Федеральное государственное автономное  
образовательное учреждение высшего образования  
"Национальный исследовательский университет   
"Высшая школа экономики"

*Факультет социально-экономических и компьютерных наук*

Косвинцев Степан Константинович

**ЛАБОРАТОРНАЯ РАБОТА № 2**

студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.03.04 Программная инженерия*

Руководитель  
Дацун Н.Д.

Пермь, 2024 год

Оглавление

[1 Общая постановка задачи 3](#_Toc24542)

[2 Постановка задачи для варианта 11 3](#_Toc25355)

[3 Анализ задачи 3](#_Toc18133)

[4 Алгоритм 3](#_Toc23091)

[5 Программа 12](#_Toc4059)

[6 Результаты работы программы 12](#_Toc23501)

[ПРИЛОЖЕНИЕ А Текст программы 13](#_Toc15523)

Задача 1

Требуется построить рекурсивную формулу и вычислить функции для заданных значений (см. Таблицу 1).

Таблица 1.Функции и аргументы

|  |  |
| --- | --- |
| Функция | Аргументы |
| Add(x, y) = x + y | *x*= 4, *y*= 2 |
| *Mult*(*x*,*y*)  = *x* × *y* | *x*= 4, *y*= 2 |
| *Power*(*x*,*y*)  = *x**y* | *x*= 4, *y*= 2 |

Вспомогательная функция Decr

Decr(x)=x-1

Задается как обратная Inc с помощью минимизации

Decr(x)=μy(Inc(y)=x)

Add

Рекурсивная формула

С помощью суперпозиции и рекурсии построим функцию сложения из функция Inc и Decr.

Add(x,y)=x+1+1…+1+1=(x+1+1…+1)+1=Add(x,y-1)+1=Inc(Add(x,y-1))

Add(x,0)=x

Add(x,y)=Inc(Add(x, Decr(y)))

Add(4,2)

Результат равен 6. Дерево вызовов представленно на рисунке 1.

Рисунок 1. Вычисление Add(4,2)

Программа

#include <iostream>

int AddIter(int x, int y) {

int s = x;

for (int i = 0; i < y; i++) {

s++;

}

return s;

}

int AddReq(int x, int y) {

if (y == 0) {

return x;

}

return AddReq(x + 1, y - 1);

}

int main()

{

std::cout << AddIter(4,2)<<' '<<AddReq(4,2)<<'\n';

}

Пошаговое выполнение

2 Постановка задачи для варианта 11

Заданна функция 

Её разложение в ряд Маклорена  (1)

Количество слагаемых n=10.

3 Анализ задачи

Формула члена суммы  где n=0,1,2, ... .

Факториал нуля считается равным единице.

Член суммы относится к 3 типу, то есть его можно представитьв виде произведания  где , а .

Первый множитель разумно считать каждый раз заново, а второй -- по рекуррентному соотношению.

4 Алгоритм

Блок схемы алгоритмов представлены на рисунках 1-8.

Схема перебора значений x представлена на рисунке 1.

Схема вычислений с конкретным x представлена на рисунке 2.

Схема вычисления точного значения функции представлена на рисунке 3.

Схема приближенного вычисления с заданной точностью представлена на рисунке 4.

Схема приближенного вычисления с заданным количеством итераций представлена на рисунке 5.

Схема создания первого слагаемого представлена на рисунке 6.(Тут под слагаемым подразумеваются его характеристики, а не значение).

Схема создания следующего слагаемого из предыдущего представлена на рисунке 7.(Тут под слагаемым подразумеваются его характеристики, а не значение).

Характеристики слагаемого -- его индекс(0,1,2,...), значение нереккурентного и реккурентного множителей.

Схема вычисления значения слагаемого представлена на рисунке 8.

