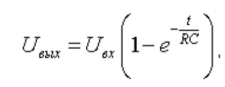
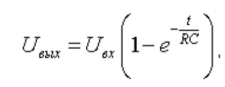
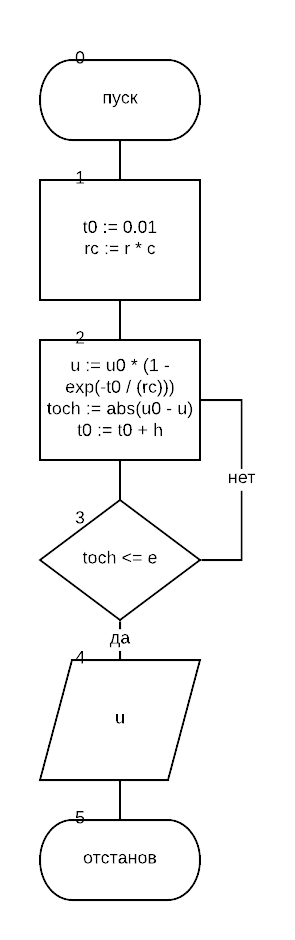
# Лабораторная работа № 8

## 1 задание

1. Итерационные циклические вычислительные процессы с управлением по индексу и функции.
2. Научится реализовать алгоритм для вычисление итерационных циклических вычислительных процессов с управлением по индексу и функции средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. Дан процесс, связанный с изменением выходного напряжения Uвых на обкладках конденсатора электрической цепи, которая включает активное сопротивление R = 2 Ом и конденсатор с емкостью С = 0.01 Ф. Построить переходную характеристику заряда конденсатора по схеме RC цепочки с заданной точностью ε = 10-3, Uвх = 50 В:  
     
   начальное значение t = 0.01, с шагом 0.01.
5. 
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| u | Выходное напряжение | real |
| t0 | Начальное значение | real |
| e | Точность | real |
| u0 | Входное напряжение | real |
| h | Шаг | real |
| r | Сопротивление | real |
| c | Ёмкость конденсатора | real |
| rc | Временная переменная | real |

**program** pr8;

**const**

e = 0.001; u0 = 50; h = 0.01; r = 2; c = 0.01;

**var**

u, t0, toch, rc: real;

**begin**

t0 := 0.01;

rc := r \* c;

**repeat**

u := u0 \* (1 - exp(-t0 / (rc)));

toch := abs(u0 - u);

t0 := t0 + h;

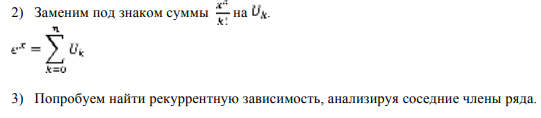
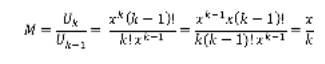
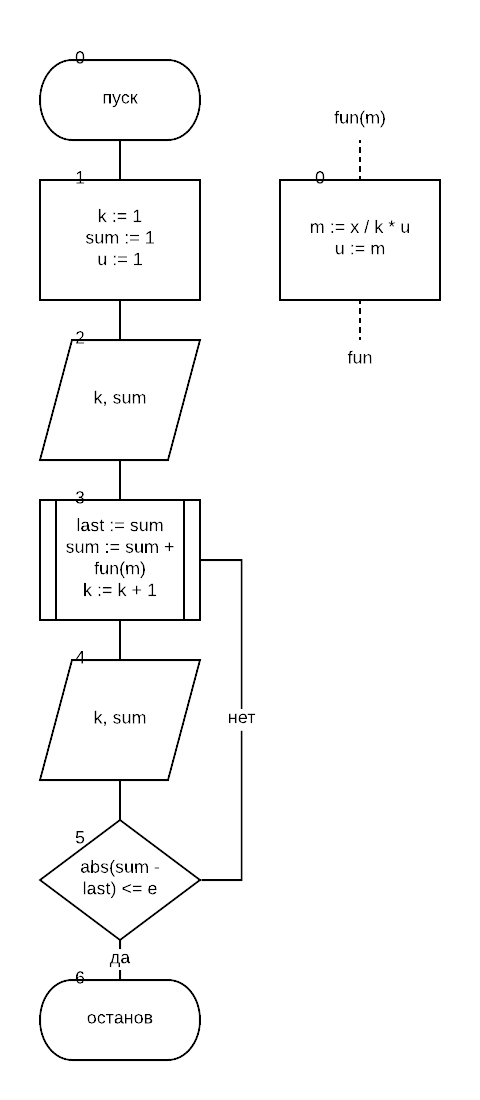
**until** toch <= e;

writeln(u)

**end**.

1. 
2. Для упрощения реализации программы ввёл промежуточную переменную rc.
3. Реализовал программу для подсчёта выходного напряжения.

