится к следующей схеме:

```
Инициализация цикла;  
WHILE (условие) DO  
begin  
    оператор;  
    изменение значения переменных цикла  
end;
```

## Примеры программ, использующие цикл WHILE.

### Пример.

Программа распечатывает таблицу перевода дюймов в сантиметры (1дюйм=2.54 см) от 1 дюйма до 35 дюймов с шагом 2 дюйма

```
{ Программа распечатывает таблицу перевода дюймов в сантиметры }
```

```
const n=1; { начальное значение } k=35; { конечное значение } h=2; { шаг }
```

```
var x,y:real; { x - дюймы , y - сантиметры }
```

```
begin
```

```
    x:=n; { переменной цикла x присваивается начальное значение }
```

```
    WHILE (x<=k) DO
```

```
        begin
```

```
            y:=2.54*x; { вычисление сантиметров }
```

```
            writeln( x:6:1, ' duim = ', y:6:2, ' cm'); { печать строки таблицы }
```

```
            x:=x+h; { x увеличивается на шаг - меняется переменная входящая в условие цикла }
```

```
        end; { конец цикла }
```

```
    readln;
```

```
end.
```

**Пример.** Вычислить сумму N членов геометрической прогрессии  
 $S=1+2+4+8+\dots+$ ;

**Решение.**

Эта задача решалась с использованием цикла FOR. При использовании цикла WHILE возникают два отличия: 1. надо задать условие работы цикла ( количество слагаемых – счетчик цикла переменная I меньше N ); 2. счетчик цикла (переменная I) необходимо увеличивать в цикле самостоятельно.

**Текст программы:**

```
Var S, n, a, i, q : integer;
Begin
Writeln ( ' Программа вычисления суммы S=1+2+4+8+ . . . + ');
Writeln ( ' Ввод количество слагаемых ');
Write('n->');
Readln(n);
S:=0;  a:=1; q:=2; i:=1;
WHILE (i<=n) DO
  Begin
    S:=S+a;
    Writeln( 'i= ',i 'a= ', a, 'S= ', S);
    a:= a*q;
    i:= i+1;
  End;
Writeln('S= ', S);
Readln;
End.
```

**Пример.** Вычислить сумму геометрической прогрессии  
 $S=1+2+4+8+\dots+$ ; Вычисление прекратить, когда значение суммы превысит N ;

**Решение.**

Эта задача отличается от предыдущей тем, что заранее не известно количество слагаемых, а известно условие работы цикла ( Сумма меньше N ). Это условие и надо поставить в заголовок цикла. В остальном все без изменений.

### Текст программы:

```
Var S, n, a, i, q : integer;
Begin
Writeln ( ' Программа вычисления суммы S=1+2+4+8+ . . . + ');
Writeln ( ' Ввод максимального значения суммы ');
Write('n->');
Readln(n);
S:=0;  a:=1; q:=2; i:=1;
WHILE (S<=n) DO
  Begin
    S:=S+a;
    Writeln( 'i= ',i 'a= ', a, 'S= ', S);
    a:= a*q;
    i:= i+1;
  End;
Writeln('S= ', S);
Readln;
End.
```

### Работа с числами ( Примеры программ, использующие цикл WHILE).

#### Пример.

Найти сумму цифр числа. ( a=2378 S=2+3+7+8=20 );

Решение

Решение задачи основано на операции целочисленного деления: остаток деления на 10 дает последнюю цифру числа с (последняя цифра) =  $a \bmod 10$  ; а целая часть деления на 10 дает первоначальное число, но уже без последней цифры:  $a:=a \operatorname{div} 10$  . Процесс извлечения и отбрасывания последней цифры продолжается пока в числе еще есть цифры ( число больше нуля ) Тогда алгоритм решения задачи запишется так:

```
WHILE (A>0) DO
  BEGIN
    Получить последнюю цифру
    c:= a mod 10;
    Удалить последнюю цифру из числа
    a:= a div 10;
    Просуммировать последнюю цифру
    s:= s+a;
  END;
```

Текст программы:

```
Var a, s, c : longint;
Begin
  Writeln(' Программа находит сумму цифр числа ');
  Writeln ( ' Введите число ');
  Write( 'a->'); readln(a);
  S:=0;
  WHILE (a<>0) DO
    Begin
      c:= a mod 10;
      a:= a div 10;
      S:=S+c;
    End;
  Writeln ( 'S=', S);
  readln;

End;
```

**Пример.**

Записать число в обратном порядке . ( a=2378 S=8732 );