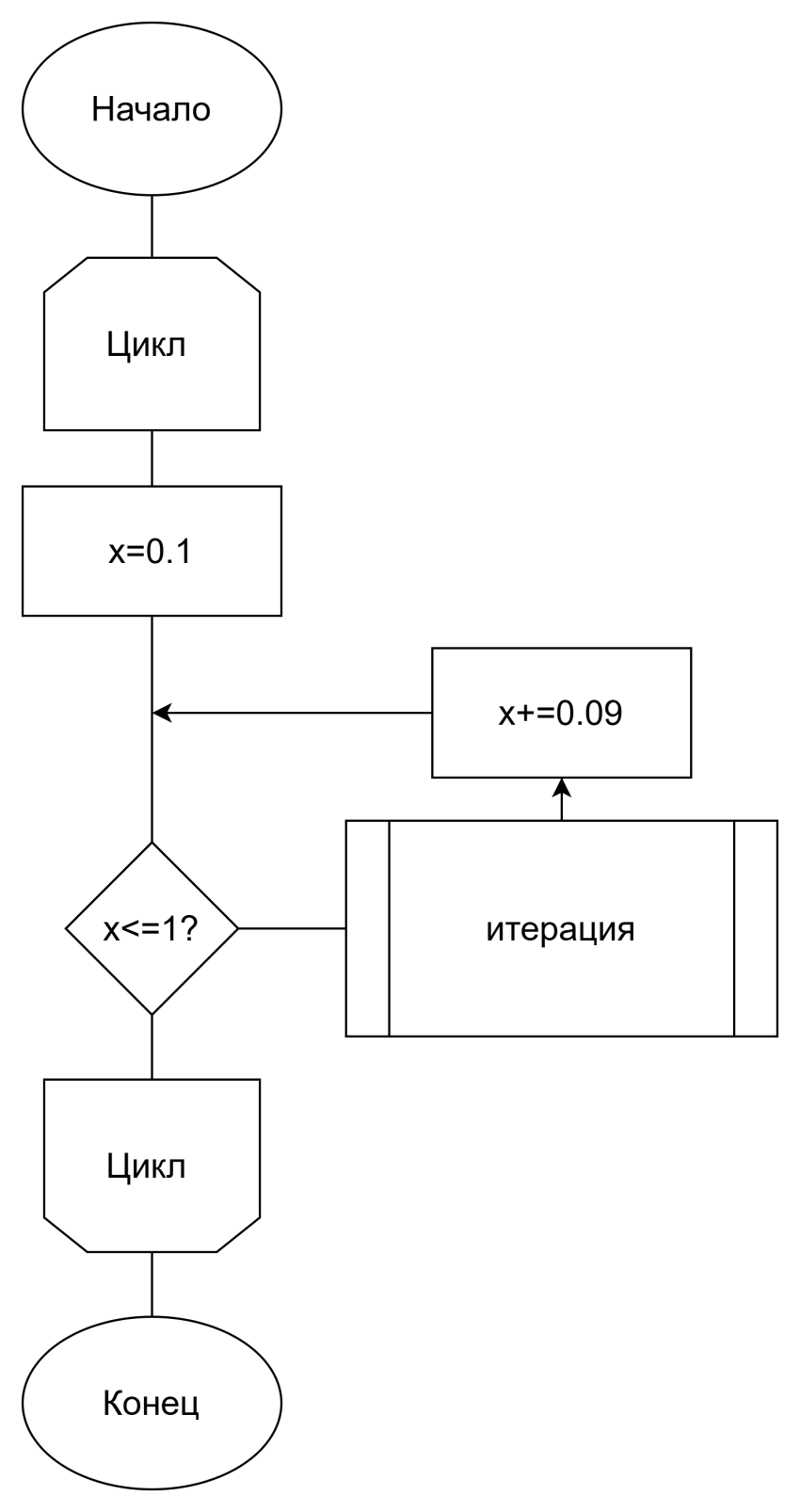


Рисунок 1. Общая схема алгоритма

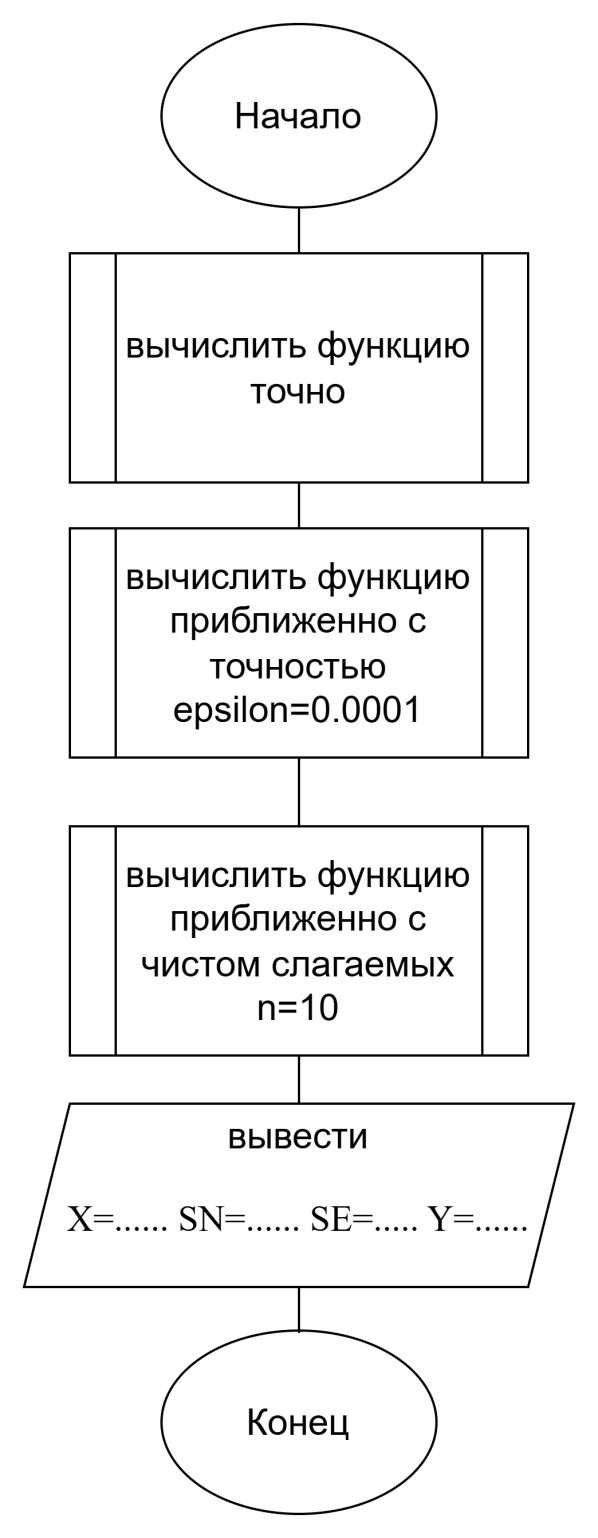


Рисунок 2. Схема итерации

