Министерство образования Республики Беларусь

Учреждение образования

Кафедра ИИТ

Лабораторная работа №1

По дисциплине «Традиционные и интеллектуальные информационные технологии»

# Тема: « **Решение задач методом Монте-Карло**»

Выполнил:

Студент 1 курса

Группы ИИ-23

Романюк А.П.

Проверил:

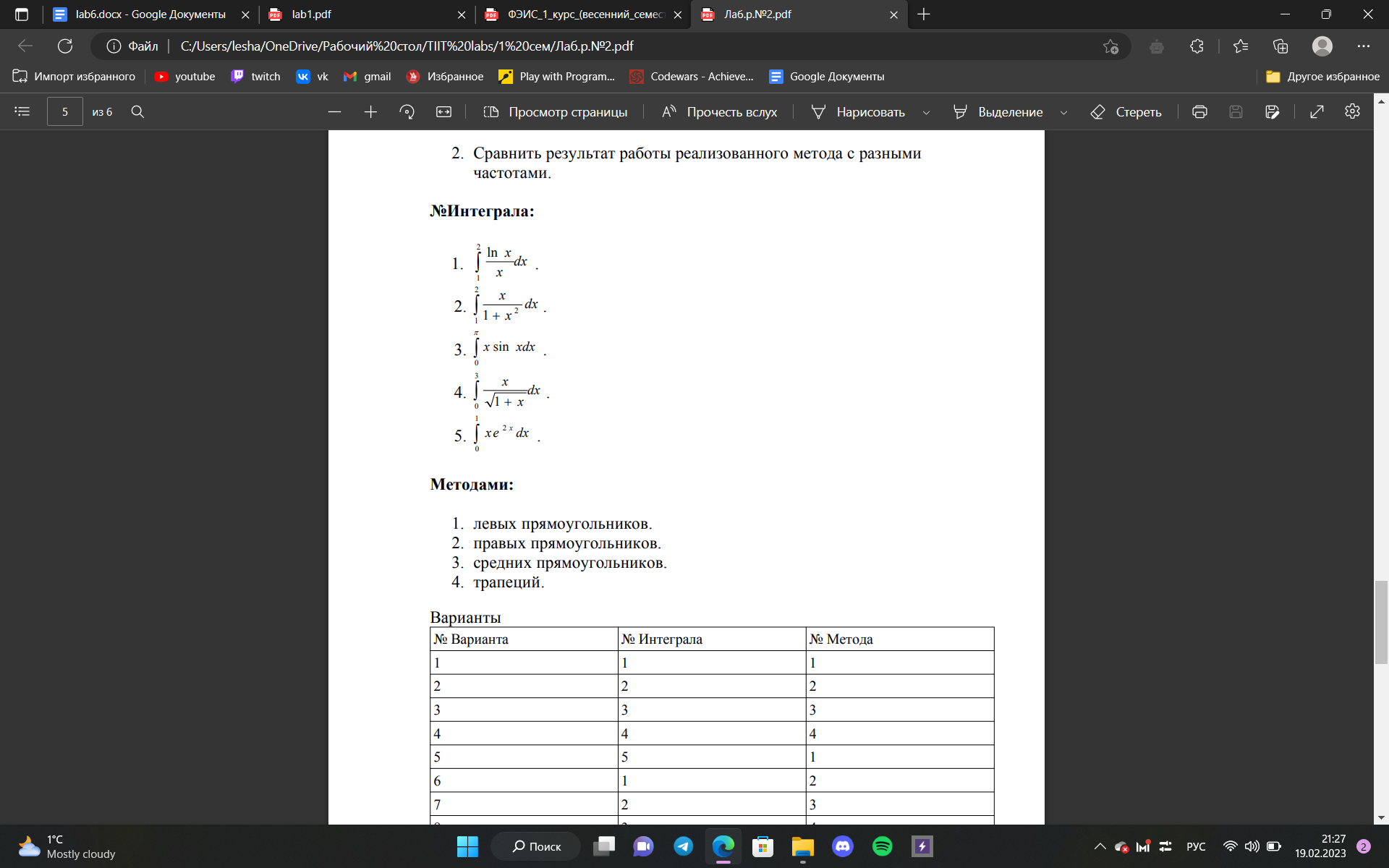
Анфилец С. В.

2023

Вариант 8

Задача: Вычислить численное значение интеграла на интервале

Ход работы:

1)Заданный интеграл:Значение интеграла:3.14159.

2)Возьмем областью для генерации точек Ox:pi, Oy:4;

3)Площадь полученного прямоугольника 12.566.

4)Проведем эксперименты для разных N точек при помощи программы.

