Министерство образования Республики Беларусь

Учреждение образования

Кафедра ИИТ

Лабораторная работа №4

По дисциплине «Традиционные и интеллектуальные информационные технологии»

# Тема: «Подключение внешних модулей и применение параллельного подхода»

Выполнил:

Студент 1 курса

Группы ИИ-23

Романюк А.П.

Проверил:

Анфилец С. В.

2023

Вариант 8

Задание. 1. Выделить в отдельный файл функции: расчёта функционала, пересчёта новой позиции точки (вспомогательные, если есть). 2. Реализовать их в виде подключаемых библиотек (создать заголовочный файл \*.h и пр. ) 3. Подключить вашу библиотеку к новому проекту( #include). 4. Создать в новом проекте популяцию из 1000 элементов (A[1000][3]). 5. Реализовать популяционный алгоритм поиска экстремума. Реализация должна использовать метод параллельных потоков. Распараллеливание алгоритма предлагается сделать в месте пересчёта новых значений точек (т.к. их количество 1000 и изменение каждой из них не зависит от других).

Код программы:

Lab4.cpp:

#include <iostream>

#include <omp.h>

#include "functions.h"

using namespace std;

int main()

{

srand(time(NULL));

double alf, alf\_min = 0.0000001, C = 0, b = 5;

double a[1000][3]{};

cout << "Введите значение шага: ";

cin >> alf;

while (C < 0.75 || C > 0.99) {

cout << "\nВведите параметр затухания [0.75,0.99]: ";

cin >> C;

if (C < 0.75 || C > 0.99) cout << "Error";

}

#pragma omp parallel for

{

for (int i = 0; i < 1000; i++) {

a[i][1] = frand(-1000, 1000);

minim(b, alf, alf\_min, C, a[i][0], a[i][1], a[i][2]);

}

}

double min;

min = a[0][0];

for (int i = 0; i < 1000; i++) {

if (a[i][0] < min) min = a[i][0];

}

cout << "Минимум: " << min;

}

functions.h:

#pragma once

#include <iostream>

#include <cmath>

using namespace std;

double f(double x, double b) {

return cos(x) + (1 / b) \* cos(7 \* x + 1) + (1 / pow(b, 2)) \* cos(49 \* x + 2) + (1 / pow(b, 3)) \* cos(343 \* x + 3) + (1 / pow(b, 4)) \* cos(2401 \* x + 4);

//return sin(x);

}

double frand(double fMin, double fMax)

{

double f = (double)rand() / RAND\_MAX;

return fMin + f \* (fMax - fMin);

}

void minim(double b, double alf, double alf\_min, double C, double&y, double& x1,double& x2) {

while (alf > alf\_min) {

x2 = x1 + alf \* frand(-1, 1);

if (f(x2, b) < f(x1, b)) {

x1 = x2;

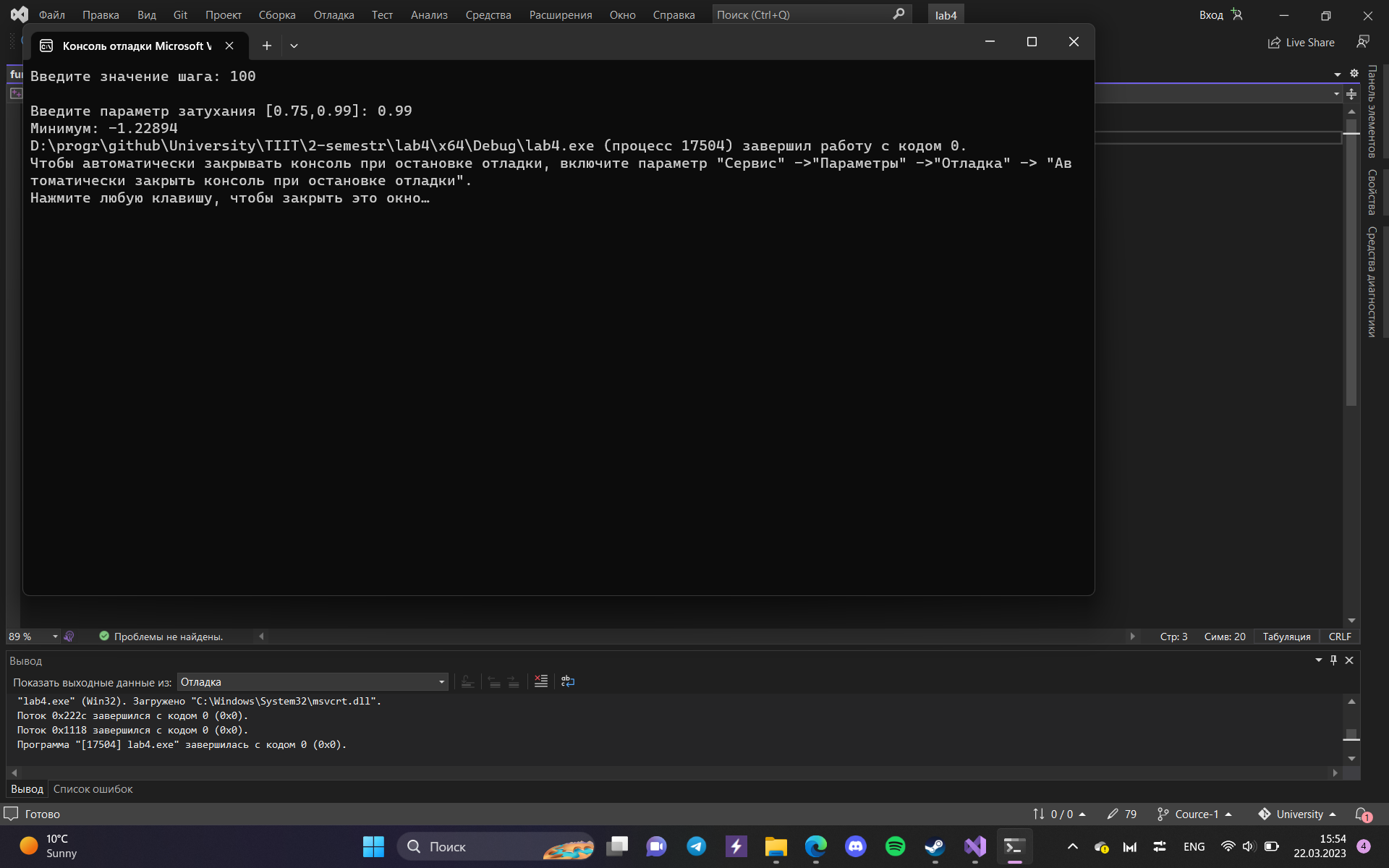
}

alf \*= C;

y = f(x1, b);

}

}



Вывод: Изучил подключение внешних модулей и применение параллельного подхода.