Тема работы: Цикл с неизвестным числом повторений

Исходные данные:

Разработать программу, вычисляющую значение функции по формуле сходящегося ряда в соответствии с вариантом задания.

При создании программы должны быть выполнены следующие дополнительные условия:

1. Значение функции, к которой сходится сумма ряда, должно вычисляться с погрешностью, задаваемой пользователем.
2. Программа должна фиксировать количество членов ряда, необходимое для достижения заданной точности.

+значение с ф-ции с погр 4 пункта

Формула для задания:

Содержание пояснительной записки:

Курсовая работа содержит в себе такие разделы, как:

“Содержание”,” Введение”,” Заключение”,” Список использованных источников”

**Аннотация**

В данной курсовой работе предстоит разработать программу, вычисляющую значение функции по формуле сходящегося ряда в соответствии с вариантом задания. В программе будет взаимодействие с пользователем. А именно, он сможет вводить нужное значение для x и eps, попадающие под условия формулы. Для этого будут использованные данные, полученные в ходе выполнения лабораторных работ. В итоге пользователь получит программу полностью советующим требованиям.

**Annotation**

In this course work is to develop a program that calculates the value of the function according to the formula of a converging series in accordance with the task option. The program will interact with the user. Namely, he will be able to enter the desired value for x and eps, which fall under the conditions of the formula. For this, the data obtained in the course of laboratory work will be used. As a result, the user will receive the program fully advising requirements.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc27898372)

[1 Описание структуры данных 7](#_Toc27898373)

[2 Описание функций 8](#_Toc27898374)

[3 Описание интерфейса пользователя 9](#_Toc27898375)

[4 Описание алгоритма работы программы 10](#_Toc27898376)

[5 Пример работы программы 11](#_Toc27898377)

[Заключение 14](#_Toc27898378)

[Список используемой литературы 15](#_Toc27898379)

[Приложение Листинг кода 16](#_Toc27898380)

[Приложение 2 19](#_Toc27898381)

# ВВЕДЕНИЕ

Цель курсовой работы – получение практических навыков в технологии управления вычислительными процессами и программировании типовых задач циклической обработки числовых данных.

В ходе выполнения работы мы воспользуемся всеми знаниями, полученными при выполнении лабораторных работ.

Содержание пояснительной записки к курсовой работе соответствует стандартным требованиям к пояснительным документам для программного продукта.

Задачи работы включают выполнения следующих этапов:

1. Ввод переменной Х
2. Ввод погрешности Eps
3. Максимальное число шагов ввод
4. Суммирование элементов ряда по формуле, соответствующей индивидуальному заданию
5. Вывод обработанной формулы на экран

# Описание структуры данных

В данной курсовой работе мы используем типы данных int и double.

Тип данных int – это целочисленный тип данных способный содержать положительные и отрицательные значения используются для представления чисел.

Тип данных double – это тип данных, используемый для хранения отрицательных и положительных чисел с плавающей точкой. Данные типы данных используются и включены в стандартную библиотеку языка C++.

Тип данных string– класс с методами и переменными для организации работы со строками в языке программирования C++. Он включён в стандартную библиотеку C++.

Таблица 1.1- Основные переменные

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | Тип | Байт | Диапазон принимаемых значений |
| Точность | double | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |
| Х |
| Сумма ряда |
| Член ряда |
| Максимальное число итераций | Int | 4 | -2 147 483 648   /   2 147 483 647 |

Таблица 1.2 - Вспомогательные переменные

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | Тип | Байт | Диапазон принимаемых значений |
| Значение log(1+x) для проверки точности расчёта | double | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |
| Количество знаков после запятой в форматировании ответа | Int | 4 | -2 147 483 648   /   2 147 483 647 |
| Количество челнов ряда |
| Буфер вводимых значений для обработки некорректного ввода | String | Динамическое выделение памяти | |

# Описание функций

**Функции стандартных библиотек**

1. **math.h**

Данная стандартная библиотека позволяет использовать математические формулы и вычисления, такие как возведение в степень, взятие корня числа, модуля, вычисление тригонометрических функций sin(x), cos(x), tg(x) и т.д.

В данной работе используется функция log() для вычисления логарифма при проверке работы программы.

1. **iostream**

Поток ввода-вывода. Содержит классы, позволяющие управлять потоками ввода и вывода.

В данной работе были необходимы следующие функции:

1. cin для управления потоком ввода.

Для реализации проверки корректности пользовательского ввода используются atoi() и atof() преобразующие введённые строки к типам int и double соответственно.

1. сout для управления потоком вывода.
2. String для использования в качестве буфера для потока ввода

# Описание интерфейса пользователя

При запуске программы, перед пользователем появляется меню выбора, представленное на рис. 1.