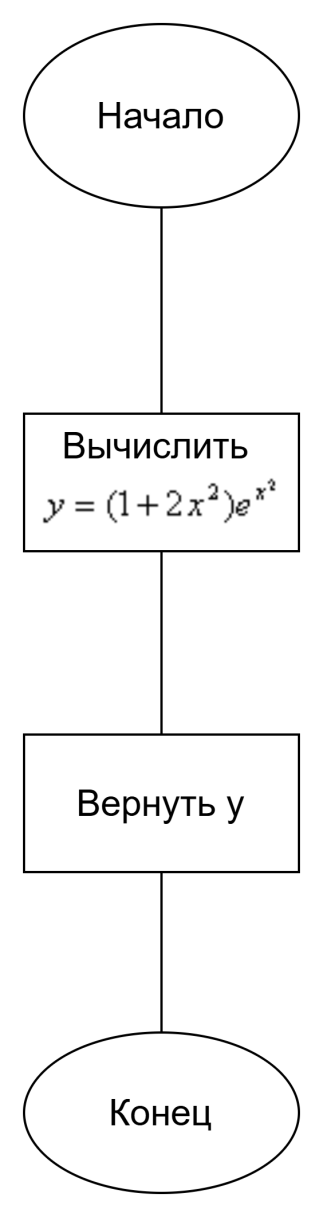


Рисунок 3. Схема вычисления точного значения функции

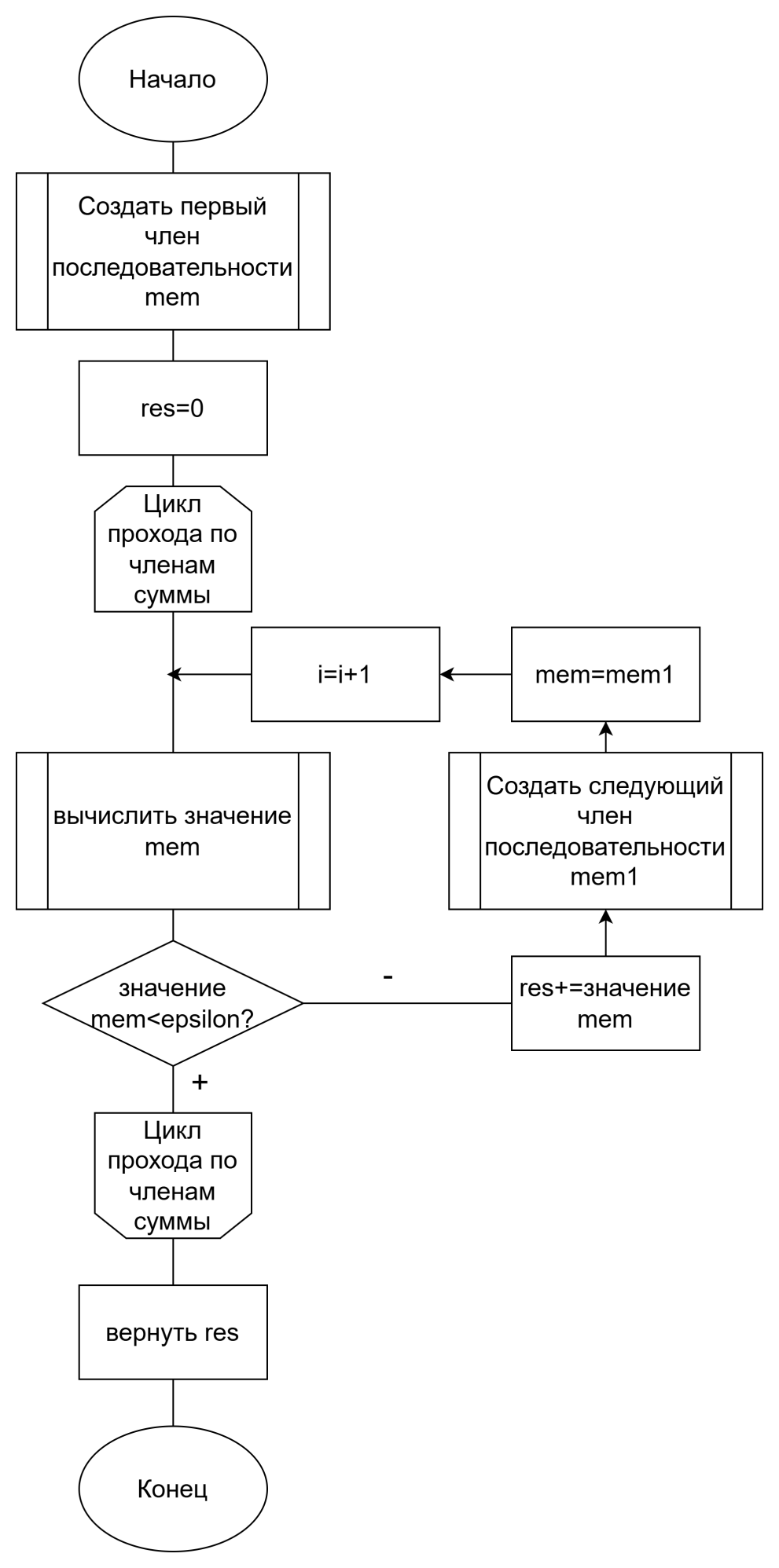