## 2 задание

1. Итерационные циклические вычислительные процессы с управлением по индексу и функции.
2. Научится реализовать алгоритм для вычисление итерационных циклических вычислительных процессов с управлением по индексу и функции средствами компилятора и языка программирования Pascal.
3. ПК, FreePascal 3.0.0.
4. Вычислить e(x) с точность 10-4. Начальные условия: k = 1, U0 = 1, S0 = 1, x = 0.5.
5.   
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| e | Точность | real |
| k | Номер члена | integer |
| x | Параметр функции | real |
| m | Элемент суммы | real |
| sum | Сумма | real |
| u | Множитель | real |
| last | Предпоследняя сумма | real |

**program** pr8;

**const**

e = 0.0001; x = 0.5;

**var**

k: integer;

m, sum, u, last: real;

**function** fun(m: real): real;

**begin**

m := x / k \* u;

u := m;

**end**;

**begin**

k := 1; sum := 1; u := 1;

writeln('k = ', k, ', S = ', sum :0:4, '.');

**repeat**

last := sum;

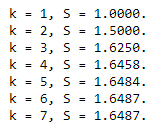
sum := sum + fun(m);

k := k + 1;

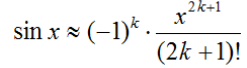
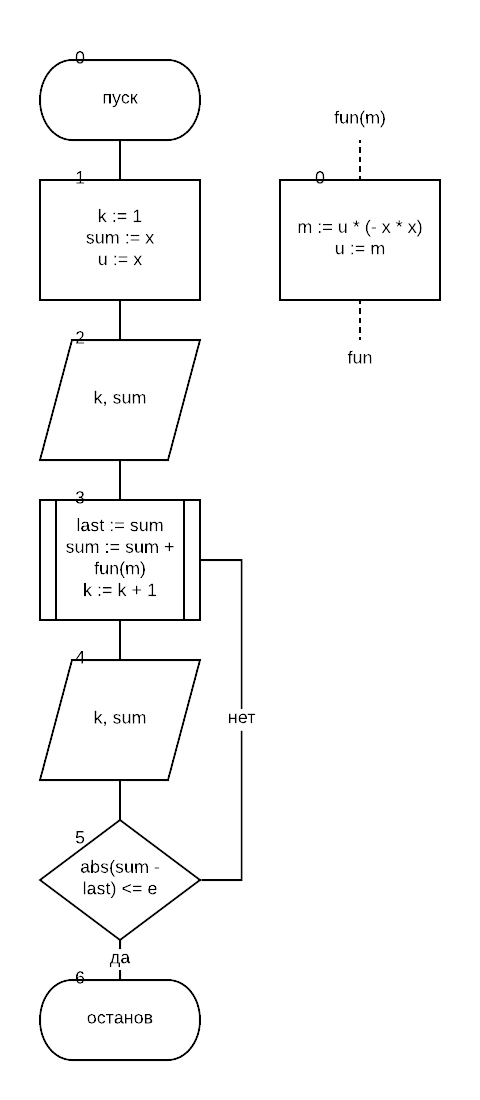
writeln('k = ', k, ', S = ', sum :0:4, '.');

**until** abs(sum - last) <= e

**end**.

1. 
2. Так как в выражении несколько раз высчитывается элемент суммы, то рационально написать для его вычисления отдельную функцию.
3. Вычислил e(x) с точность 10-4.