Код программы:

#include <iostream>

#include <stdlib.h>

#include <cmath>

#include <math.h>

#define pi 3.14159

using namespace std;

double func(double x) {

return x \* sin(x);

}

int main()

{

double x, y, ox = pi, oy = 4, N = 0, n = 0, I = pi;

double ux, uy;

for (int i = 0; i < 10001; i++) {

ux = (float)rand() / (float)RAND\_MAX;

uy = (float)rand() / (float)RAND\_MAX;

x = ox \* ux;

y = oy \* uy;

if (y <= func(x)) n++;

if (i == 100) cout << "N = " << i << " n = " << n << " Sф = " << ((ox \* oy) \* n) / i << " I = " << I << " Error = " << abs((((ox \* oy) \* n) / i) - I) / I << "\n";

if (i == 500) cout << "N = " << i << " n = " << n << " Sф = " << ((ox \* oy) \* n) / i << " I = " << I << " Error = " << abs((((ox \* oy) \* n) / i) - I) / I << "\n";

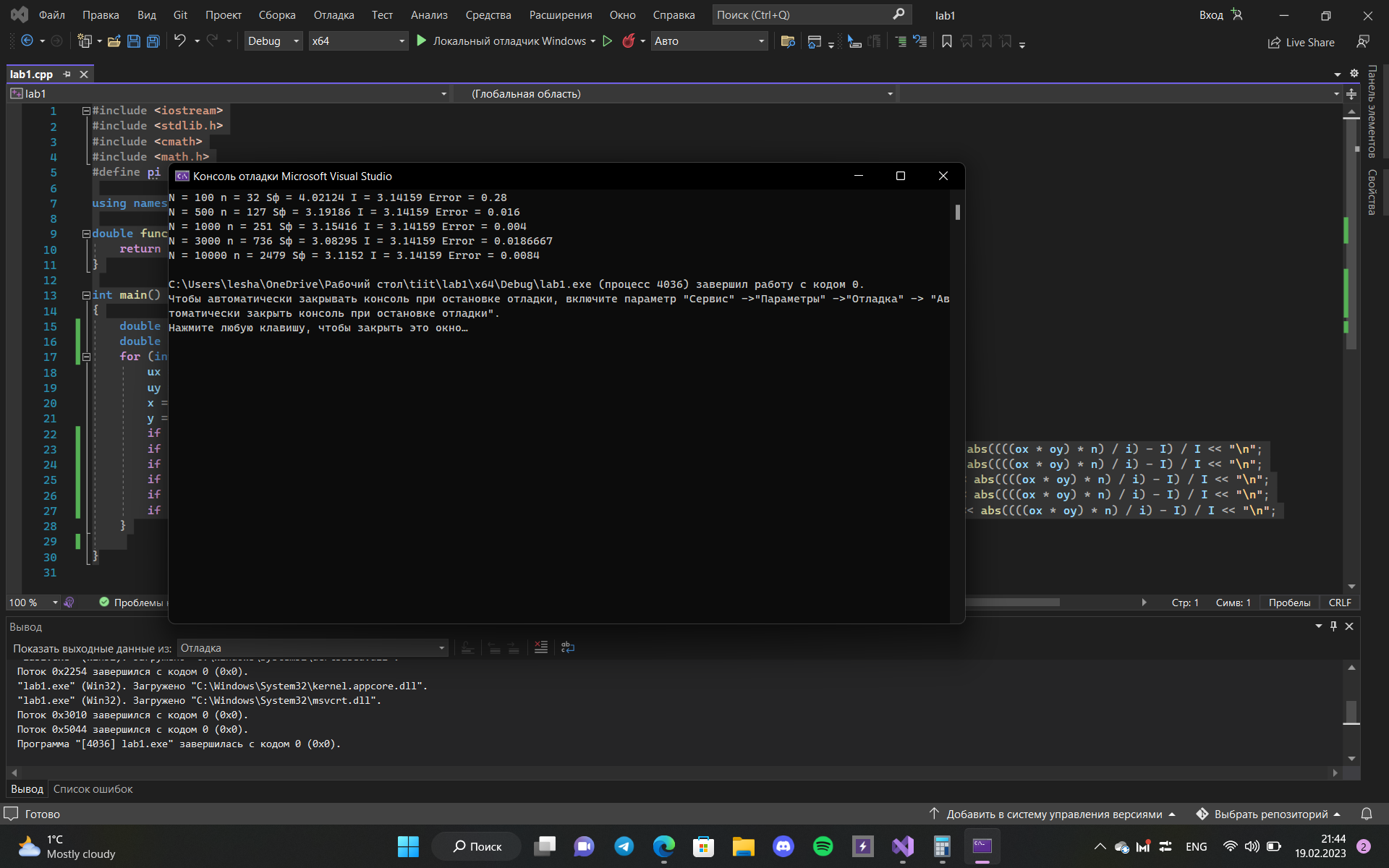
if (i == 1000) cout << "N = " << i << " n = " << n << " Sф = " << ((ox \* oy) \* n) / i << " I = " << I << " Error = " << abs((((ox \* oy) \* n) / i) - I) / I << "\n";

if (i == 3000) cout << "N = " << i << " n = " << n << " Sф = " << ((ox \* oy) \* n) / i << " I = " << I << " Error = " << abs((((ox \* oy) \* n) / i) - I) / I << "\n";

if (i == 10000) cout << "N = " << i << " n = " << n << " Sф = " << ((ox \* oy) \* n) / i << " I = " << I << " Error = " << abs((((ox \* oy) \* n) / i) - I) / I << "\n";

}

}



| № | N | n | Sф | I | Error |
| --- | --- | --- | --- | --- | --- |
| 1 | 100 | 32 | 4.02124 | 3.14159 | 0.28 |
| 2 | 500 | 127 | 3.19186 | 3.14159 | 0.016 |
| 3 | 1000 | 251 | 3.15416 | 3.14159 | 0.004 |
| 4 | 3000 | 736 | 3.08295 | 3.14159 | 0.0186667 |
| 5 | 10000 | 2479 | 3.1152 | 3.14159 | 0.0084 |

Вывод: Научился пользоваться методом Монте-Карло.