Рисунок 4. Схема приближенного вычисления с заданной точностью

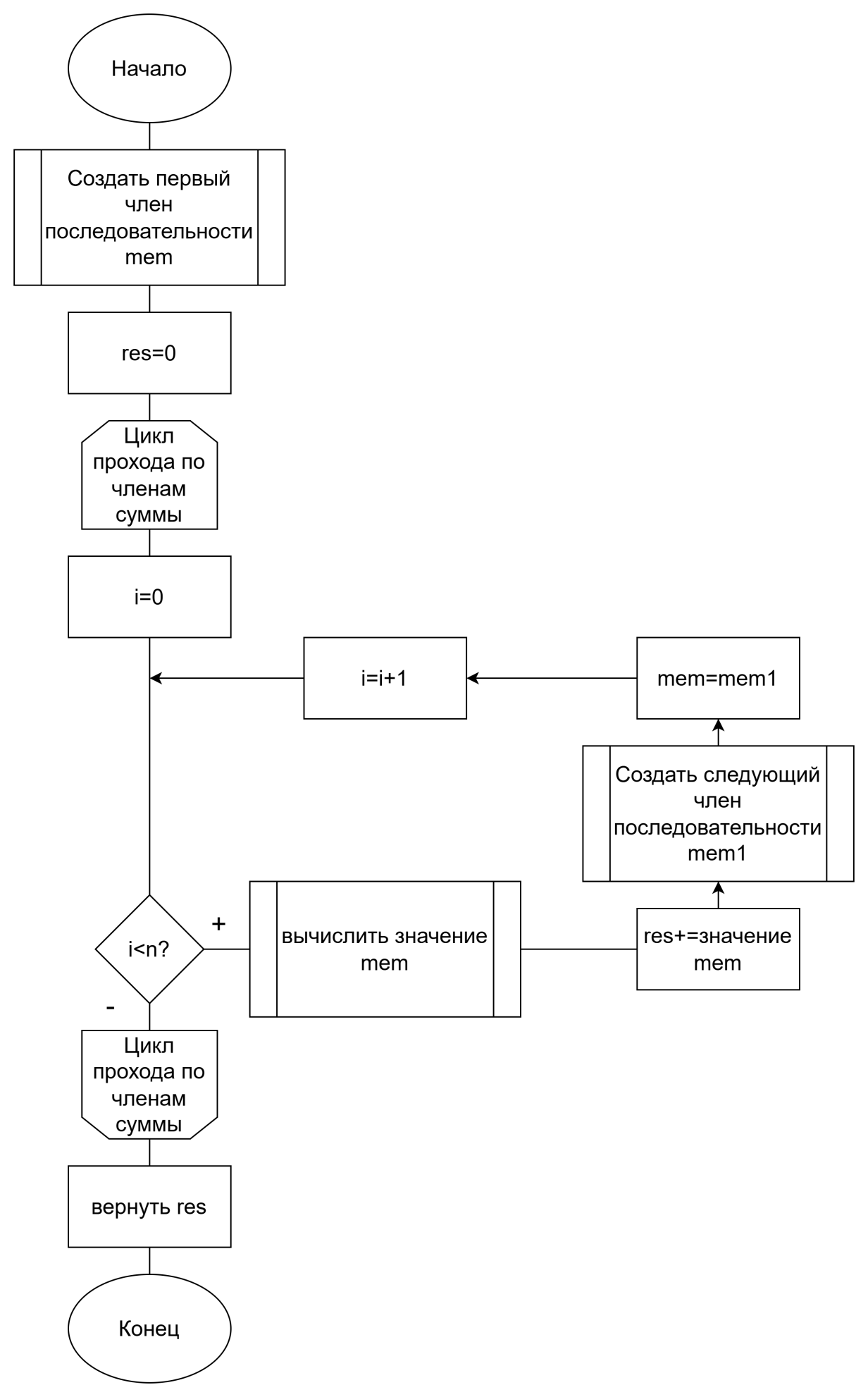


Рисунок 5. Схема приближенного вычисления с заданным количеством слагаемых