## 3 задание

1. Итерационные циклические вычислительные процессы с управлением по индексу и функции.
2. Научится реализовать алгоритм для вычисление итерационных циклических вычислительных процессов с управлением по индексу и функции средствами компилятора и языка программирования Pascal.
3. ПК, FreePascal 3.0.0.
4. Вычислить Sin(x) с точностью 10-4. Начальные условия: k = 1, U0 = x, S0 = x, x = π/6. 
5. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| e | Точность | real |
| k | Номер члена | integer |
| x | Параметр функции | real |
| m | Элемент суммы | real |
| sum | Сумма | real |
| u | Множитель | real |
| last | Предпоследняя сумма | real |

**program** pr8;

**const**

e = 0.0001; x = Pi/6;

**var**

k: integer;

m, sum, u, last: real;

**function** fun(m: real): real;

**begin**

m := u \* (- (x \* x));

u := m;

**end**;

**begin**

k := 1; sum := x; u := x;

writeln('k = ', k, ', S = ', sum :0:4, '.');

**repeat**

last := sum;

sum := sum + fun(m);

k := k + 1;

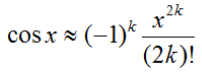
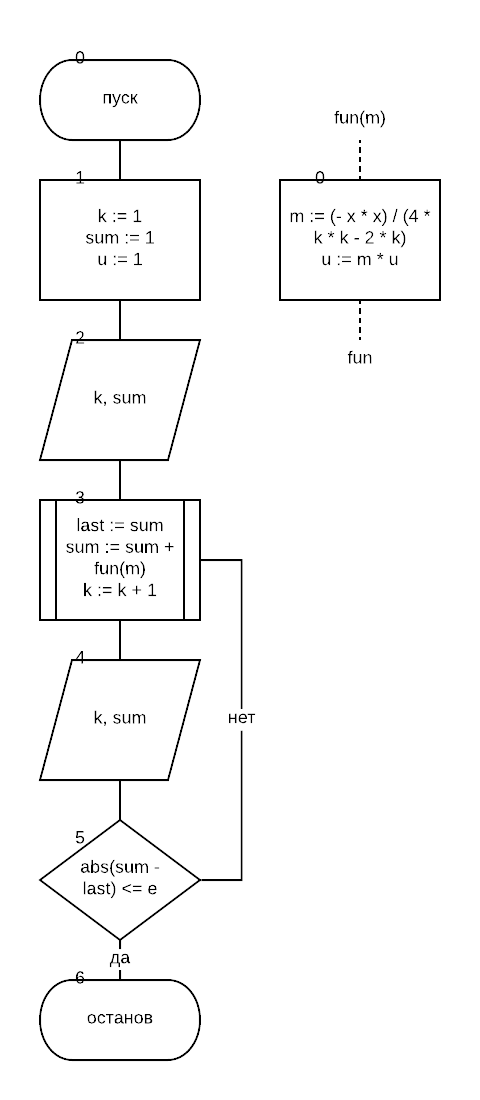
writeln('k = ', k, ', S = ', sum :0:4, '.');

**until** abs(sum - last) <= e

**end**.

1. 
2. Так как в выражении несколько раз высчитывается элемент суммы, то рационально написать для его вычисления отдельную функцию.
3. Вычислил Sin(x) с точностью 10-4.

## 4 задание

1. Итерационные циклические вычислительные процессы с управлением по индексу и функции.
2. Научится реализовать алгоритм для вычисление итерационных циклических вычислительных процессов с управлением по индексу и функции средствами компилятора и языка программирования Pascal.
3. ПК, FreePascal 3.0.0.
4. Вычислить Cos(x) с точностью 10-4. Начальные условия: k = 1, U0 = 1, S0 = 1, x = π/6. 
5. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| e | Точность | real |
| k | Номер члена | integer |
| x | Параметр функции | real |
| m | Элемент суммы | real |
| sum | Сумма | real |
| u | Множитель | real |
| last | Предпоследняя сумма | real |

**program** pr8;

**const**

e = 0.0001; x = Pi/6;

**var**

k: integer;

m, sum, u, last: real;

**function** fun(m: real): real;

**begin**

m := (- x \* x) / (4 \* k \* k - 2 \* k);

u := m \* u;

**end**;

**begin**

k := 1; sum := 1; u := 1;

writeln('k = ', k, ', S = ', sum:0:4, '.');

**repeat**

last := sum;

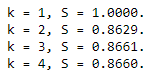
sum := sum + fun(m);

k := k + 1;

writeln('k = ', k, ', S = ', sum:0:4, '.');

**until** abs(sum - last) <= e

**end**.

1. 
2. Так как в выражении несколько раз высчитывается элемент суммы, то рационально написать для его вычисления отдельную функцию.
3. Вычислил Cos(x) с точностью 10-4.