

ЛАБОРАТОРНАЯ РАБОТА № 4

Управляющие операторы для организации циклов.

Программы с циклами

1. Краткие теоретические сведения

Циклом называется процесс исполнения группы операторов программы заданное количество раз, либо до тех пор, пока выполняется или не выполняется некоторое поставленное условие. Процесс исполнения группы операторов цикла один раз называется итерацией цикла. Группа операторов, расположенная между началом и концом цикла называется *телом цикла*. В С# есть три основных вида циклов: цикл `for` с параметром (счётчиком), цикл `while` с предусловием и цикл `dowhile` с постусловием.

Оператор цикла `for`

Оператор цикла `for` (для) служит для организации циклов с параметром (счётчиком). Это наиболее распространённый оператор цикла. Он проще не только для чтения и понимания, но и проверки корректности цикл. Оператор цикла `for` имеет следующую структуру.

```
for (выражение 1; условие; выражение 2)  
{  
    // тело цикла  
}
```

Открывающиеся и закрывающаяся фигурная скобки являются соответственно началом и концом тела цикла. В круглых скобках указаны следующие параметры цикла:

- *выражение 1*, в нём указывается имя переменной управления циклом и присваивается ей начальное значение;
- *условие*, представляет собой булево выражение, проверяющее значение переменной управления циклом, если результат проверки истинен, то цикл продолжается, если ложен, то цикл завершается;
- *выражение 2*, как правило, это арифметическое выражение, счётчик, определяет на каждой итерации цикла порядок изменения переменной управления циклом, на определённую величину – шаг цикла.

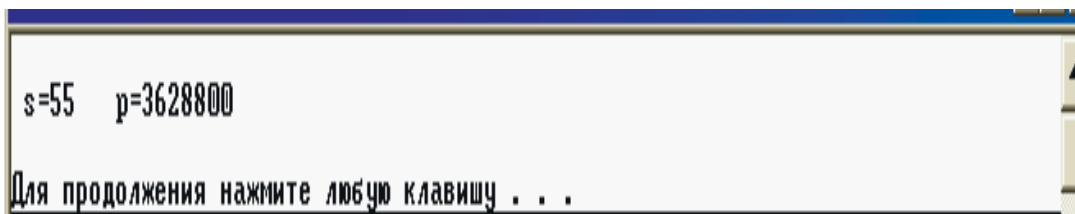
При работе цикла сначала выполняется *выражение 1*. В результате переменная управления циклом принимает своё первоначальное значение. Затем вычисляется *условие* и, если оно истинно, выполняются операторы тела цикла, заключённые в фигурные скобки. По достижению конца тела цикла (закрывающаяся фигурная скобка) управление переходит к вычислению *выражение 2*. В результате переменная цикла принимает новое значение, после чего вновь вычисляется *условие* и в случае его истинности цикл повторяется, в противном случае управление передаётся первому оператору, следующему за закрывающейся фигурной скобкой.

Пример 1.

```
int s, p, n;
s = 0;
p = 1;
for (n = 1; n <= 10; n = n + 1)
{
    s = s + n;
    p = p * n;
```

```
}
Console.WriteLine(" s=" + s + " p=" + p);
```

В данной программе определяется сумма s и произведение p чисел от одного до десяти включительно. Имя переменной управления циклом n , её заданное начальное значение 0 . Условие продолжения цикла $n \leq 10$. Порядок изменения переменной управления циклом, счётчик – $n = n + 1$. Таким образом, n изменяется от 1 до 10 с шагом 1 . При этом на каждой итерации сумма s увеличивается на n , а произведение p в n раз. Ниже представлены результаты работы программы.