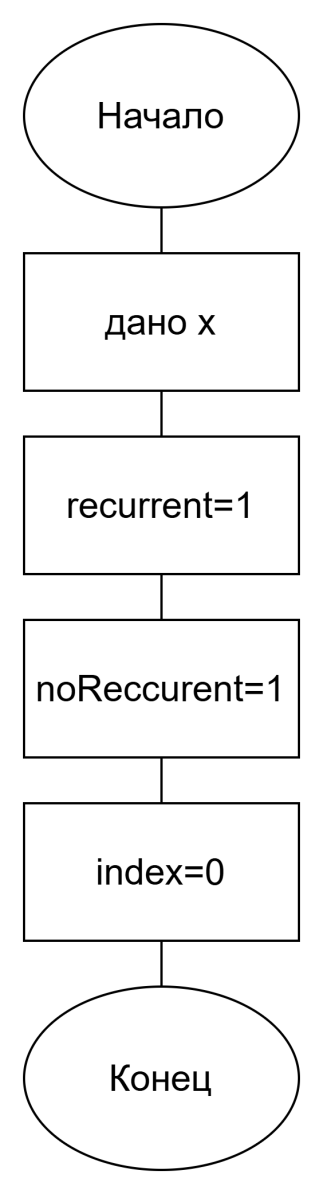


Рисунок 6. Схема вычисления первого слагаемого

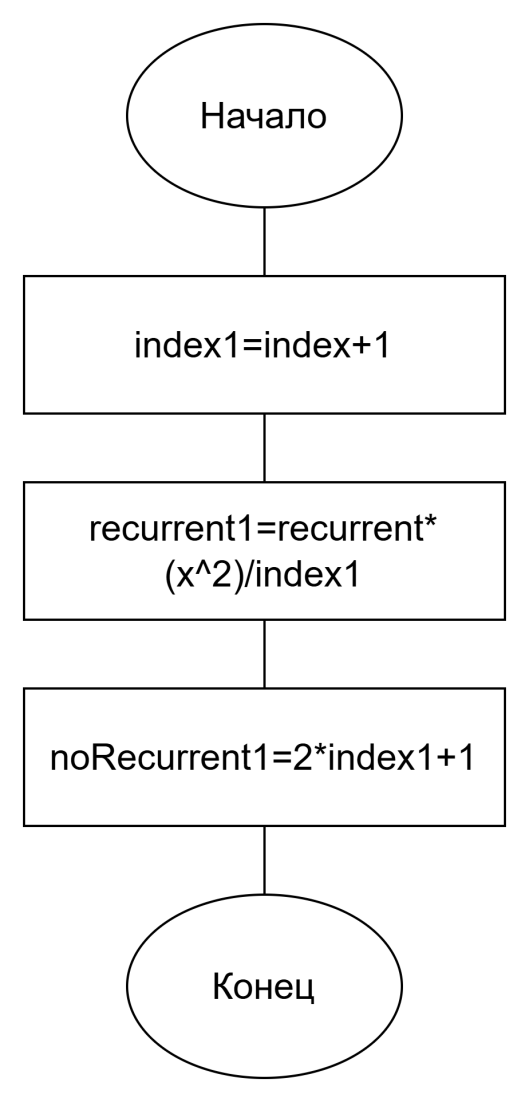


Рисунок 7. Схема вычисления следующего слагаемого



Рисунок 8. Схема вычисления значения слагаемого

5 Программа

Текст программы представлен в Приложении А.