Решение

Решение этой задачи отличается от предыдущей только тем, что при добавлении последней цифры в сумму эту сумму надо первоначально умножить на 10.  $S:=S*10+c$   
Тогда формирование числа будет происходить так:

a=	c=	S=
2378	-	0
237	8	$0*10+8=8$
23	7	$8*10+7=87$
2	3	$87*10+3=873$
0	2	$873*10+2=8732$

Алгоритм решения :

```
WHILE (A>0) DO
  BEGIN
    Получить последнюю цифру
    c:= a mod 10;
    Удалить последнюю цифру из числа
    a:= a div 10;
    Добавить последнюю цифру к формируемому числу
    s:= 10*s+a;
  END;
```

12

## Оператор цикла REPEAT

вид :

**REPEAT**

операторы

**UNTIL** (условие);

Работа:

1. Выполняются операторы цикла
2. Вычисляется условие
3. Если условие принимает значение «Ложь», то переходит к 1, если значение «Истина», то выполняется оператор, следующий за **UNTIL**

Его отличие от **WHILE**:

1. Условие проверяется в **WHILE** до выполнения тела цикла, а в **REPEAT** после
2. **WHILE** работает когда условие Истина, а **REPEAT** когда –Ложно

Цикл REPEAT используется гораздо реже цикла WHILE, один из немногих случаев, где REPEAT имеет преимущество по сравнению с WHILE это контроль ввода переменных.

Пример. Ввести переменную  $N > 0$ .

REPEAT

Writeln(' Введите  $N > 0$  ');

Write(' N-> ');

Readln(N)

UNTIL ( $N > 0$ );

Цикл будет выполняться до тех пор, пока не будет введена положительная N.

## Примеры программ, использующие цикл REPEAT.

**Пример.** Вычислить сумму N членов геометрической прогрессии

$S = 1 + 2 + 4 + 8 + \dots +$ ;

**Решение.**

Эта задача решалась с использованием циклов FOR и WHILE. При использовании REPEAT отличия минимальны.

**Текст программы:**

```
Var S, n, a, i, q : integer;
```

```
Begin
```

```
Writeln (' Программа вычисления суммы  $S = 1 + 2 + 4 + 8 + \dots +$  ');
```

```
Writeln ( ' Ввод количества слагаемых ');
```

```
Write('n->');
```

```
Readln(n);
```

```
S:=0; a:=1; q:=2; i:=1;
```

```
REPEAT
```

```
    S:=S+a;
```

```
    Writeln( 'i= ',i 'a= ', a, 'S= ', S);
```

```
    a:= a*q;
```

```
    i:= i+1;
```

```
UNTIL (i<=n);
```

```
Writeln('S= ', S);
```

```
Readln;
```

```
End.
```

### **Пример**

Программа считывает число с клавиатуры, определяет четное или нет введённое целое число, после этого выдается запрос на продолжение работы. При положительном ответе – считывается следующее число, при отрицательном - выход из программы.

Эту программу решим двумя способами с использованием циклов WHILE и REPEAT

#### **Текст программы, использующий цикл WHILE.**

```
var k:integer;
    ch:char;
begin
ch:='Y'; {инициализация цикла первоначально символьной переменной ch присваивается
        Y' }
  WHILE ch <>'N' DO { пока ch <>'N' выполнять}
    begin {начало цикла }
      writeln('Введите целое число'); readln(k); { считывание переменной }
      IF (k mod 2 = 0) THEN writeln (' Введённое число ', n:5, ' чётно')
        ELSE writeln (' Введённое число ', n:5, ' нечётно')
      writeln(' continue? Y/N '); { запрос на продолжение }
      readln(ch);( считывание в переменную ch ответа)
    end;{ конец цикла }
  end.{ конец программы }
```

#### **Текст программы, использующий цикл REPEAT**

```
var k:integer;
    ch:char;
begin
  REPEAT
    writeln('Введите целое число'); readln(k); { считывание переменной }
    IF (k mod 2 = 0) THEN writeln (' Введённое число ', n:5, ' чётно')
      ELSE writeln (' Введённое число ', n:5, ' нечётно')
    writeln(' continue? Y/N '); { запрос на продолжение }
    readln(ch);( считывание в переменную ch ответа)
  UNTIL ( ch='N'){ конец цикла }
end.{ конец программы }
```

### Задачи для самостоятельного решения.

Задача №1. Сформировать и распечатать последовательность из 15 целых чисел.

