Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №3

По дисциплине

“Объектно-ориентированное программирование”

# Тема “Вычисление функций с использованием их разложения в степенной ряд”

Вариант 15

Выполнил работу

студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**1 ПОСТАНОВКА ЗАДАЧИ**

Для х изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x), используя ее разложение в степенной ряд в двух случаях:

а) для заданного n;

б) для заданной точности  (=0.0001).

Для сравнения найти точное значение функции.

|  |  |  |  |
| --- | --- | --- | --- |
| функция | Диапазон  Изменения аргумента | n | сумма |
|  |  | 30 |  |

**2 АЛГОРИТМ ПРОГРАММЫ**

1. Статический метод вычисления общего члена ряда по *x* и *n* по формуле:
2. Статический метод вычисления значения функции по *x*:
3. Статические методы вычисления суммы ряда:

|  |  |
| --- | --- |
| С *K* итераций | До заданной точности |
| S = 0, i = 0  i < K  S += a(++i)  return S | S = 0, i = 0  S – F <= E  S += a(++i)  return S |

**3 ИСХОДНЫЙ КОД ПРОГРАММЫ**

using System;

namespace Lab3

{

delegate double GetNumber(double number);

class Program

{

private const int c\_iR = 3;

private const int c\_iN = 30;

private const double c\_dMin = 0.1d;

private const double c\_dMax = 1d;

private const int c\_iK = 10;

private const double c\_dD = (c\_dMax - c\_dMin) / c\_iK;

private const double c\_dE = 0.0001;

private const string c\_sOutput =

"x\t\t= {0}\nSn\t\t= {1}\nSe (n:{4})\t= {2}\ny = f(x)\t= {3}\n";

private static GetNumber GetRound = number => Math.Round(number, c\_iR);

private static GetNumber GetReal = number => number;

public static void Main(string[] args)

{

Run(GetRound);

Console.ReadKey();

}

private static void Run(GetNumber GetNum)

{

for(double x = c\_dMin; x <= c\_dMax; x += c\_dD)

{

Console.WriteLine(

c\_sOutput, x,

GetNum(SeriesArithmetic(x)),

GetNum(SeriesDifferential(x, out int n)),

GetNum(Function(x)), n);

}

}

private static double Function(double x)

{

return (1 + Math.Pow(x, 2)) / 2 \* Math.Atan(x) - x / 2;

}

private static double Series(double x, int n)

{

return Math.Pow(-1, n + 1) \* Math.Pow(x, 2 \* n + 1)

/ (4 \* Math.Pow(n, 2) - 1);

}

private static double SeriesArithmetic(double x)

{

double Sum = 0;

for (int i = 1; i <= c\_iN; ++i)

Sum += Series(x, i);

return Sum;

}

private static double SeriesDifferential(double x, out int i)

{

i = 0;

double Sum = 0;

double Result = Function(x);

while(Math.Abs(Sum - Result) > c\_dE)

Sum += Series(x, ++i);

return Sum;

}

}

}

**4 СКРИНШОТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

|  |  |
| --- | --- |
| Без округления | С округлением |
|  |  |