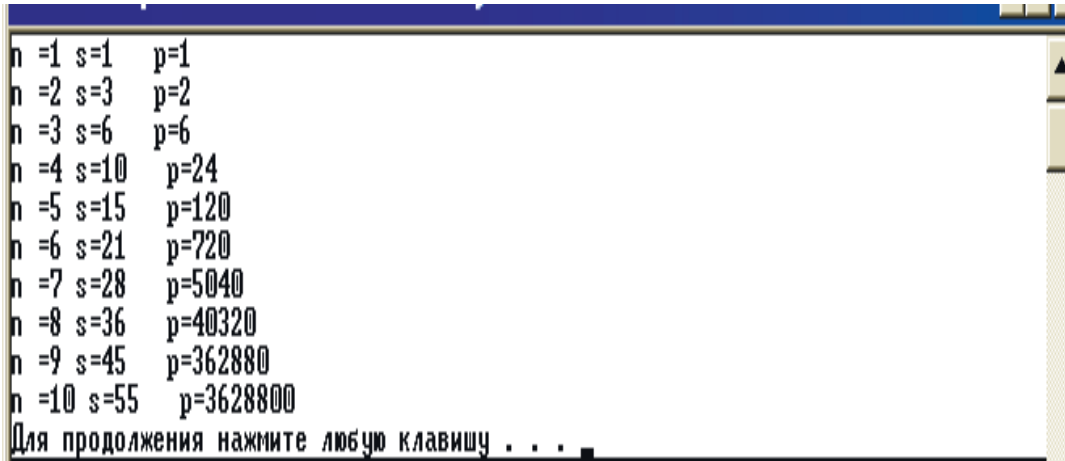


Если в рассмотренной программе вывод организовать внутри цикла, то можно проанализировать увеличение сумм s и произведения p на каждой итерации цикла. Кроме того можно записать программу более компактно см. пример 2.

Пример 2.

```
int s, p, n;
s = 0;
p = 1;
for (n = 1; n <= 10; n++) // или n += 1
{
    s = s + n;
    p = p * n;
    Console.WriteLine("n =" + n + "s=" + s + "p="
```

```
+p);  
}
```



```
n =1 s=1 p=1  
n =2 s=3 p=2  
n =3 s=6 p=6  
n =4 s=10 p=24  
n =5 s=15 p=120  
n =6 s=21 p=720  
n =7 s=28 p=5040  
n =8 s=36 p=40320  
n =9 s=45 p=362880  
n =10 s=55 p=3628800  
Для продолжения нажмите любую клавишу . . .
```

Результаты работы данной программы

Поскольку добавление 1 к переменной, в частности к $n = n + 1$, является распространённой операцией в языке C# предусмотрена сокращённая запись этой операции, а именно $n ++$, как в примере 2, или $n += 1$. Этот оператор называется оператором инкремента, он часто используется как в цикле `for` так и в других операторах цикла.

В теле цикла `for` нельзя изменять его параметры. Если в цикле отсутствует параметр *условие*, то это соответствует значению `true`. Цикл, записанный в виде `for(;;)` – является бесконечным циклом.

Оператор цикла `while`

Оператор цикла `while` (пока), как и оператор цикла `for`, является циклом с предусловием. Это означает, что *условие* проверяется до начала цикла, и если оно имеет значение `false`, то цикл ни

разу не выполняется. Отличие оператор цикла **while** от оператора цикла **for** состоит в том, что в нём заранее не определено количество итераций в цикле. Форма записи оператора.

```
while (условие)
{
    // тело цикла
}
```

Операторы в теле цикла исполняются до тех пор, пока *условие* цикла выполняется, т.е. имеет значение **true**. Если вместо *условия* указано служебное слово **true**, т.е. **while (true)**, то цикл будет бесконечным.

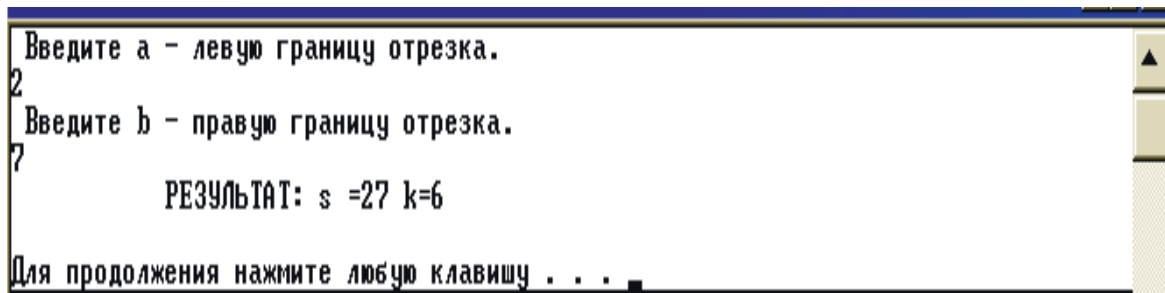
Пример 3. Составить программу, которая определяет сумму *s* и количество *k* целых чисел лежащих на заданном отрезке [*a*; *b*]. Границы отрезка целые числа.

```
int a,b,s,k;
Console.WriteLine(" Введите a" +
"- левую границу отрезка.");
a = int.Parse((Console.ReadLine()));
Console.WriteLine(" Введите b" +
"- правую границу отрезка.");
b = int.Parse((Console.ReadLine()));
s = 0;
k = 0;
while ( a <= b)
{
    S += a;        // означает s = s + a
    k++;           // означает k = k+1
}
```

```

        a++;          // означает a = a + 1
    }
    Console.WriteLine('\t' + "    РЕЗУЛЬТАТ:" +
        " s =" + s + " k=" + k + '\n');

