Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

|  |
| --- |
|  |
|  |

# оТЧЕТ

по лабораторной работе

на тему:

СЛОЖНЫЕ ЦИКЛЫ

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 951002 |  | А. А. Радько |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2019

1. Теоретические сведения по теме лабораторной работы

**Алгоритм** - система правил, четко описывающая последовательность действий, которые необходимо выполнить для решения задачи.

**Способы описания алгоритмов:**

1. запись на естественном языке (словесное описание)
2. изображение в виде схемы (графическое описание)
3. запись на алгоритмическом языке (составление программы)

**Критерии правильного алгоритма:**

1. **Дискретность** – значения величин в каждый следующий момент времени должны получаться по определенным правилам из значений величин, имевшихся в предшествующий момент времени.
2. **Определенность (детерминированность)** – каждое правило алгоритма должно быть однозначным. Значения величин, получаемых в какой-то момент времени, однозначно связаны со значениями величин, вычисленных ранее.
3. **Результативность (конечность)** – алгоритм должен приводить к решению задачи за конечное число шагов.
4. **Массовость** – алгоритм должен разрабатываться в общем виде так, чтобы его можно было применить для класса задач, различающихся лишь исходными данными.

**Можно выделить три типа алгоритмов:**

* **Линейные** (направление вычислений является единственным);
* **Разветвляющиеся** (направление вычислений определяется некоторыми условиями);
* **Циклические.** (отдельные участки вычислений выполняются многократно.)

**Цикл** – процесс, в котором отдельные участки вычислений выполняются многократно.

**Классификация циклов:**

В соответствии со взаимным расположением циклов в теле программы или алгоритма различают следующие циклы:

**1) простые** – циклы, не содержащие внутри себя других циклов;

**2) сложные** – циклы, содержащие внутри себя другие циклы;

**а) вложенные** **(внутренние)** – циклы, входящие в состав других циклов (цикл в цикле);

**б) внешние** – циклы, не являющиеся составной частью других циклов, но содержащие в своем составе внутренние циклы.

В зависимости от местоположения условия выполнения цикла различают следующие циклы:

1) циклы с предусловием;

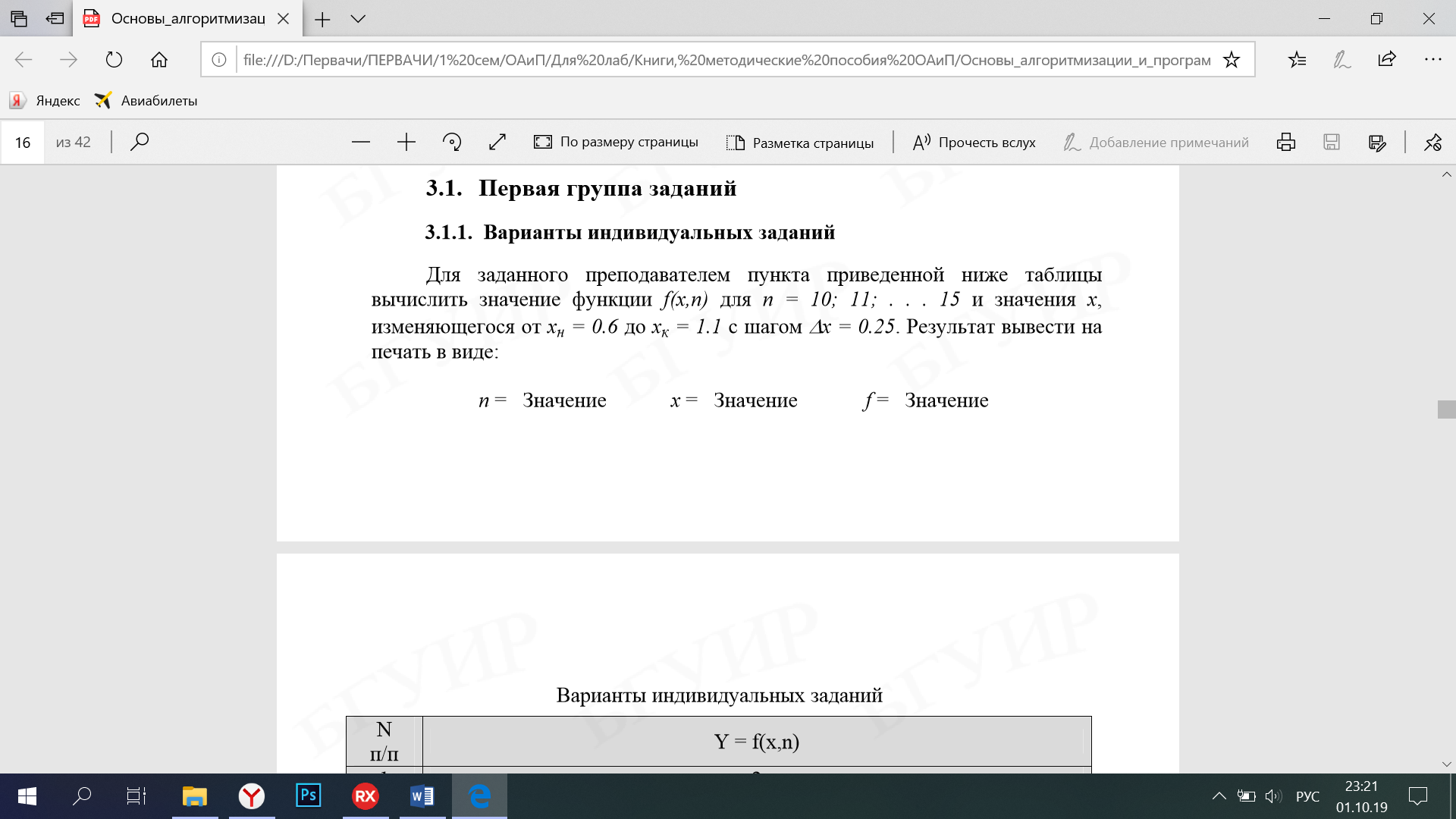
2) циклы с постусловием.