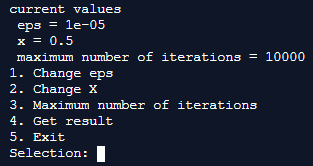


Рисунок 1. Меню программы

Пользователь сразу видит значения переменных, заданные по умолчанию и имеет возможность:

1. Ввести значение для Eps.
   1. При выполнении задачи появляется следующий текст:

“Введите значение для Eps:”

* 1. Пользователь вводит значение и нажимает (return/enter).
  2. Программа проверяет значение, если значение соответствует условию работоспособности программы, то программа выводит пользователя в меню.

1. Ввести собственное значение для X.
   1. При выполнении задачи появляется следующий текст:

“Введите значение для X:”

* 1. Пользователь вводит значение и нажимает (return/enter).
  2. Программа проверяет значение, если значение соответствует условию работоспособности программы, то выводит пользователя в меню.

1. Ввести значение максимального числа итераций.
   1. При выполнении задачи появляется следующий текст:

“Введите максимальное число итераций:”

* 1. Пользователь вводит значение и нажимает (return/enter).
  2. Программа проверяет значение, если значение соответствует условию работоспособности программы, то выводит пользователя в меню.

1. Программа производит расчет и выводит результат
   1. Количество знаков после запятой, используемое для форматирования ответа, рассчитанное на основе требуемой eps.
   2. Использованное количество членов ряда
   3. Сумму ряда
   4. Значение ln(1+x)
   5. Модуль разности ln(1+x) и суммы ряда
   6. Напоминает пользователю заданную им точность для сравнения с получившейся
   7. Последний член ряда
2. Закончить работу с программой

# Описание алгоритма работы программы

Производится объявление переменных и инициализация значений по умолчанию

  double eps=0.00001,x=0.5,sum,chlen\_ryada,test;

  int i, k, max=10000, input;

string buffer;

bool run=true;

Пока переменна run является true производится вывод описания интерфейса и происходит ожидание выбора пользователя

while(run)

{

cout<<"current values\n"<<" eps = "<<eps<<"\n x = "<<x<<"\n maximum number of iterations = "<< max<<"\n";

cout<<"1. Change eps\n";

cout<<"2. Change X\n";

cout<<"3. Maximum number of iterations\n";

cout<<"4. Get result\n";

cout<<"5. Exit\n";

cout<<"Selection: ";

Некорректные введённые данные обрабатываются посредством использования строки-буфера и функции atoi

getline(cin, buffer);

input = atoi(buffer.c\_str());

Выбор пользователя обрабатывается посредством конструкции switch — case, пользователь имеет возможность изменить точность, координату Х, максимальное число итераций.

При выборе пользователем пункта меню "4. Get result\n”:

1. Производится вспомогательный расчёт количества знаков после запятой для представления вывода на экран, основываясь на требуемой точности вычислений.
2. Инициализация первого члена ряда

chlen\_ryada= ryad (1, x);

1. Производится вычисление суммы ряда до достижения необходимой точности с использованием итеративного вычисления k-го члена ряда с учётом максимально допустимого числа итераций.

chlen\_ryada = ryad (k, x);

 sum += chlen\_ryada;

Точность оценивается сравнением абсолютных величин введённого значения eps и изменения суммы, которое оценивается следующим членом ряда.

1. Вывод результатов на экран пользователя.

После окончания выполнения любого пункта пользовательского меню производится последующий вывод описания интерфейса пользователя и происходит ожидание выбора пользователя.

В случае выбора пользователем пункта меню, отсутствующего в описании, выводится сообщение об ошибке и повторное описание интерфейса пользователя с последующим ожиданием выбора пользователя.

# Пример работы программы

Для тестирования работы программы были использованы вспомогательные функции:

1. Функция расчёта члена ряда для проверки корректности расчёта по рекуррентной формуле
2. Функция расчёта ln(1+x) для произведения полученной суммы ряда с действительным значением аппроксимируемой функции