```



Переменная *a* управления циклом имеет начальное значение равное левой границе заданного отрезка. Перед каждым шагом она сравнивается с переменной *b*, которая имеет значение правой границы заданного отрезка. Если результат сравнения *true*, то цикл продолжается. Сумма чисел отрезка, переменная *s*, увеличивается на величину значения переменной *a*, которая является очередным числом заданного отрезка. Количество чисел отрезка, переменная *k*, и переменная *a* увеличиваются на единицу. Если результат сравнения *false*, то цикл заканчивается, и выдаются, например следующие результаты работы программы.

Оператор цикла *do-while*

Оператор цикла *do-while* является версией цикла *while* с постусловием. Это означает, что *условие* цикла проверяется после исполнения операторов тела цикла. Следовательно, в таком цикле

одна, первая итерация, всегда будет выполняться. Иногда это удобно использовать. Форма записи оператора.

```
do
{
    //тело цикла
}
while (условие)
```

После исполнения первой итерации цикла цикл продолжает работать до тех пор, пока *условие* цикла имеет значение `true`, в противном случае цикл заканчивается. В случае использования данного оператора в программе примера 3 она будет выглядеть следующим образом, см. пример 4.

Пример 4.

```
int a, b, s, k;
Console.WriteLine(" Введите a"+
" - левую границу отрезка");
a =int.Parse((Console.ReadLine()));
Console.WriteLine(" Введите b"+
" - правую границу отрезка");
b = int.Parse((Console.ReadLine()));
s = 0;
k = 0;
do
{s += a; k++; a++;}
while (a <= b);
Console.WriteLine('\t' + "    РЕЗУЛЬТАТ:" +
" s =" + s + " k=" + k + '\n');
```

Операторы break и continue

Для управления циклом имеются специальные операторы – `break` и `continue`. Оператор `break` вызывает прекращение выполнения цикла и передачу управления первому оператору, следующему непосредственно за циклом. Оператор `continue` передаёт управление в начало цикла, к проверке условия. Обычно эти операторы используются совместно с оператором `if`.

Вложенные циклы

Цикл можно размещать внутри другого цикла. Размещение одной конструкции цикла в другой называется вложением циклов, а сам цикл вложенным.

При организации циклов, в том числе и вложенных, необходимо соблюдать следующие правила:

- переменная, объявленная в теле цикла, определена только внутри этого цикла;
- вход в цикл осуществляется через его заголовок;
- вход во внутренний цикл осуществляется после входа во внешний;
- выход из цикла, в том числе и вложенного возможен в любом месте тела цикла;
- параметры циклов со счётчиком не должны изменяться в теле цикла;
- вложенные циклы не должны пересекаться, т.е. начало и конец внутреннего цикла должны находиться во внешнем цикле;
- допускается делать вложение разных конструкций циклов.

