МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ГРАЖДАНСКОЙ АВИАЦИИ  
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

**ЛАБОРАТОРНАЯ РАБОТА № 2 (2020)**

**ОСНОВЫ РАБОТЫ С КЛАССАМИ В С++**

По дисциплине: «Объектно-ориентированное программирование»

Вариант 9

Выполнил:

студент группы ПМ4-1

Фейзуллин К.М.

***Цель работы.***

Целью данной лабораторной работы является освоение синтаксиса языка С++, включая:

Целью данной лабораторной работы является освоение:

- определения класса и создания объекта;

- использования в классе функций-членов;

- конструкторов и деструкторов, в том числе параметрических;

- использования встроенных функций,

- массивов объектов.

***Задание на выполнение лабораторной работы.***

Разработать программу, которая в соответствии со своим вариантом позволяет:

1. Создать сlass и объект, осуществляющий вычисление значения функции на основе разложения (в сходящийся ряд) с использованием оператора цикла while.

2. Создать сlass и объект с конструктором (для инициализации исходных значений) и деструктором, осуществляющий вычисление значения функции на основе разложения (в сходящийся ряд) с использованием оператора цикла while.

3. Создать сlass и объект с конструктором с параметром (для инициализации исходных значений) и деструктором, осуществляющий вычисление значения функции тремя способами. Использовать встроенную функцию. Вычислить N значений функции тремя способами с заданным шагом и заданной точностью.

4. Выполнить п.3 с использованием массива объектов.

Все вычисления значений функции должны находиться в объектах.

Исходные данные должны быть подобраны таким образом, чтобы можно было проверить правильность работы программы.

**Варианты заданий на выполнение работы - задание 3 лабораторной работы №1**

***Задание 3:***



***Код программы***

﻿#include <iostream>

#include <cmath>

#include <stdlib.h>

#include <fstream>

using namespace std;

class FindSeqOfSinX

{

private:

double \_x;

double \_eps;

int \_n = 0;

int \_rn = 0;

int \_recurn = 0;

bool \_flag\_n = false;

double FuncForSeq(int n);

int Factorial(int n);

double FuncRecurnt(int n);

public:

FindSeqOfSinX()

{

\_x = 0.5;

\_eps = 1e-6;

}

FindSeqOfSinX(double x)

{

\_x = x;

\_eps = 1e-6;

}

FindSeqOfSinX(double x, double eps)

{

\_x = x;

\_eps = eps;

}

FindSeqOfSinX(double x, int n)

{

\_x = x;

\_n = n;

\_flag\_n = true;

}

~FindSeqOfSinX() {

cout << "\nУдаление объекта с Х = " << \_x << "; Точность = " << \_eps << endl;

};

double CalcSeqDefault();

double CalcSeqRec();

double CalcSeqRecurnt();

};

int main()

{

setlocale(LC\_ALL, "RUS");

/\*

cout << "Конструктор по умоланию\n";

FindSeqOfSinX find1;

find1.CalcSeqDefault();

cout << "\n======================\n";

cout << "\nКонструктор с заданным x" << endl;

FindSeqOfSinX find2(0.5);

find2.CalcSeqDefault();

cout << "\n======================\n";

cout << "\nКонструктор с заданным x и n" << endl;

FindSeqOfSinX find3(0.5, (int)15);

find3.CalcSeqDefault();

\*/

cout << "\n======================\n";

cout << "\nКонструктор с заданным x и заданной точностью" << endl;

FindSeqOfSinX find4(0.5, 1e-1);

find4.CalcSeqDefault();

cout << "\n======================\n";

cout << "\nКонструктор с заданным x и заданной точностью" << endl;

FindSeqOfSinX find5(0.5, 1e-4);

cout << "Рекурсивная сумма ряда sin(x)^2 для Х = " << 0.5 << " = ";

cout << find5.CalcSeqRec() << endl;

cout << "Математическая функция sin(x)^2 = " << sin(0.5) \* sin(0.5) << endl;

cout << "\n======================\n";

cout << "\nКонструктор с заданным x и заданной точностью" << endl;

FindSeqOfSinX find6(0.5);

find5.CalcSeqRecurnt();

cout << "\n======================\n";

cout << "\nИспользование массива объектов с варьированием X" << endl;

double h = 0.1;

int m = 0.9 / h;

FindSeqOfSinX\*\* finds = new FindSeqOfSinX\*[m];

for(int i = 0; i < m; i++)

{

cout << endl;

finds[i] = new FindSeqOfSinX(i \* h);

finds[i]->CalcSeqDefault();

cout << endl;

}

for(int i = 0; i < m; i++)

{

delete finds[i];

}

delete[] finds;

return 0;

}

int FindSeqOfSinX::Factorial(int n)

{

int res = 1;

if (n >= 2)

{

for (int i = 2; i <= n; i++)

{

res \*= i;

}

}

return res;

}

double FindSeqOfSinX::FuncForSeq(int n)

{

int one = ((n % 2) ? 1 : -1);

return one \* pow(2, 2 \* n - 1) \* pow(\_x, 2 \* n) / Factorial(2 \* n);

}

double FindSeqOfSinX::CalcSeqDefault()

{

double sum = 0, res;

int n = 0;

do

{

n++;

res = FuncForSeq(n);

sum += res;

if (\_flag\_n && (n >= \_n))

{

break;

}

} while (\_flag\_n || (fabs(res) > \_eps));

if (!\_flag\_n)

{

cout << "Сумма ряда Тейлора функции при X = " << \_x << " и точности: " << \_eps << " , и количестве членов = " << n << endl;

}

else

{

cout << "Сумма ряда Тейлора функции при X = " << \_x << " и количестве членов = " << n << endl;

}

cout << sum << endl;

cout << "\nМатематическая функция sin(x)^2 при X = " << \_x << endl

<< sin(\_x) \* sin(\_x) << endl;

return sum;

}

double FindSeqOfSinX::CalcSeqRec() {

\_rn++;

double el = FuncForSeq(\_rn);

if (fabs(el) < \_eps) {

\_rn = 0;

return el;

}

return el + CalcSeqRec();

}

double FindSeqOfSinX::FuncRecurnt(int n) {

double mul = \_x \* \_x;

for (int i = 1; i < n; i++) {

mul \*= -((double)2 \* 2) \* (\_x \* \_x) / (((double)2 \* n - 1) \* ((double)2 \* n));

}

return mul;

}

double FindSeqOfSinX::CalcSeqRecurnt() {

double sum = 0, res;

int n = 0;

do

{

n++;

res = FuncRecurnt(n);

sum += res;

if (\_flag\_n && (n >= \_n))

{

break;

}

} while (\_flag\_n || (fabs(res) > \_eps));

if (!\_flag\_n)

{

cout << "Сумма ряда Тейлора через рекуррентую формулу функции при X = " << \_x << " и точности: " << \_eps << " , и количестве членов = " << n << endl;

}

else

{

cout << "Сумма ряда Тейлора через рекуррентую формулу функции при X = " << \_x << " и количестве членов = " << n << endl;

}

cout << sum << endl;

cout << "\nМатематическая функция sin(x)^2 при X = " << \_x << endl

<< sin(\_x) \* sin(\_x) << endl;

return sum;

}

***Результаты работы программы***