№	Последовательность
1	-1, 3, 7, 11....
2	-4, -1, 2, 5....
3	-2, 1, 4, 7...
4	-2, 3, 8, 13...
5	1, 3, 9, 27....
6	2, 8, 32, 128...
7	2, 10, 50, 250...
8	3, 6, 12, 18...
9	1, -3, 9, -27....
10	2, -8, 32, -128...
11	2, -10, 50, -250...
12	3, -6, 12, -18...

Задача №2. Распечатать таблицу значений функций. Аргумент изменяется равномерно от начального значения, равного а, до конечного значения, равного b, с шагом, равным h.

№	Функция f(x)	Отрезок [a, b]	Число значений аргумента n
1.	$0.5 \cdot \log(0.5 \cdot x^2)$	[1.6; 3.2]	9
2.	$\log(x^2+1)/(x+1)$	[0.8; 1.6]	10
3.	$(0.5 \cdot x+1) \cdot \sin(0.5 \cdot x)$	[1.2; 2.8]	11
4.	$\sqrt{x} \cdot \cos(x^2)$	[0.4; 1.2]	16
5.	$\sin(2 \cdot x) / x^2$	[0.8; 1.2]	9
6.	$(2 \cdot x+0.5) \cdot \sin(x)$	[0.4; 1.2]	12
7.	$\operatorname{tg}(0.5+x^2)/(1+2 \cdot x^2)$	[0.4; 0.8]	13
8.	$\sqrt{x+1} \cdot \cos(x^2)$	[0.2; 0.5]	15
9.	$(\lg(x^2+2))/(x+1)$	[1.4; 2.2]	9
10.	$\operatorname{tg}(x^2)/(1+x)$	[0.5; 1.2]	16
11.	$\sin(x^2-1)/(2 \cdot \sqrt{x})$	[1.3; 2.1]	10
12.	$\sqrt{x+1} \cdot \lg(x+3)$	[1.2; 2.8]	14

Задача№3. Вычислить сумму 15 слагаемых.

№	Слагаемые
1	$S=-1 + 3 + 7 + 11 \dots$
2	$S=-4 + (-1) + 2 + 5 \dots$
3	$S=-2 + 1 + 4 + 7 \dots$
4	$S=-2 + 3 + 8 + 13 \dots$
5	$S=1 + 3 + 9 + 27 \dots$
6	$S=2 + 8 + 32 + 128 \dots$
7	$S=2 + 10 + 50 + 250 \dots$
8	$S=3 + 6 + 12 + 18 \dots$
9	$S=1 + (-3) + 9 + (-27) \dots$
10	$S=2 + (-8) + 32 + (-128) \dots$
11	$S=2 + (-10) + 50 + (-250) \dots$
12	$S=3 + (-6) + 12 + (-18) \dots$

Задача№4. Вычислить сумму N слагаемых.( N вводится)

№	Слагаемые
1	$S=1/3 - 2/6 + 3/12 - 4/24 \dots$
2	$S=2/3 - 4/9 + 6/27 - 8/81 \dots$
3	$S=2/3 - 5/6 + 8/12 - 11/24 \dots$
4	$S=1/2 - 4/6 + 7/18 - 11/54 \dots$
5	$S=3/1 - 8/3 + 13/9 - 18/27 \dots$
6	$S=1/1 - 5/4 + 9/16 - 13/64 \dots$
7	$S=3/4 - 5/8 + 7/16 - 9/32 \dots$
8	$S=4/3 - 8/9 + 12/27 - 16/81 \dots$
9	$S=1/1 - 5/2 + 9/4 - 13/8 \dots$
10	$S=1/5 - 6/25 + 11/125 - 16/625 \dots$
11	$S=1/5 - 7/10 + 13/20 - 19/40 \dots$
12	$S=5/4 - 8/12 + 11/36 - 14/108 \dots$

Задача№5. Вычислить сумму из задания № 4, вычисления производить, пока значение слагаемого остается больше введенного числа E.



Задача №6. Вычислить сумму, пока значение суммы остается меньше введенного числа А.

