Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

**«ОБЧИСЛЕННЯВИЗНАЧЕНОГО ІНТЕГРАЛУ»**

з дисципліни: «Інформатика 2»

|  |  |
| --- | --- |
|  | Виконав: Клижевский Денис  Група: РЕ-11  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

* Мета роботи: навчитися працювати з функціями користувача та за їх допомогою скласти програму для обчислення визначеного інтегралу кількома методами, дослідити отримані результати.

Код програми:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

double function(double x){

return x = x\*x;

}

double right\_columns(double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit;

double integral = 0;

for (int i = 1; i <= interval; i++)

{

integral = integral + function(pointer) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double left\_columns (double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit + width\_columns;

double integral = 0;

for (int i = 1; i <= interval; i++)

{

integral = integral + function(pointer) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double trapeze (double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit;

double integral = 0;

for (int i = 1; i < interval; i++)

{

integral = integral + ((function(pointer) + function(pointer + width\_columns)) / 2) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double simpson(double left\_limit, double right\_limit, double interval) {

double integral=0;

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit + width\_columns;

while (pointer < right\_limit)

{

integral += 4\*function(pointer);

pointer += width\_columns;

if (pointer >= right\_limit) break;

integral += 2\*function(pointer);

pointer += width\_columns;

}

integral = (width\_columns / 3) \* (integral + function(left\_limit) + function(right\_limit));

return integral;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double first\_integral, second\_integral, left\_limit, right\_limit, error;

int spacing\_option, metod\_integration;

unsigned long int counter, interval = 2;

{//шапка

printf("\n\n\t++===================================================++\n");

printf("\t|| №2 ||\n");

printf("\t++---------------------------------------------------++\n");

printf("\t|| Посчитать определенный интеграл ||\n");

printf("\t++===================================================++\n\n");

}

{//ввод начальных данных

printf("\t Введите левую границу\t"); scanf("%lf", &left\_limit);

printf("\t Введите правую границу\t"); scanf("%lf", &right\_limit);

}

{//выбор метода

printf("\n\t\t 1 - Правых прямоугольников");

printf("\n\t\t 2 - Левых прямоугольников");

printf("\n\t\t 3 - Трапеций");

printf("\n\t\t 4 - Симпсона");

printf("\n\t Выберите метод вычисления определенного интеграла:\t"); scanf("%d", &metod\_integration);

}

{//выбор способа

printf("\n\t\t 1 - Через рассчитанное количество промежутков");

printf("\n\t\t 2 - Через своё количество промежутков");

printf("\n\t Выберите способ вычисления определенного интеграла:\t"); scanf("%d", &spacing\_option);

}

if (spacing\_option == 1)

{

printf("\n\t\t От 3 до 5 порядка");

printf("\n\t Введите погрешность для вычисления определенного интеграла:\t"); scanf("%lf", &error);

error = 1/pow(10, error);

switch (metod\_integration) {

case 1: {

do {

first\_integral = right\_columns (left\_limit, right\_limit, interval);

second\_integral = right\_columns (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 2: {

do {

first\_integral = left\_columns (left\_limit, right\_limit, interval);

second\_integral = left\_columns (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 3: {

do {

first\_integral = trapeze (left\_limit, right\_limit, interval);

second\_integral = trapeze (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 4: {

do {

first\_integral = simpson (left\_limit, right\_limit, interval);

second\_integral = simpson (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

}}

if (spacing\_option == 2)

{

printf("\n\t Введите количество промежутков для вычисления определенного интеграла:\t"); scanf("%lu", &counter);

switch (metod\_integration) {

case 1: {

first\_integral = right\_columns (left\_limit, right\_limit, counter);

} break;

case 2: {

first\_integral = left\_columns (left\_limit, right\_limit, counter);

} break;

case 3: {

first\_integral = trapeze (left\_limit, right\_limit, counter);

} break;

case 4: {

first\_integral = simpson (left\_limit, right\_limit, counter);

} break;

}}

{//результаты

printf("\n\t Левый предел:\t"); printf("%lf", left\_limit);

printf("\n\t Правый предел:\t"); printf("%lf", right\_limit);

printf("\n\t Количество промежутков:\t"); printf("%lu", counter);

printf("\n\t Погрешность вычислений:\t"); printf("%lf", error);

printf("\n\n\t Изначальная функция:\t"); printf("x^2");

printf("\n\t Аналитическое интегрирование:\t"); printf("(x^3)/3");

printf("\n\t Результат вычислений:\t"); printf("%lf\n\n", first\_integral);

}

printf("\n\t\t\tПосле нажатия любой клавиши программа повторится\n\n");

getch(); system("cls"); main();

return 0;

}