Пример 5. (пример выполнения задания 1)

Написать и отладить программу вычисления значений функции $y = \frac{ax}{\sqrt{1+ax^2}}$, для каждого из заданных значений параметра a (0,5; 1,0; 1,5; 2,0) и при всех заданных значениях аргумента x (от 1 до 7 с шагом 0,25).

```
StreamWriter p = new StreamWriter("rez.txt");
float a,x,y;
p.WriteLine("          РЕЗУЛЬТАТЫ РАСЧЁТА");
for (a = 1; a <= 2; a += 0.5f) // Заголовок внешнего
                               // цикла по a
{
    // Начало внешнего цикла по a
    p.WriteLine(" a=" + a);
    for (x = 1; x <= 7; x += 0.25f) //Заголовок
                                    // внутреннего
                                    // цикла по x
    {
        // Начало внутреннего цикла по x
        y = (float)(a * Math.Cos(x) /
        Math.Sqrt(1 + a * x * x));
        p.WriteLine("      x= " + x +
                    '\t' + "      y= " + y);
        /* Конец внутреннего цикла по x */
    }
    /* Конец внешнего цикла по a */
}
p.Close();
```

Результаты расчёта

(файл rez.txt)

РЕЗУЛЬТАТЫ РАСЧЁТА	
a=1	
x= 1	y= 0,3820514
x= 1,25	y= 0,1969803
x= 1,5	y= 0,03923794
...	
x= 6,75	y= 0,1308689
x= 7	y= 0,1066179
a=1,5	
x= 1	y= 0,5125757
x= 1,25	y= 0,2586599
x= 1,5	y= 0,05072828
...	
x= 6,75	y= 0,1608578
x= 7	y= 0,1310171
a=2	
x= 1	y= 0,6238874
x= 1,25	y= 0,310508
x= 1,5	y= 0,06032489
...	

Суммирование рядов

Выше было рассмотрено использование циклических конструкций для вычисления, накапливания сумм (примеры 1 – 4). Накапли-

вание сумм используется при вычислении, суммировании рядов, их членов. С помощью рядов могут вычисляться различные функции. Рассмотрим такую задачу на примере.

Пример 6. (пример выполнения задания 2)

Написать и отладить процедуру для приближённого вычисления функции $y = \sin(x)$ с помощью ряда

$$s(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Суммирование членов ряда проводить включительно до члена ряда, значение которого по абсолютной величине меньше чем 10^{-6} .

Вычислить сумму членов заданного ряда для угла в 30 градусов и сравнить полученное значение со значением, вычисленным непосредственно, с помощью функции $y = \sin(x)$.

Программа примера 6

```
double g, pi, x, y, s, a, p, d, чл, зн;
int n, k;
m1: Console.WriteLine(" Введите значение угла в
градусах");
g = double.Parse((Console.ReadLine()));
pi = Math.PI;      // Задание числа pi
x = g*pi/180;      // Перевод градусов в радианы.
Console.WriteLine('\t'+"Промежуточные результаты"
+'\n');

K = 0;           // Номер итерации.
S = 0; p = 1;
n = 3; d = 1;
a = x;          // Первый член ряда.
```

```

while (Math.Abs(a)>= 0.000001)
{
    S += a;           // Сумма ряда на текущей
                      // итерации.
    чл = x*x*x*d;     // Вычисление числителя ряда.
    p = -p*(n-1)*n;   // Вычисление факториала.
    зн = p;           // Знаменатель ряда.
    a = чл/зн;        // Член ряда на текущей итерации.
    d = x*x;
    n += 2;
    k++;              // Счётчик итераций.
    Console.WriteLine('\t + " Итерация №"
+ k + '\n' + " a=" + a + " s=" + s + '\n');
}
y = Math.Sin(x);     // Непосредственное вычисление
                      // синуса.
Console.WriteLine('\t' + " РЕЗУЛЬТАТЫ:" + '\n'
+ " Заданное значение угла в град.=" + g + '\n'
+ " Вычисленная сумма ряда S =" + s + '\n' + "
Количество членов ряда - " + k + '\n'
+ " Функция SIN(X)=" + y + '\n');
Console.WriteLine('\n' + " Для повторного ввода"
+ " набрать любую клавишу." + '\n'
+ " Для завершения программы нажмите Enter.");
string rep = Console.ReadLine();
if (rep != "") goto m1;

```

Результаты расчёта по программе примера 6

```

Введите значение угла в градусах
30

Промежуточные результаты

Итерация №1
a=-0,023924596203935 s=0,523598775598299

Итерация №2
a=0,000327953194428671 s=0,499674179394364

Итерация №3
a=-7,80840939115883E-06 s=0,500002132588792

Итерация №4
a=1,08450130432761E-07 s=0,499994324179401

РЕЗУЛЬТАТЫ:
Заданное значение угла в градусах =30
Вычисленная сумма ряда S =0,499994324179401
Количество членов ряда - 4
Функция SIN(X)=0,5

Для повторного ввода набрать любую клавишу.
Для завершения программы нажмите Enter.

