ГУАП КАФЕДРА № 34

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
| Язык Си |
| по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |
| --- | --- | --- |
| СТУДЕНТ ГР. № | 3145 17.06.2022 | Щербак. А.А |
|  | подпись, дата | инициалы, фамилия |

Санкт-Петербург 2021

**Задание:** Значение синуса с заданной точностью (ряд Тейлора)

**Цель работы:** изучить методы разработки консольных приложений, способы из запуска и обработки кодов возврата

**Ход работы:**

Для начала подключаем директиву #define. Она определяет идентификатор и последовательность символов, которой будет замещаться данный идентификатор при его обнаружении в тексте программы.

Объявляем функцию (nfabs) и условия для переменной функции. Тоже самое проделываем для вычисления значения синуса.

С помощью цикла while вычисляем значение синуса с помощью ряда Тейлора:

temp = temp \* (-1) \* (x \* x) / (2 \* n) / (2 \* n + 1);

sum += temp;

Продолжаем до тех пор, пока выполняется следующее условие: while (nfabs(temp) > E)

Как только условие не выполняется, выводим получившийся результат

Для того, чтобы удостовериться, что все выполнено правильно, проводим тесты:

r = r || (nfabs(sin\_e(0)) - 0 >= E);

r = r || (nfabs(sin\_e(PI/2)) - 1 >= E);

r = r || (nfabs(sin\_e(PI)) - 0 >= E);

r = r || (nfabs(sin\_e(3\*PI/2)) - 1>= E);

r = r || (nfabs(sin\_e(2\*PI)) - 0 >= E);

r = r || (nfabs(sin\_e(-1 \* PI / 2)) - 1 >= E);

r = r || (nfabs(sin\_e(-1 \* PI)) - 0 >= E);

r = r || (nfabs(sin\_e(-1 \* 3 \* PI /2)) - 1 >= E);

r = r || (nfabs(sin\_e(- 2 \* PI)) - 0 >= E);

r = r || (nfabs(sin\_e(PI / 6)) - 0.5 >= E);

r = r || (nfabs(sin\_e(7\* PI / 6)) - 0.5 >= E);

**Код программы:**

#define E 0.01

#define PI 3.1415

double nfabs(double x){

if (x > 0)

return x;

else return -1 \* x;

}

double sin\_e(double x) {

double temp = x;

double sum = temp ;

int n = 0;

while (nfabs(temp) > E) {

n++;

temp = temp \* (-1) \* (x \* x) / (2 \* n) / (2 \* n + 1);

sum += temp;

}

return sum;

}

int test\_sin() {

int r = 0;

r = r || (nfabs(sin\_e(0)) - 0 >= E);

r = r || (nfabs(sin\_e(PI/2)) - 1 >= E);

r = r || (nfabs(sin\_e(PI)) - 0 >= E);

r = r || (nfabs(sin\_e(3\*PI/2)) - 1>= E);

r = r || (nfabs(sin\_e(2\*PI)) - 0 >= E);

r = r || (nfabs(sin\_e(-1 \* PI / 2)) - 1 >= E);

r = r || (nfabs(sin\_e(-1 \* PI)) - 0 >= E);

r = r || (nfabs(sin\_e(-1 \* 3 \* PI /2)) - 1 >= E);

r = r || (nfabs(sin\_e(- 2 \* PI)) - 0 >= E);

r = r || (nfabs(sin\_e(PI / 6)) - 0.5 >= E);

r = r || (nfabs(sin\_e(7\* PI / 6)) - 0.5 >= E);

//r = r || (nfabs(sin\_e()) - smth >= E);

return r;

}

int main() {

return test\_sin();

}

**Вывод:** мы изучили методы разработки консольных приложений, способы из запуска и обработки кодов возврата. Провели тесты программы и удостоверились в правильности результата