Вынос характеристик слагаемого разложения Маклорена в один объект класса SeriesMember обоснован тем, что они используются вместе и описывают одну сущность.

Код вычисления первого слагаемого, следующего слагаемого, значения слагаемого, точного значения функции, приближенного с точностью, приближенного с количеством слагаемых вынесен, соответственно, в функции First, Next, Value, Func, WithAccuracy, WithLenght. Это сделано, чтоб избежать дублирования кода и увеличить читабельность.

6 Результаты работы программы

Результаты работы программы представлены в таблице 1.

Таблица 1. Результаты работы программы.

|  |  |  |  |
| --- | --- | --- | --- |
| X | SN | SE | Y |
| 0,1 | 1,0302511704258515 | 1,0302511666666667 | 1 |
| 0,19 | 1,1116135544051267 | 1,1116129118611668 | 1,1116135544051267 |
| 0,28 | 1,2511430486909287 | 1,2511427729296043 | 1,251143048690929 |
| 0,37 | 1,460683619919616 | 1,4606834983256964 | 1,4606836199196298 |
| 0,45999999999999996 | 1,7585820964987346 | 1,7585804175225566 | 1,7585820964997985 |
| 0,5499999999999999 | 2,1719464706177027 | 2,171945750121244 | 2,1719464706559815 |
| 0,6399999999999999 | 2,740106651976622 | 2,740100544148256 | 2,740106652778428 |
| 0,7299999999999999 | 3,5198471218862477 | 3,519844198248773 | 3,5198471331704244 |
| 0,8199999999999998 | 4,593306269489299 | 4,593304798397038 | 4,593306386609249 |
| 0,9099999999999998 | 6,079949395200671 | 6,079939807032784 | 6,079950350958813 |
| 0,9999999999999998 | 8,154839065255725 | 8,154839065255725 | 8,154845485377129 |

ПРИЛОЖЕНИЕ А   
Текст программы

internal class Program

{

    static double Func(double x)

    {

        return (1+2\*x\*x)\*Math.Pow(Math.E,x\*x);

    }

    static double WithAccuracy(double x, double epsilon){

        SeriesMember mem=SeriesMember.First(x);

        double res=mem.Value();

        while(Math.Abs(mem.Value())>epsilon){

            mem=mem.Next();

            res+=mem.Value();

        }

        return res;

    }

    static double WithLenght(double x, int n){

        SeriesMember mem=SeriesMember.First(x);

        double res=0;

        for(int i=0; i<n; i++){

            res+=mem.Value();

            mem=mem.Next();

        }

        return res;

    }

    private static void Main(string[] args)

    {

        double epsilon=0.0001;

        int n=10;

        double left=0.1;

        double right=1;

        int k=10;

        double step=(right-left)/k;

        for(double x=left; x<=right;x+=step)

        {

            Console.WriteLine($"X={x}   SN={WithLenght(x,n)}    SE={WithAccuracy(x,epsilon)}    Y={Func(x)}");

        }

    }

    class SeriesMember

    {

        public int index;

        public double x;

        public double noRecurrent;

        public double recurrent;

        SeriesMember(int \_index, double \_x, double \_noRecurrent, double \_recurrent)

        {

            index=\_index;

            x=\_x;

            noRecurrent=\_noRecurrent;

            recurrent=\_recurrent;

        }

        public static SeriesMember First(double x)

        {

            return new SeriesMember(0, x, 1, 1);

        }

        public SeriesMember Next()

        {

            int n=this.index+1;

            return new SeriesMember(

                n,

                this.x,

                2\*n+1,

                this.recurrent\*this.x\*this.x/n

            );

        }

        public double Value(){

            return noRecurrent\*recurrent;

        }

    }

}