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

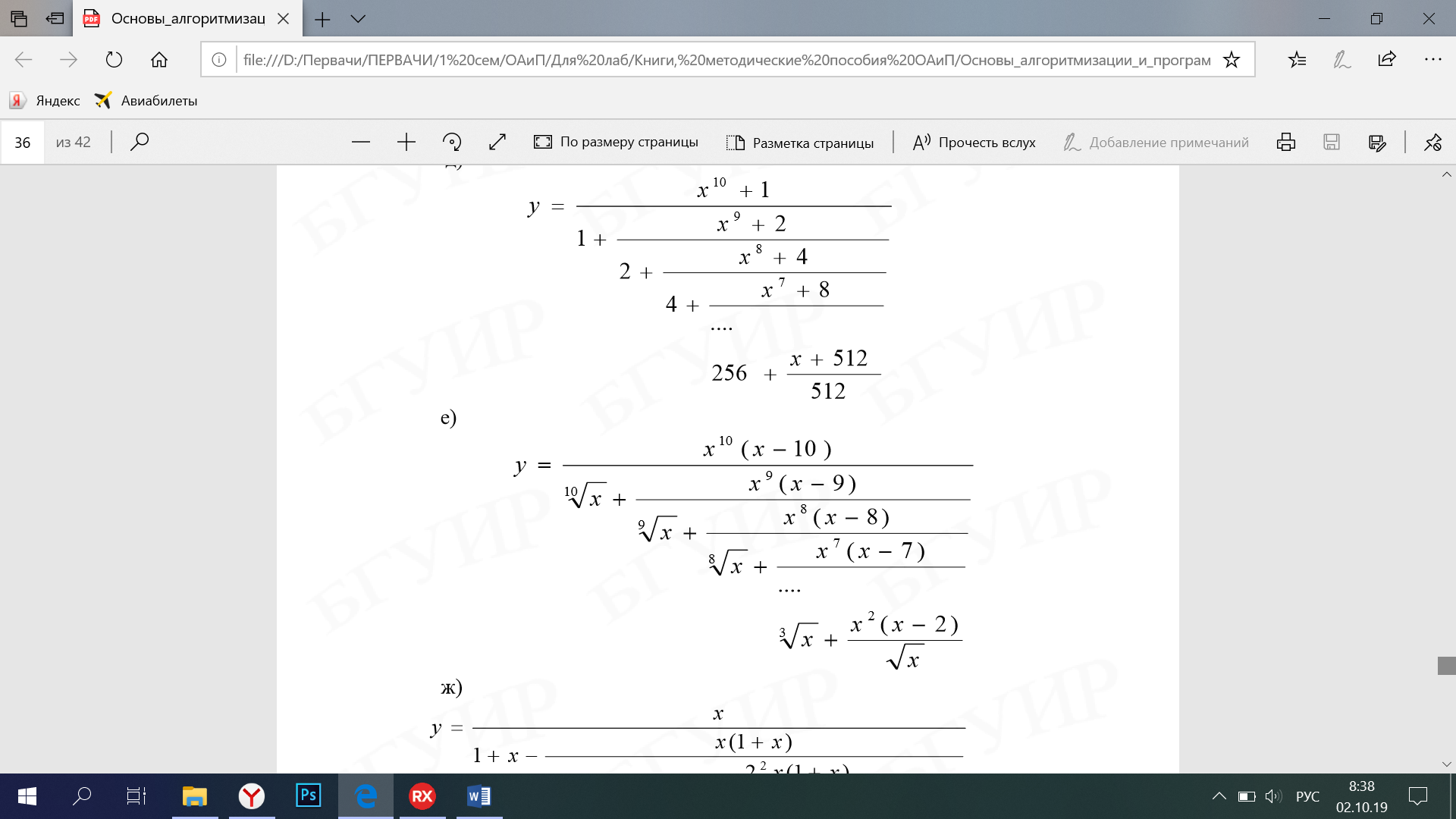
виды:

1) циклы с параметром;

2) итерационные циклы.