№	Слагаемые
1	$S = 1 + 1*3 + 1*3*5 + 1*3*5*7 + \dots$
2	$S = 2 + 2*6 + 2*6*10 + 2*6*10*14 + \dots$
3	$S = 2 + \frac{2^2}{1+2} + \frac{2^3}{1+2+3} + \frac{2^4}{1+2+3+4} + \dots$
4	$S = 2 + 2*4 + 2*4*8 + 2*4*8*16 + \dots$
5	$S = 1 + \frac{1*2*3}{2*2*2} + \frac{1*2*3*4}{2*2*2*2} + \frac{1*2*3*4*5}{2*2*2*2*2} + \dots$
6	$S = 3 + \frac{3*3}{1+3} + \frac{3*3*3}{1+3+5} + \frac{3*3*3*3}{1+3+5+7} + \dots$
7	$S = \frac{1*2}{3^2} + \frac{1*2*3}{3^3} + \frac{1*2*3*4}{3^4} + \dots$
8	$S = 2 + \frac{2^3}{2+4} + \frac{2^5}{2+4+6} + \frac{2^7}{2+4+6+8} + \dots$
9	$S = 1 + \frac{2}{1+2} + \frac{2*3}{1+2+3} + \frac{2*3*4}{1+2+3+4} + \dots$
10	$S = 1 + \frac{1*3}{1+3} + \frac{1*3*5}{1+3+5} + \frac{1*3*5*7}{1+3+5+7} + \dots$
11	$S = 1 + \frac{2*4}{2+4} + \frac{2*4*6}{2+4+6} + \frac{2*4*6*8}{2+4+6+8} + \dots$
12	$S = \frac{2}{1+3} + \frac{4}{1+3+5} + \frac{8}{1+3+5+7} + \frac{16}{1+3+5+7+9} + \dots$

Задача №7. Вычислить сумму  $S$ , вычисления производят пока значение слагаемого остается больше введенного числа E.

№	Слагаемые
1	$S = 1 + \frac{1}{1*3} + \frac{1}{1*3*5} + \frac{1}{1*3*5*7} + \dots$
2	$S = \frac{1}{2} + \frac{1}{2*6} + \frac{1}{2*6*10} + \dots$
3	$S = \frac{1}{2} + \frac{1+2}{2^2} + \frac{1+2+3}{2^3} + \frac{1+2+3+4}{2^4} + \dots$
4	$S = \frac{1}{2} + \frac{1}{2*4} + \frac{1}{2*4*8} + \frac{1}{2*4*8*16} + \dots$
5	$S = 1 + \frac{2^3}{1*2*3} + \frac{2^4}{1*2*3*4} + \frac{2^5}{1*2*3*4*5} + \dots$
6	$S = 3 + \frac{1+3}{3^2} + \frac{1+3+5}{3^3} + \frac{1+3+5+7}{3^4} + \dots$
7	$S = \frac{3^2}{2} + \frac{3^3}{2*3} + \frac{3^4}{2*3*4} + \frac{3^5}{2*3*4*5} + \dots$
8	$S = 2 + \frac{2+4}{2^3} + \frac{2+4+6}{2^5} + \frac{2+4+6+8}{2^7} + \dots$
9	$S = 1 + \frac{1+2}{2} + \frac{1+2+3}{2*3} + \frac{1+2+3+4}{2*3*4} + \dots$
10	$S = 1 + \frac{1+3}{1*3} + \frac{1+3+5}{1*3*5} + \frac{1+3+5+7}{1*3*5*7} + \dots$
11	$S = 1 + \frac{2+4}{2*4} + \frac{2+4+6}{2*4*6} + \frac{2+4+6+8}{2*4*6*8} + \dots$
12	$S = \frac{1+3}{2} + \frac{1+3+5}{4} + \frac{1+3+5+7}{8} + \frac{1+3+5+7+9}{16} + \dots$

## ЛИТЕРАТУРА

1. Попов В. Б. Turbo Pascal 7.0 для школьников. М.: Финансы и статистика, 1996.

## ОГЛАВЛЕНИЕ

Введение _____	3
Оператор цикла FOR _____	3
Использование оператора FOR для вычисления сумм. _____	4
Оператор цикла WHILE _____	9
Примеры программ, использующие цикл WHILE. _____	9
Работа с числами _____	11
Оператор цикла REPEAT _____	13
Примеры программ, использующие цикл REPEAT. _____	13
Задачи для самостоятельного решения. _____	15
Литература _____	19

Борис Николаевич Сидоров  
Алексей Михайлович Никулин

Методические указания к лабораторной работе по  
курсу " Информатика"

«Операторы цикла. Примеры программирование циклических алгоритмов»

*Редактор М.А.Соколова*

Подписано в печать      объем      п.л.

Тираж 100 экз. Бесплатно. Заказ

---

Ротап rint МАТИ – РГТУ , Берниковская наб.,14