**Пример 1.**

Вводим значения для X и Eps

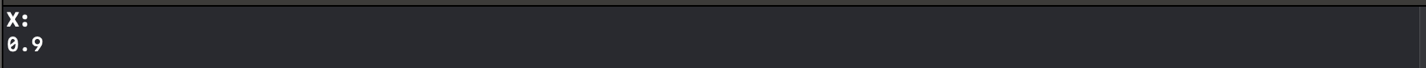


Рисунок 1 – Ввод значения для X



Рисунок 2 – Ввод значения для Eps

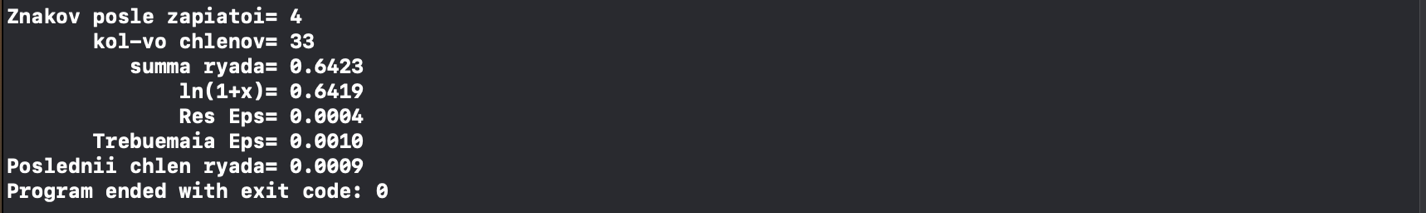


Рисунок 3 – Вывод результата

Заранее указать значение

**Пример 2.**

Вводим значения для X и Eps



Рисунок 4 – Ввод значения для X

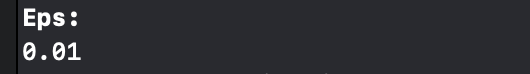


Рисунок 5 – Ввод значения для Eps

Программа выводит результат

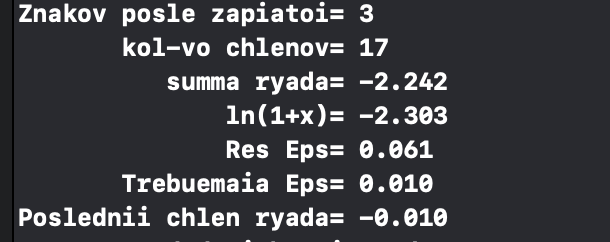


Рисунок 6 – Вывод результата

**Пример 3**

Согласно формуле, значение для X должно быть в диапазон от [-1;1]. Если пользователь вводит значение не попадающие в диапазон, программа будет выдавать следующие.

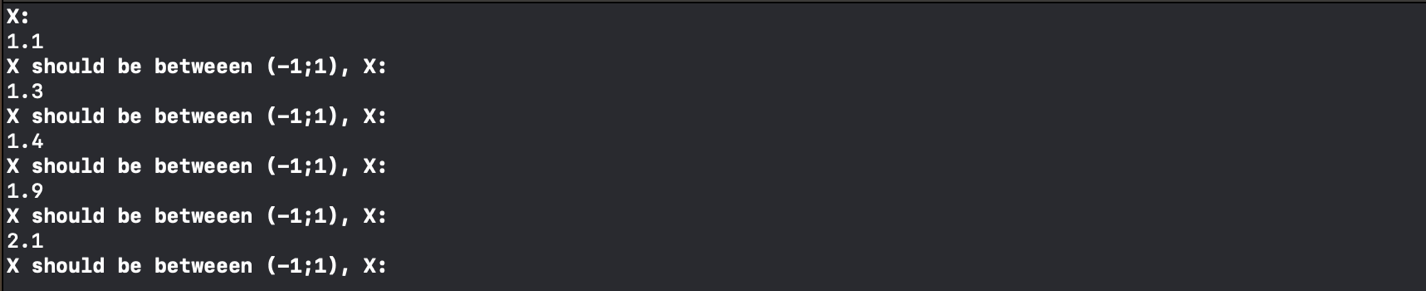


Рисунок 7 – Не корректный ввод X

Так как, точность задается пользователем, и она не может быть нулевой или отрицательной, выполняется проверка.

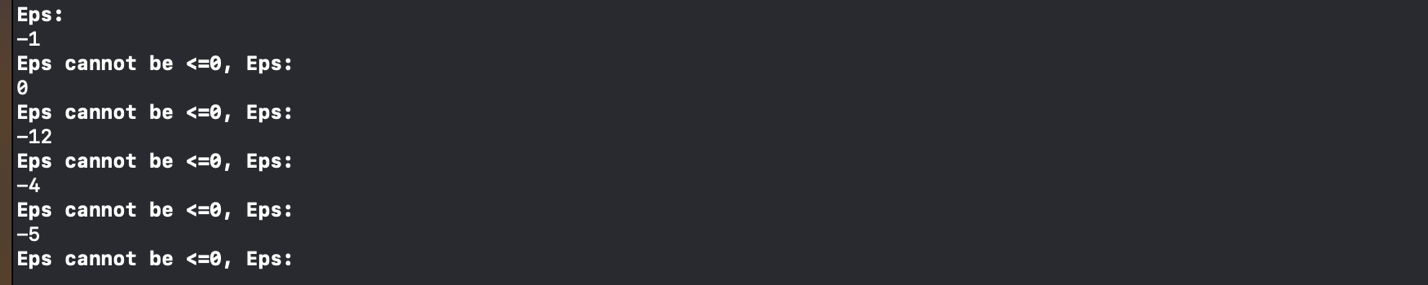


Рисунок 8 – Не корректный ввод Eps

# Заключение

В данной курсовой работе была разработана программа, вычисляющая значение сходящегося ряда в введённой пользователем точке x с задаваемой точностью:

* Реализован интерфейс, позволяющий путём взаимодействия с пользователем многократно использование алгоритмов работы программы за один запуск
* Для уменьшения затрат вычислительной мощности пользовательской машины была выведена рекуррентная формула для членов ряда:

Согласно произведённому тестированию, программа работает полностью корректно и все поставленные задачи выполняются. Принятое приближение подобия абсолютной величины следующего члена ряда и точности вычислений позволяет достаточно хорошо оценить достижение задаваемой точности пользователем, а установленное программное ограничение на максимальное количество членов ряда для суммирования, а также проверка вводимых пользователем значений позволяет обеспечить достаточную отказоустойчивость программного продукта в требуемых ситуациях использования.

В ходе выполнения данной курсовой работы мною были получены навыки реализации программного продукта с неизвестным числом повторений и интерфейса пользователя с возможностью многократного использования программы без перезапуска.

# Список используемой литературы

1. Пахомов, Б. C/C++ и MS Visual C++ 2010 для начинающих / Б. Пахомов. - М.: БХВ-Петербург, 2011. - 736 c.
2. Довбуш, Галина Visual C++ на примерах / Галина Довбуш , Анатолий Хомоненко. - М.: БХВ-Петербург, 2012. - 528 c.
3. Неформальное введение в C++ и Turbo Vision. - Москва: **ИЛ**, **2010**. - 384 c.

+плюс хкод

# Приложение Листинг кода

#include <iostream>

#include <math.h>

using namespace std;

double ryad (int n, float x)

{

  double tmp;

  tmp = pow(x,n)/n;

  if (n % 2 == 0)

    return -1\* tmp;

  else

    return 1 \* tmp;

}

int main()

{

  double eps=0.00001,x=0.5,sum,chlen\_ryada,test;

  int i, k, max=10000, input;

string buffer;

bool run=true;

while(run)

{

cout<<"current values\n"<<" eps = "<<eps<<"\n x = "<<x<<"\n maximum number of iterations = "<< max<<"\n";

cout<<"1. Change eps\n";

cout<<"2. Change X\n";

cout<<"3. Maximum number of iterations\n";

cout<<"4. Get result\n";

cout<<"5. Exit\n";

cout<<"Selection: ";

getline(cin, buffer);

input = atoi(buffer.c\_str());

switch ( input )

{

case 1:

//Ввод точности

  cout << "Eps:\n";

getline(cin, buffer);

eps = atof(buffer.c\_str());

  //Проверка корректности ввода точности

  while (eps <= 0)

  {

    cout << "Eps cannot be <=0, Eps:\n";

getline(cin, buffer);

eps = atof(buffer.c\_str());

  }

break;

case 2:

  //Ввод Х

  cout << "X:\n";

getline(cin, buffer);

x = atof(buffer.c\_str());

  //Проверка корректности ввода Х

  while (x>=1||x<=-1)

   {

    cout << "X should be betweeen (-1;1), X:\n";

getline(cin, buffer);

x = atof(buffer.c\_str());

  }

break;

case 3:

  //Ввод предельного числа циклов

   cout << "Maximum shagov:\n";

getline(cin, buffer);

max = atoi(buffer.c\_str());

  //Проверка корректности ввода

  while (max <= 0 || max >=1000000)

  {

    cout << "max <= 0 || max >=1000000, max:\n";

getline(cin, buffer);

max = atoi(buffer.c\_str());

  }

break;

case 4:

 //Расчёт количества знаков после запятой для вывода на экран на основе заданной точности вычислений

   cout.setf(ios::fixed);

   test = eps;

   for (i = 0; test < 1; i++)

     test \*= 10;

   cout.precision(i + 1);

chlen\_ryada= ryad (1, x);

sum = chlen\_ryada;

   //Вычисление суммы ряда с требуемой точностью

for (k = 2; k < max && abs(chlen\_ryada) >= eps; k++)

{

   chlen\_ryada = ryad (k, x);

 sum += chlen\_ryada;

  }

  test = log(1 + x);

cout << "Result:\n";

 cout << "Znakov posle zapiatoi= " << i+1 << "\n";

 cout << " kol-vo chlenov= " << k-1 << "\n";

cout << " summa ryada= " <<sum<<"\n";

cout << " ln(1+x)= " << test << "\n";

cout << "|ln(1+x)-summa ryada|= " << abs(test-sum) << "\n";

cout << " Trebuemaia Eps= " << eps << "\n";

cout << "Poslednii chlen ryada= " << chlen\_ryada << "\n\n\n";

break;

case 5:

cout<<"Thank you for usage!\n";

run=false;

break;

default:

cout<<"Error, bad input, try again\n";

return 0;

break;

}

}

}

# Приложение 2

