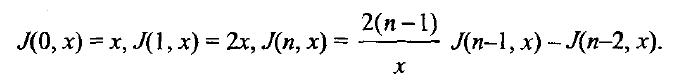
ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

1. Задание

Вычислить функцию Бесселя 8-го порядка с аргументом x:



1. Планируемая реализация

В методе Main() создать вещественную переменную x для хранения аргумента функции и целочисленную переменную n для хранения порядка функции Бесселя. В этом же методе вызывать методы расчёта значений функции рекурсивно и через цикл. При расчёте через цикл, рассчитанные значения планируется хранить в массиве вещественного типа.

Также в задании указано сравнить время расчёта разными способами. В работе планируется отсчитывать время с помощью методов и свойств класса Stopwatch.

1. Итоговая реализация

В методе Main() были созданы целочисленная переменная n и вещественная x. Для отсчёта времени выполнения, за которое разными способами рассчитывается значение функции, используются методы Start() (начало отсчёта времени) и Stop() (конец отсчёта) класса Stopwatch. Расчёт значения функции рекурсивно выполняет метод FuncBesselRecur(), циклом – FuncBesselCicle(). В метод FuncBesselCicle() необходимо передавать значение n на единицу больше в связи с нульбазируемостью массивов в C#. Иначе рассчитывается значение седьмого порядка вместо восьмого.

Фрагмент программного кода

static void Main(){

int n = 8;

double x = 4;

var RecurMethod = Stopwatch.StartNew();

Console.WriteLine(FuncBesselRecur(n, x));

RecurMethod.Stop();

Console.WriteLine(RecurMethod.Elapsed);

var CicleMethod = Stopwatch.StartNew();

Console.WriteLine(FuncBesselCicle(n + 1, x));

CicleMethod.Stop();

Console.WriteLine(CicleMethod.Elapsed);

Console.ReadLine();

}

Рекурсивный расчёт значения функции выполняется методом FuncBesselRecur(). На вход подаются порядок и аргумент функции.

Фрагмент программного кода

private static double FuncBesselRecur(int n, double x){

if (n == 0){

return x;

}

else{

if (n == 1){

return 2.0 \* x;

}

else{

return 2.0 \* (n - 1) \* FuncBesselRecur(n - 1, x) / x - FuncBesselRecur(n - 2, x);

}

}

}

Расчёт значения функции с помощью цикла выполняется методом FuncBesselCicle(). На вход подаётся порядок, увеличенный на единицу, и аргумент функции. С помощью цикла результат рассчитывается быстрее.

Фрагмент программного кода

private static double FuncBesselCicle(int n, double x){

double[] FuncBesselCicle = new double[n];

FuncBesselCicle[0] = x;

FuncBesselCicle[1] = 2 \* x;

for (int i = 2; i < n; i++){

FuncBesselCicle[i] = 2.0 \* (i - 1) \* FuncBesselCicle[i - 1] / x - FuncBesselCicle[i - 2];

}

return FuncBesselCicle[n - 1];

}