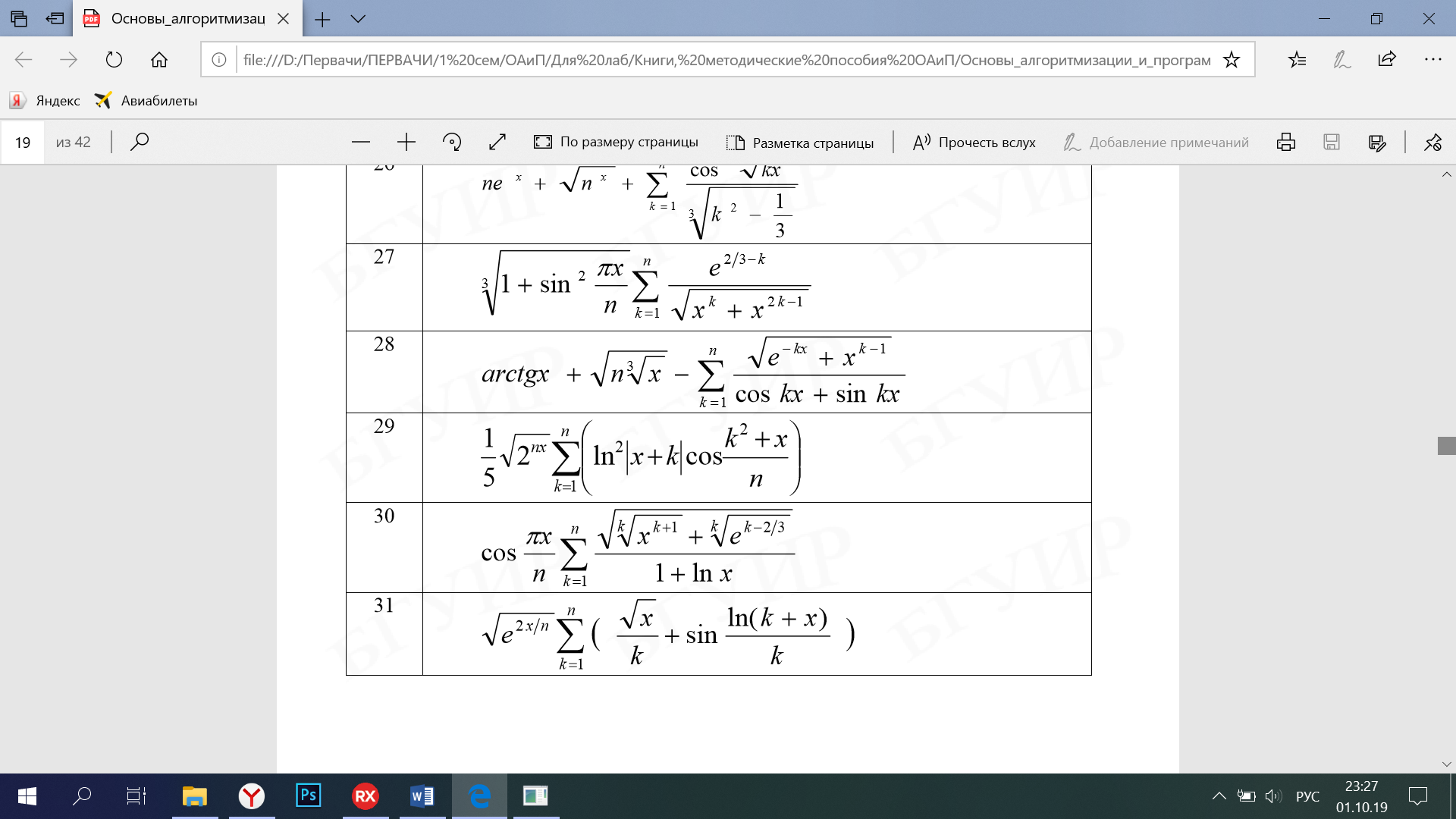
В соответствии с видом условия выполнения циклы делятся на следующие виды:

1) циклы с параметром;

2) итерационные циклы.

1. Задание на лабораторную работу
   1. Постановка задачи



* 1. 
  2. Эскиз ожидаемого результата

n = 10 x = 1.10 f = 0.9068590983

n = 11 x = 1.10 f = 0.9016174525

n = 12 x = 1.10 f = 0.8975405673

n = 13 x = 1.10 f = 0.8943129299

n = 14 x = 1.10 f = 0.8917175113

n = 15 x = 1.10 f = 0.8896015831

1. Выполнение

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя  идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения | Имя цикла, в котором происходит изменение переменной |
| n | Счетчик | integer | 1 | n+1 | Цикл B |
| k | Счетчик | integer | 10 | k+1 | Цикл A |
| x | Текущее значение аргумента | real | 0.6 | x+0.25 | Циклы А, В, С |
| f | Переменная для подсчета функции | real | - | Ffirst\*Sum | Цикл В |
| Ffirst | Вспомогательная переменная | real | - | (exp(ln((1+(sin(pi\*x/n)\*  \*sin(pi\*x/n))))/3)) | Цикл **B** |
| Sum | Переменная для подсчета суммы | real | 0 | Sum+((exp(2/3-k))/(sqrt(exp(ln(x)\*k))\*exp(ln(x)\*(2\*k-1)))) | Циклы А, В |