```

2. Практическая часть

Задание 1. Составить и отладить программу вычисления заданной в таблице 4.1 функции $y(x)$ для каждого из заданных значений параметра a и при всех заданных значениях аргумента x .

Таблица 4.1

	Заданная функция $y(x)$	Значение аргумента $x_{нач}; x_{кон}; \Delta x$	Значение параметра a
1.	$y = (1/a) \cdot \exp(-(x/a)^2)$	-1,5; 1,5; 0,1	1; 1,1; 1,2; 1,3
2.	$y = x \cdot e^{-x/a}$	0; 4; 0,2	0,25; 0,5; 0,75; 1,0
3.	$y = 1 / \sqrt{(1 - x^2)^2 + 4a^2 x^2}$	0; 2; 0,05	0,1; 0,2; 0,3; 0,4

4.	$y = x \cdot \operatorname{tg} a - x^2 / \cos^2 a$	0; 0,5; 0,02	$15^0; 30^0; 45^0; 60^0$
5.	$y = a \cdot x^a \cdot e^{-x/a}$	0; 10; 0,25	1; 1,25; ... 2,0
6.	$y = e^{-xa} \cdot \sin x$	0; π ; $\pi/36$	0; 0,5; ... 2,0
7.	$y = ((x+a)^{2/3} - (x-a)^{2/3})/a$	-4; 4; 0,2	1; 2; 3; 4
8.	$y = a^{-x} - a^{-a \cdot x}$	0; 2; 0,05	10; 8; 6; 4; 2
9.	$y = (1 - \exp(-(x/a)^2))/a$	-2; 2; 0,1	1,25; 1,5; 1,75; 2
10.	$y = a^3 / (a^2 + x^2)$	-2; 2; 0,05	0,5; 1,0; 1,5; 2,0
11.	$y = \frac{\sqrt[4]{(a+1)x + e^{-x^3}}}{\sqrt{2ax}}$	1; 7; 0,25	0,5; 0,75; 1,0; 1,25
12.	$y = \frac{\sqrt[3]{ax^2 + e^{-x^2}}}{\sqrt{(a+1)x}}$	1; 7; 0,25	0,5; 0,75; 1,0; 1,25
13.	$y = \frac{\sqrt[3]{x + ae^{-x^2}}}{\sqrt{ax}}$	1; 7; 0,25	0,5; 0,75; 1,0; 1,25
14.	$y = (\sqrt{x + ae^{-x^2}})/(ax^2)$	1; 7; 0,25	0,5; 0,75; 1,0; 1,25
15.	$y = \frac{\operatorname{arctg}(\frac{x^2}{2a})}{x^3 + a}$	0; 2,5; 0,1	0,5; 1,0; 1,5; 2,0
16.	$y = \frac{\operatorname{arctg}(\frac{x}{2a})}{x^2 + 2a}$	0; 3; 0,1	0,5; 0,75; 1; 1,25
17.	$y = \frac{ax}{a + \sqrt[3]{1 + x^2}}$	-3; 3; 0,2	0,5; 1,0; 1,5; 2,0
18.	$y = \operatorname{Cos}^2(2ax)/(3a)$	$0^0; 360^0; 6^0$	1; 1,25; 1,5; 2,0
19.	$y = \operatorname{Sin}^2(ax)/(a+2)$	$0^0; 360^0; 6^0$	1; 1,25; 1,5; 2,0
20.	$y = a \operatorname{Cos}(2x)/(a+4)$	$0^0; 360^0; 6^0$	1; 2; 3; 4
21.	$y = a^2 e^{-x} / (2 + a^2)$	-4; 4; 0,25	1; 2; 3; 4

