Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине «Традиционные и интеллектуальные информационные технологии»

Тема: «Решение задач методом Монте-Карло»

**Выполнил:**

Студент 1 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Анфилец С.В.

Брест 2021

**Цель работы:** Вычислить численное значение интеграла на интервале.

**Ход работы:**

1. **Задание взять из Лаб.работы №2 (прошлый семестр). Выписать постановку задачи и полученное точное значение интеграла (I).**

Постановка задачи:

Реализовать программу вычисления приблизительным методом. Частота n = 10, 20,50,100, 500 (количество отрезков на которые разбивается интервал интегрирования).Вариант 6.

Точное значение интеграла: 0.24023.

1. **Определить область для генерации точек. Значения по оси OX даны по заданию, а значение по оси OY нужно определить.**

Значения по оси OX из прошлого задания : [1,2].

Значения по оси OY найдём по следующей программе:

**Код программы:**

#include <iostream>

#include <cmath>

int main() {

for (double x = 1; x < 2.1; x += 0.1)

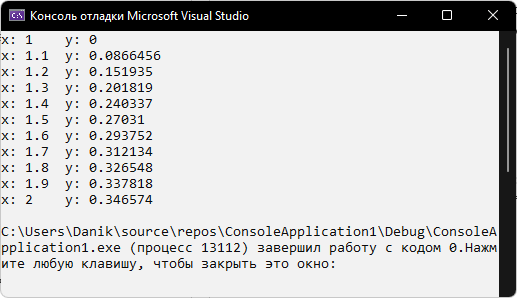
std::cout << "x: " << x << " y: "

<< log(x) / x << std::endl;

return 0;

}

**Результат программы:**



1. **Рассчитать площадь полученного прямоугольника (Sпр) в п.2:**

Sпр = (2-1) \* (0.346574-0) = 0.346574.

1. **Провести эксперименты с N = 100, 500, 1000, 3000, 10000. Где N — количество точек, сгенерированных случайным образом:**

**Код программы:**

#include <iostream>

#include <cmath>

#include <random>

#include <chrono>

#include <iomanip>

using namespace std;

using namespace chrono;

default\_random\_engine dre((unsigned)(system\_clock::now().time\_since\_epoch().count()));

void Monte\_Karlo(double n) {

double min = log(1) / 1,

max = log(2) / 2,

square\_rect = (2 - 1) \* (max-min),count = 0.0;

uniform\_real\_distribution<double> dist\_X(1.0, 2.0);

uniform\_real\_distribution<double> dist\_Y(min, max);

for (double i = 0; i < n; i++){

double x = dist\_X(dre);

double y = dist\_Y(dre);

if (y <= log(x) / x) count++;}

double square\_figure = count / n \* square\_rect;

cout <<"count: "<<count<<setprecision(5) << " S: "<< square\_figure << endl; }

int main() {

cout << "100: "; Monte\_Karlo(100);

cout << "200: "; Monte\_Karlo(200);

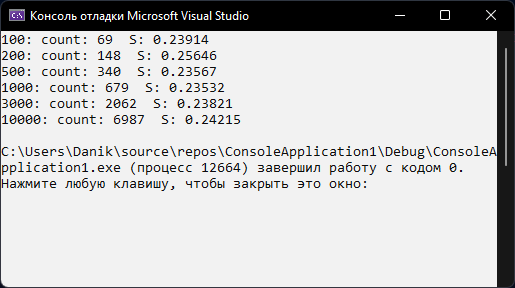
cout << "500: "; Monte\_Karlo(500);

cout << "1000: "; Monte\_Karlo(1000);

cout << "3000: "; Monte\_Karlo(3000);

cout << "10000: "; Monte\_Karlo(10000);return 0;}

**Результат программы:**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | N | n | Sф | I | Error |
| 1. | 100 | 69 | 0.23914 | 0.24023 | 0,00109 |
| 2. | 200 | 148 | 0.25646 | 0.24023 | 0,01623 |
| 3. | 500 | 340 | 0.23567 | 0.24023 | 0,00456 |
| 4. | 1000 | 679 | 0.23532 | 0.24023 | 0,00491 |
| 5. | 3000 | 2062 | 0.23821 | 0.24023 | 0,00202 |
| 6. | 10000 | 6987 | 0.24215 | 0.24023 | 0,00192 |

**Вывод:** В ходе лабораторной работы был изучен метод Монте-Карло, реализована программа для вычисления численного значения интеграла на интервале.