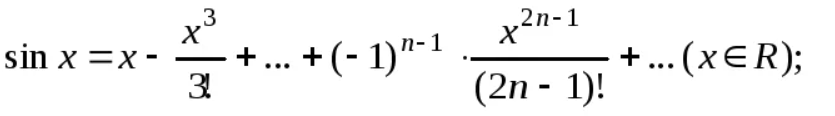
**Цель работы:**

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

**Ход работы:**

Реализуем функцию вычисления синуса, путем разложения в ряд Тейлора. Используем данную формулу



Чтобы написать функцию синуса, необходимы функции факториала и возведения в степень, а также функция модуля, которая понадобится для проверки. Запишем их

/\*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);

}

/\* Функция нахождения модуля числа \*/

double fabs(double x) {

if (x > 0)

return x;

else return x \* -1;

}

С использованием этих функций напишем функцию синуса, вычисляющую сумму первых n членов ряда Тейлора

/\* Функция нахождения синуса разложением в ряд Тейлора \*/

double sinus(double x) {

int n;

double sin;

sin = 0.0;

for (n = 1; n <= 5; n++)

{

sin = sin + (RaiseToPow(-1, n - 1) \* RaiseToPow(x, (2 \* n - 1)) / fact(2 \* n - 1));

}

return(sin);

}

Для проверки этой функции составим набор параметров с помощью калькулятора, с учетом выбранной точности (0.0001).

sin(0)=0

sin(1)=0.8415

sin(2)=0.9093

sin(4)=-0.7568

sin(10)=-0.5440

Далее напишем функцию проверяющую нашу функцию синуса на данных значениях, которая возвращает 0, если функция синуса возвращает значение в допустимой области согласно заданной точности, и 1, если результат не удовлетворяет условиям

/\*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;

}

При проверке пришлось увеличить количество членов в ряду Тейлора до 18. После выполнения программа выводит 0. При изменении значений или знака, программа выдает 1.

**Вывод:**

Я написал функцию вычисляющую синус, с помощью разложения в ряд Тейлора, тем самым изучил методы разработки консольных приложений, способы их запуска и обработки кодов возврата.