22.	$y = \frac{\sqrt[3]{ax^2 + e^{-x}}}{ax^2}$	1; 7; 0,25	0,5; 1,0; 1,5; 2,0
23.	$y = \frac{\sqrt{x + a^2} e^{-x^2}}{a + x}$	1; 7; 0,25	0,5; 1,0; 1,5; 2,0
24.	$y = (x^2 + a)^{\frac{xa}{x-1}}$	0; 0,8; 0,05	0,5; 1,0; 1,5; 2,0
25.	$y = (x + 4a)^{\frac{ax}{x-1}}$	0; 0,8; 0,05	0,5; 1,0; 1,5; 2,0
26.	$y = \ln^2 \left \frac{xa}{a+x} \right $	2; 12; 0,5	0,5; 1,0; 1,5; 2,0
27.	$y = \ln \left \frac{ax}{1+ax} \right $	2; 12; 0,5	0,5; 1,0; 1,5; 2,0
28.	$y = -\ln \left \frac{x+a}{1+x^2} \right $	1; 10; 0,5	1; 2; 3; 4
29.	$y = \frac{ax^2}{\sqrt{1 + ax^2}}$	-2; 2; 0,1	1; 2; 3; 4
30.	$y = -\ln \left \frac{ax}{a+x} \right $	1; 11; 0,5	1; 2; 3; 4

Задание 2. Составить и отладить программу для приближённого вычисления заданной функции $y(x)$ путём суммирования членов заданного её ряда $s(x)$ см. таблицу 4.2. Суммирование членов ряда проводить до члена ряда, значение которого по абсолютной величине не будет превышать 10^{-6} .

Вычислить сумму ряда $s(x)$ и непосредственно функцию $y(x)$ при указанных в таблице 4.2 контрольных значениях аргумента x . Сравнить и проанализировать полученные значения.

Таблица 4.2

№	Ряд	Контрольные значения аргумента x и функция y(x)
1	$S = 1 + \frac{x}{2} - \frac{1 \cdot x^2}{2 \cdot 4} + \frac{1 \cdot 3 \cdot x^3}{2 \cdot 4 \cdot 6} - \frac{1 \cdot 3 \cdot 5 \cdot x^4}{2 \cdot 4 \cdot 6 \cdot 8} + \dots$	-0.84; 1; 2; $y = \sqrt{x+1}$
2	$S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$	1; 10; -10; $y = e^x$
3	$S = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{2^{2n-1}}{(2n)!} x^{2n}$	$\pi/6$; $13\pi/6$; $25\pi/6$; $y = \sin^2 x$
4	$S = 1 + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-3)}{n! \cdot 2^{3n}} (x-4)^n$	2; 1; 0; $y = 0.5 \cdot \sqrt{x}$
5	$S = \frac{x}{2} + \frac{x^2}{2^2} + \frac{x^3}{2^3} + \dots$	-1; 1; 1.9; $y = \frac{x}{2-x}$
6	$S = \frac{x}{1} + \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots$	0.5; $\sqrt{2}/2$; -1; $y = \arcsin x$
7	$S = \frac{x}{1} - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$	$1/\sqrt{3}$; $-1/\sqrt{3}$; 1; $y = \arctg x$
8	$S = 1 + \sum_{n=1}^{\infty} \frac{(n^2 + 1) \cdot x^n}{n! \cdot 2^n}$	2; 20; -15; $y = (z^2 + z + 1)e^x$, где $z = x/2$
9	$S = \sum_{n=1}^{\infty} \sin(\pi n / 4) \frac{\sqrt{2^n}}{n!} x^n$ *см. ЧОСКУ	$\pi/4$; π ; -5.5π ; $y = e^x \sin x$
10	$S = x + \sum_{n=1}^{\infty} (-1)^n \frac{2^{2n} \cdot x^{2n+1}}{(2n)!}$	$\pi/6$; π ; 4π ; $y = x \cdot \cos 2x$
11	$S = 2 \cdot \left(\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots \right)$	0.5; 0.9; 0.99; $y = \ln \frac{1+x}{1-x}$
12	$S = e^{-2} \left(1 + \sum_{n=1}^{\infty} \frac{(x+2)^n}{n!} \right)$	1; 9; -15; $y = e^x$