1. Задание на лабораторную работу
   1. Постановка задачи

Вычислить цепную дробь для Х=0.5(0.05)0.8



* 1. Эскиз ожидаемого результата

x:0.50 y: -0.01014569503604

x:0.55 y: -0.02669848849392

x:0.60 y: -0.06805241829035

x:0.65 y: -0.15882155001111

x:0.70 y: -0.58579988394734

x:0.75 y: 0.27703193734811

x:0.80 y: -0.40317499943169

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 | 2 | n+1 | Цикл B |
| X | Текущее значение аргумента | real | 0.5 | x + 0.05 | Циклы А, В |
| Y | Переменная для подсчета функции | real | 0 | (exp(ln(x) \* n) \* (x - n)) / (exp(ln(x) / n) + y) | Цикл В |

* 1. Разработка алгоритма

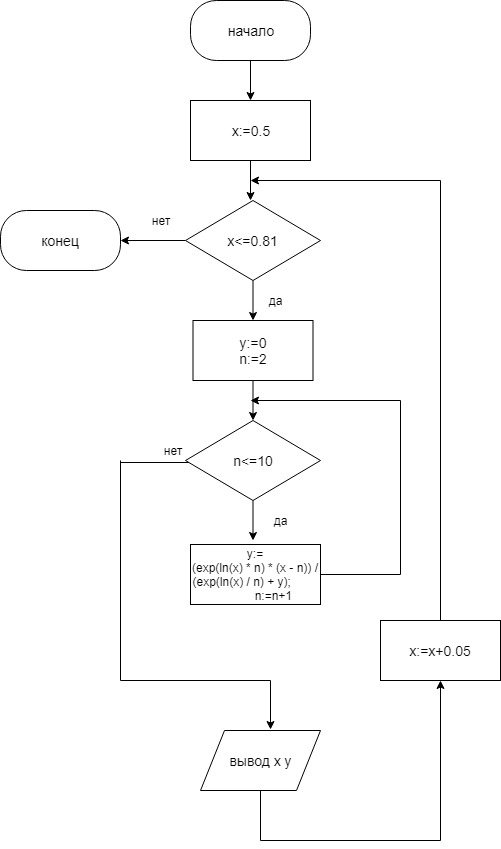


Схема работы программы

Рисунок 3.1 – Схема работы программы

* 1. Текст программы и его описание

program Project1;

{$APPTYPE CONSOLE}

{$R \*.res}

var

n, i: integer;

y, x: real;

begin

x := 0.5;

while x <= 0.81 do // loop A

begin

y := 0;

n := 2;

while n <= 10 do // loop B

begin

y := (exp(ln(x) \* n) \* (x - n)) / (exp(ln(x) / n) + y);

n := n + 1;

end;

writeln;

writeln('x:', x:3:2, ' y: ', y:15:14);

x := x + 0.05;

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 |  | x:0.50 y: -0.01014569503604  x:0.55 y: -0.02669848849392  x:0.60 y: -0.06805241829035  x:0.65 y: -0.15882155001111  x:0.70 y: -0.58579988394734  x:0.75 y: 0.27703193734811  x:0.80 y: -0.40317499943169 | Тест  пройден |

* 1. Итоговый текст программы

program Project1;

{$APPTYPE CONSOLE}

{$R \*.res}

var

n, i: integer;

y, x: real;

begin

x := 0.5;

while x <= 0.81 do // loop A

begin

y := 0;

n := 2;

while n <= 10 do // loop B

begin

y := (exp(ln(x) \* n) \* (x - n)) / (exp(ln(x) / n) + y);

n := n + 1;

end;

writeln;

writeln('x:', x:3:2, ' y: ', y:15:14);

x := x + 0.05;

end;

readln;

end.

Текст программы