МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 34

РАБОТА ЗАЩИЩЕНА С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | К.А. Жиданов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
|  |
| по дисциплине: ЯЗЫКИ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТЫ ГР. | 3143 |  |  |  | А.С. Медведев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2022

**Цель работы:** изучение методов разработки консольных приложений, способы их запуска и обработки кодов возврата.

**Ход работы:** реализуем функцию вычисления экспоненты на языке программирования C, с помощью разложения в ряд Тейлора.

Используем формулу для разложения элементарных функций(функции экспоненты):

https://www.ok-t.ru/studopediaru/baza9/1239845371934.files/image016.jpg

Для того, чтобы использовать данные функции, мы напишем предварительные функцию возведения числа в степень и функцию для вычисления факториала:

double Stepen(double x, int st)

{

double res;

int i;

res = 1.0;

if (st == 0) {

return 1;

}

else if (st == 1) {

return x;

}

else

for (i = 1; i <= st; i++)

{

res = res \* x;

}

return(res);

}

double Factorial(int n) {

if (n < 2)

return 1;

return n \* Factorial(n - 1);

}

Запишем функцию для вычисления модуля числа:

double Module(double x) {

if (x > 0)

return x;

else return x \* -1;

}

И теперь напишем функцию для вычисления экспоненты в степени через разложение в ряд Тейлора:

double Exponent(double x) {

int k;

double exp;

exp = 0.0;

for (k = 0; k <= 50; k++)

{

exp = exp + (Stepen(x, k ) / Factorial(k));

}

return(exp);

}

Проверим функцию на тестовых значениях. Для проверки сравниваем модуль разности результата функции, и ее значения, подсчитанного на калькуляторе с точностью до одной стотысячной(0,00001). При расхождении будем увеличивать количеством членов в ряду Тейлора(увеличено до 50):

int Test\_exp() {

int r;

r = 0;

r = r || (Module(Exponent(0) - 1.0) >= 0.00001);

r = r || (Module(Exponent(1) - 2.71828) >= 0.00001);

r = r || (Module(Exponent(2) - 7.38905) >= 0.00001);

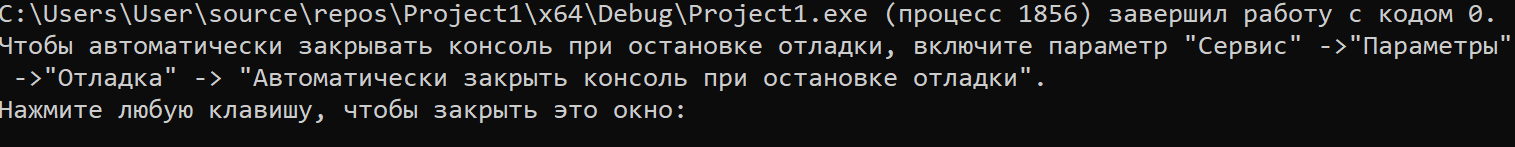
r = r || (Module(Exponent(3) - 20.08553) >= 0.00001);

r = r || (Module(Exponent(4) - 54.59815) >= 0.00001);

return r;

}

После исполнения, программа-отладчик выдала 0, что является показателем того, что экспонента вычисляется правильно.



**Вывод:** Я научился писать функцию, вычисляющую экспоненту с использованием разложения в ряд Тейлора, тем самым изучил методы разработки консольных приложений, способы их запуска и обработки кодов возврата.