13	$S = 1 + \frac{2 \cdot x}{1!} + \frac{3 \cdot x^2}{2!} + \frac{4 \cdot x^3}{3!} + \dots$	1; 9; -11; $y = (1+x) \cdot e^x$
14	$S = \frac{2 \cdot x}{1} - \frac{2^2 x^2}{2} + \frac{2^3 \cdot x^3}{3} - \dots$	1/4; -1/3; 1/2; $y = \ln(1+2x)$
15	$S = 3x + 8x^2 + \dots + n(n+2)x^n + \dots$	0.2; 0.6; 0.9; $y = \frac{x(3-x)}{(1-x)^3}$
16	$S = 1 + \frac{x \ln 2}{1!} + \frac{x^2 \ln^2 2}{2!} + \frac{x^3 \ln^3 2}{3!} + \dots$	2; 10; -10; $y = 2^x$
17	$S = 1 - \frac{2^2 x^2}{2!} + \frac{2^4 x^4}{4!} - \frac{2^6 x^6}{6!} + \dots$	$\pi/6$; π ; 5π ; $y = \cos 2x$
18	$S = 1 + \sum_{n=1}^{\infty} \frac{\cos(n\pi/4)}{n!} x^n$ *см. чоккы	$\pi/6$; π ; -5π ; $y = e^z \cos z$ где $z = x/\sqrt{2}$
19	$S = 1 - 2x + 3x^2 - 4x^3 + \dots$	0.2; 0.6; 0.9; $y = 1/(1+x)^2$
20	$S = \sum_{n=1}^{\infty} x^n \cdot \sin(n\pi/4)$ *см. чоккы	0.5; 0.8; 0.99; $y = (x/\sqrt{2})/(1-\sqrt{2} \cdot x + x^2)$
21	$S = 1 + \frac{\cos(x)}{1!} + \frac{\cos(2x)}{2!} + \dots + \frac{\cos(nx)}{n!} + \dots$ *см. чоккы	$\pi/6$; $-\pi$; 10π ; $y = e^{\cos x} \cos(\sin x)$
22	$S = -\frac{(2x)^2}{2!} + \frac{(2x)^4}{4!} - \dots (-1)^n \frac{(2x)^{2n}}{(2n)!} + \dots$	$\pi/6$; $13\pi/6$; $25\pi/6$; $y = 2(\cos^2 x - 1)$
23	$S = \sum_{n=0}^{\infty} \frac{1}{2n+1} \cdot \left(\frac{x-1}{x+1} \right)^{2n+1}$	0.5; 0.1; 0.01; $y = 0.5 \cdot \ln(x)$
24	$S = \frac{1}{\sqrt{2}} + \sum_{n=1}^{\infty} \cos\left(\frac{2n+1}{4} \cdot \pi\right) \cdot \frac{(x-\pi/2)^n}{n! \cdot 2^n}$	$2\pi/3$; $8\pi/3$; $28\pi/3$; $y = \cos(x/2)$
25	$S = x + \sum_{n=1}^{\infty} (-1)^n \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1) \cdot x^{2n+1}}{2^n \cdot n! \cdot (2n+1)}$	0.5; 0.9; 1; $y = \ln(x + \sqrt{1+x^2})$
26	$S = x + 2 \cdot \left(\frac{x^3}{1 \cdot 3} - \frac{x^5}{3 \cdot 5} + \frac{x^7}{5 \cdot 7} - \frac{x^9}{7 \cdot 9} + \dots \right)$	$1/\sqrt{3}$; $-1/\sqrt{3}$; 1 $y = (1+x^2) \cdot \arctg(x)$;

27	$S = \frac{\sqrt{3}}{2} + \sum_{n=1}^{\infty} \sin(\pi/3 + n\pi/2) \cdot \frac{\pi^n (x-1)^n}{3^n \cdot n!}$	0.5; 6.5; 13.5; $y = \sin(\pi x/3)$
28	$S = \sum_{n=1}^{\infty} (-1)^{n-1} (1+2^n) \frac{x^n}{n}$	0.25; 0.45; 0.5; $y = \ln(1+3x+2x^2)$
29	$S = -1 + \frac{x+1}{3} + \sum_{n=2}^{\infty} \frac{2 \cdot 5 \cdot 8 \cdot \dots \cdot (3n-4)}{3^n n!} (x+1)^n$	-1.331; -0.729; 0; $y = \sqrt[3]{x}$
30	$S = 4 + \sum_{n=1}^{\infty} (2+2^n) \frac{x^n}{n!}$	2; -1; -10; $y = (1+e^x)^2$

*) для рядов, отмеченных звёздочкой, при оценке погрешности в членах ряда не учитывать синусы и косинусы.