* 1. Разработка алгоритма

Схема работы программы

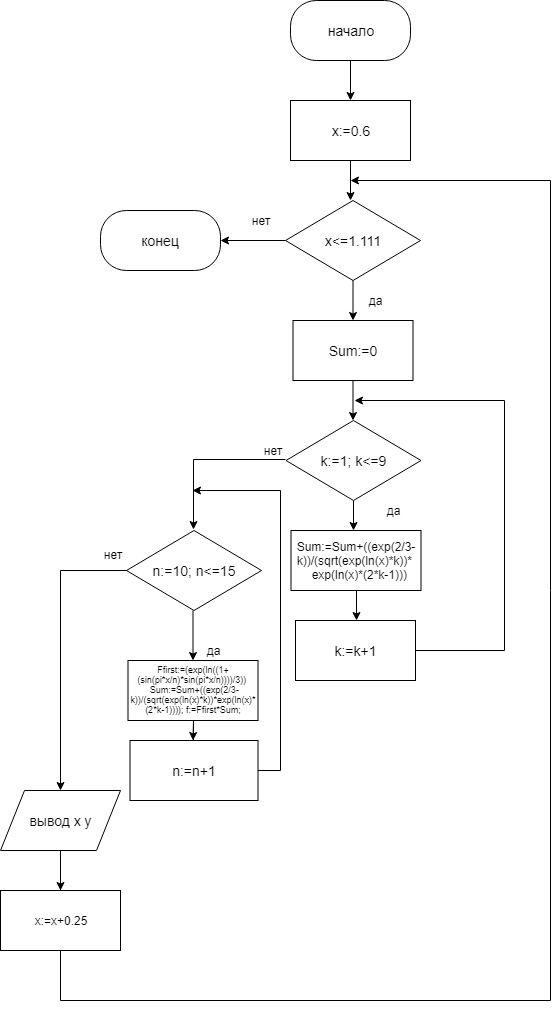


Рисунок 3.1 – Схема работы программы

* 1. Текст программы и его описание

program laba2zadanie1\_2;

{$APPTYPE CONSOLE}

{$R \*.res}

Var n, k: integer;

x,f,Ffirst,Sum:real;

begin

x:=0.6;

while x<=1.111 do // loop C

begin

Sum:=0;

for k := 1 to 9 do // loop A

begin

Sum:=Sum+((exp(2/3-k))/(sqrt(exp(ln(x)\*k))\*exp(ln(x)\*(2\*k-1))));

end;

for n := 10 to 15 do // loop B

begin

Ffirst:=(exp(ln((1+(sin(pi\*x/n)\*sin(pi\*x/n))))/3));

Sum:=Sum+((exp(2/3-k))/(sqrt(exp(ln(x)\*k))\*exp(ln(x)\*(2\*k-1))));

f:=Ffirst\*Sum;

writeln('n = ':1 ,n,'x = ':10,x:2:2,'f = ':11, f:2:10);

end;

x:=x+0.25;

writeln;

end;

readln;

end.

Текст программы

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| Мониторинг результата вычислений | 1 |  | n = 10 x = 0.60 f = 73.1241242536  n = 11 x = 0.60 f = 91.8227729353  n = 12 x = 0.60 f = 110.4984305955  n = 13 x = 0.60 f = 129.1598711786  n = 14 x = 0.60 f = 147.8124825118  n = 15 x = 0.60 f = 166.4596539676  n = 10 x = 0.85 f = 2.0830289660  n = 11 x = 0.85 f = 2.0797641082  n = 12 x = 0.85 f = 2.0782389615  n = 13 x = 0.85 f = 2.0779662689  n = 14 x = 0.85 f = 2.0786169013  n = 15 x = 0.85 f = 2.0799624839  n = 10 x = 1.10 f = 0.9068590983  n = 11 x = 1.10 f = 0.9016174525  n = 12 x = 1.10 f = 0.8975405673  n = 13 x = 1.10 f = 0.8943129299  n = 14 x = 1.10 f = 0.8917175113  n = 15 x = 1.10 f = 0.8896015831 | Тест пройден |

* 1. Итоговый текст программы

program laba2zadanie1\_2;

{$APPTYPE CONSOLE}

{$R \*.res}

Var n, k: integer;

x,f,Ffirst,Sum:real;

begin

x:=0.6;

while x<=1.111 do // loop C

begin

Sum:=0;

for k := 1 to 9 do // loop A

begin

Sum:=Sum+((exp(2/3-k))/(sqrt(exp(ln(x)\*k))\*exp(ln(x)\*(2\*k-1))));

end;

for n := 10 to 15 do // loop B

begin

Ffirst:=(exp(ln((1+(sin(pi\*x/n)\*sin(pi\*x/n))))/3));

Sum:=Sum+((exp(2/3-k))/(sqrt(exp(ln(x)\*k))\*exp(ln(x)\*(2\*k-1))));

f:=Ffirst\*Sum;

writeln('n = ':1 ,n,'x = ':10,x:2:2,'f = ':11, f:2:10);

end;

x:=x+0.25;

writeln;

end;

readln;

end.

Текст программы