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 |
|  |
| по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3145 |  | 21.04.2022 |  | Д. Д. Лобко |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

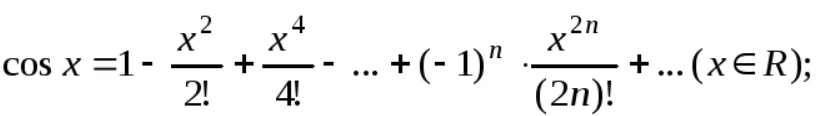
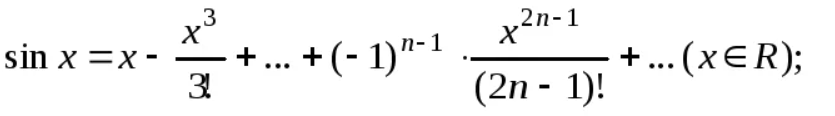
**Цель работы:**

Изучить методы разработки консольных приложений, способы их запуска и обработки кодов возврата.

**Ход работы:**

Реализуем функцию вычисления тангенса на языке программирования Си, с помощью деления синуса на косинус, полученных разложением в ряд Тейлора.

При разложении тригонометрических функций используем эти формулы



Чтобы использовать данные функции, напишем функции для вычисления факториала натурального числа, и функцию возведение числа в степень:

/\*Функция возведения числа в степень\*/

double RaiseToPow(double x, int power)

{

double res;

int i;

res = 1.0;

if (power == 0) {

return 1;

}

else if (power == 1) {

return x;

}

else

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

{

res = res \* x;

}

return(res);

}

/\* Функция нахождения факториала числа \*/

double fact(int k) {

if (k < 2)

return 1;

return k \* fact(k - 1);

}

Далее запишем функции вычисляющие синус и косинус:

|  |  |
| --- | --- |
| /\* Функция нахождения синуса разложением в ряд Тейлора \*/  double sinus(double x) {  int n;  double sin;  sin = 0.0;  for (n = 1; n <= 18; n++)  {  sin = sin + (RaiseToPow(-1, n-1) \* RaiseToPow(x, (2\*n-1)) / fact(2\*n-1));  }    return(sin);  } | /\* Функция нахождения косинуса разложением в ряд Тейлора \*/  double cosinus(double x) {  int n;  double cos;  cos = 1.0;  for (n = 1; n <= 18; n++) {  cos = cos + (RaiseToPow(-1, n) \* RaiseToPow(x, (2 \* n)) / fact(2 \* n));  }  return cos;  } |

Далее проверим функции на тестовых значениях. Для начала напишем функцию модуля числа:

/\* Функция нахождения модуля числа \*/

double fabs(double x) {

if (x > 0)

return x;

else return x \* -1;

}

Для проверки сравним модуль разности результата функции и ее значения, посчитанного на калькуляторе с точностью до одной десятитысячной с одной десятитысячной(первое должно быть меньше). При расхождении будем увеличивать количество членов в ряду Тейлора (увеличил до 18):

|  |  |
| --- | --- |
| /\*Функция для проверки точности вычисления синуса\*/  int test\_sin() {  int r;  r = 0;    r = r || (fabs(sinus(0) - 0.0) >= 0.0001);  r = r || (fabs(sinus(1) - 0.8415) >= 0.0001);  r = r || (fabs(sinus(2) - 0.9093) >= 0.0001);  r = r || (fabs(sinus(4) + 0.7568) >= 0.0001);  r = r || (fabs(sinus(10) + 0.5440) >= 0.0001);  return r;  } | /\*Функция для проверки точности вычисления косинуса\*/  int test\_cos() {  int r;  r = 0;  r = r || (fabs(cosinus(0) - 1.0) >= 0.0001);  r = r || (fabs(cosinus(1) - 0.5403) >= 0.0001);  r = r || (fabs(cosinus(2.5) + 0.8011) >= 0.0001);  r = r || (fabs(cosinus(5) - 0.2837) >= 0.0001);  r = r || (fabs(cosinus(10) + 0.8391) >= 0.0001);  return r;  } |

Написанные функции возвращают 0, если результаты тригонометрической функции совпадают с ее действительными значениями, а 1, если результат неверный. После запуска Программа выдает 0, а при изменении аргумента или знака, программа выдает 1.

Далее напишем функцию вычисления тангенса, путем деления синус на косинус.

/\*Функция для вычисления тангенса путем деления синус на косинус\*/

double tangens(double x) {

return sinus(x) / cosinus(x);

}

И для ее тестирования напишем функцию для проверки аналогичную функциям для проверки синуса и косинуса

/\*Функция для проверки точности вычисления косинуса\*/

int test\_tan() {

int r;

r = 0;

r = r || (fabs(tangens(0) - 0.0) >= 0.0001);

r = r || (fabs(tangens(1) - 1.5574) >= 0.0001);

r = r || (fabs(tangens(2.5) + 0.7470) >= 0.0001);

r = r || (fabs(tangens(-4) + 1.1578) >= 0.0001);

r = r || (fabs(tangens(10) - 0.6484) >= 0.0001);

return r;

}

После исполнения, программа выдала 0, что является показателем того, что тангенс вычисляется правильно

**Вывод:**

Я написал функцию, вычисляющую тангенс, с помощью деления синуса на косинус, полученных разложением в ряд Тейлора, тем самым изучил методы разработки консольных приложений, способы их запуска и обработки кодов возврата.