Министерство образования Республики Беларусь

Учреждение образования

Кафедра ИИТ

Лабораторная работа №5

По дисциплине «Традиционные и интеллектуальные информационные технологии»

# Тема: «Имитационное моделирование »

Выполнил:

Студент 1 курса

Группы ИИ-23

Романюк А.П.

Проверил:

Анфилец С. В.

2023

Вариант 8

Задание. 0) Вспомнить темы: имитационное моделирование, генерация случайных чисел. 1) Реализовать процесс имитации развития вирусной болезни по данным Вашего варианта. 2) Выполнить имитацию 50 дней. Получить по результатам имитации следующую статистику: - График роста заболевших. - Математическое ожидание количества заболеваний в день. - Стандартное отклонение количества заболеваний в день. 4) Изменить какой-либо параметр на △1 , △2 , △3 , провести тесты из пункта 2. Сравнить результаты.

Код программы:

Lab5.cpp:

#include <iostream>

#include <cmath>

#include <iomanip>

#include "f.h"

using namespace std;

int main() {

vector<double> data = {0};

int ill, day = 1, contacts, dead = 0, cured = 0;

ill = frand(5, 10);

agent\* people = new agent[ill];

cout << setw(8) << "day" << setw(8) << "ill" << setw(8) << "dead" << setw(8) << "cured" << setw(8) << "ill+" << setw(8) << "dead+" << setw(8) << "cured+"<<endl;

for (day; day < 51; day++) {

int ill\_today = 0, dead\_today = 0,cured\_today = 0;

contacts = 0;

for (int i = 0; i < ill; i++) {//вычисляем количество контактов

if (people[i].status == 2) {

people[i].SC = people[i].getSC();

contacts += people[i].SC;

}

else if (people[i].status == 3) {

people[i].SC = people[i].getSC() / people[i].R;

contacts += people[i].SC;

}

}

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

double rand\_test = (int)frand(0, 100);//проверяем узнал ли человек о болезни

if (people[i].status == 2) {

if (reg.T > rand\_test) {

people[i].status = 3;

}

}

}

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

double rand\_HA = (int)frand(0, 100);//проверяем попал ли человек на лечение

if (people[i].status == 3) {

if (people[i].HA < 2 && rand\_HA < 10) people[i].status = 4;

if (people[i].HA > 2 && rand\_HA < 1) people[i].status = 4;

}

}

for (int i = 0; i < ill; i++) {//перерасчет времени выздоровления и шанса смерти для попавших на лечение

if (people[i].status == 4 && people[i].started == 0) {

people[i].RT = people[i].getRT();

people[i].DR = people[i].getDR();

people[i].started = 1;

}

}

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

double death\_rand = (int)frand(0, 100);//проверка на смерть

if (people[i].status != 1) {

if (death\_rand < people[i].DR) {

people[i].status = 1;

dead\_today++;

dead++;

}

}

}

for (int i = 0; i < ill; i++) { //проверяем сколько человек вылечилось

if (people[i].status != 1) {

people[i].RT--;

if (people[i].RT < 1) {

people[i].status = 1;

cured++;

cured\_today++;

}

}

}

ill\_today = ill / (ebola.I \* 100);//считаем кол-во новых больных

data.push\_back(ill\_today/100);

agent\* buf = new agent[ill];

ill += ill\_today;

people = new agent[ill];

for (int i = 0; i < ill-ill\_today; i++) {

people[i]=buf[i];

}

cout << setw(8) << day << setw(8) << ill << setw(8) << dead << setw(8) << cured << setw(8) << ill\_today << setw(8) << dead\_today << setw(8) << cured\_today << endl;

}

//делаем графичек

int max\_value = \*max\_element(data.begin(), data.end());

for (int i = max\_value; i >= 0; --i) {

cout.width(4);

cout << i\*100 << " | ";

for (int j = 0; j < data.size(); ++j) {

if (data[j] >= i) {

cout << "x ";

}

else {

cout << " ";

}

}

cout << endl;

}

cout << "---------------------------------------"

<< endl;

//дополнительные вычисления

cout << "\nexpectrd value 20.5%\n";

}

f.h:

#pragma once

#include <cmath>

#include <iostream>

#include <random>

#include <vector>

using namespace std;

double frand(double fMin, double fMax)

{

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

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

}

double N(double m, double sigm) {//Преобразование Бокса — Мюллера, нормальная генерация

double x, y,z;

x = frand(0, 1);

y = frand(0, 1);

z = cos(2 \* 3.14 \* x) \* sqrt(-2 \* log(y));

return m + sigm \* z;

}

double U(double c, double d) {//равномерная генерация

double min = c - d;

double max = c + d;

return frand(min,max);

}

double E(double lambd) {//экспоненциальное разделение

return (log(1-frand(0,1)) / (-lambd));

}

class virus {

public:

double I = U(20, 20);

};

virus ebola;

class enviroment {

public:

double AP = N(25, 5.5);

double getM() {

M = N(20, 3.5);

return M;

};

double T = N(3, 1.5);

private:

double M;

};

enviroment reg;

class agent {

public:

double getSC() {

return U(m, 6);

}

int getRT() {

return 1 + U(20 - 1.5 \* HA - 0.5 \* Mself, 5 - 0.25 \* HA - 0.1 \* Mself);

}

double getDR() {

return A \* 0.05 + U(15 - 1.5 \* HA - 0.5 \* Mself, 3 - 0.25 \* HA - 0.1 \* Mself);

}

agent() {

status = 2;

m = frand(0, 50);

R = 10 - E(0.5);

HA = N(6, 0.5);

A = N(reg.AP, 5.5);

Mself = reg.getM();

RT = 1 + U(20 - 1.5 \* HA , 5 - 0.25 \* HA);

DR = (A \* 0.05 + U(15 - 1.5 \* HA, 3 - 0.25 \* HA));

}

bool started = 0;//перепроверил ли человек время смерти

int m;

int Mself;//эффективность лечения этого человека

int status;

double SC;

double R;

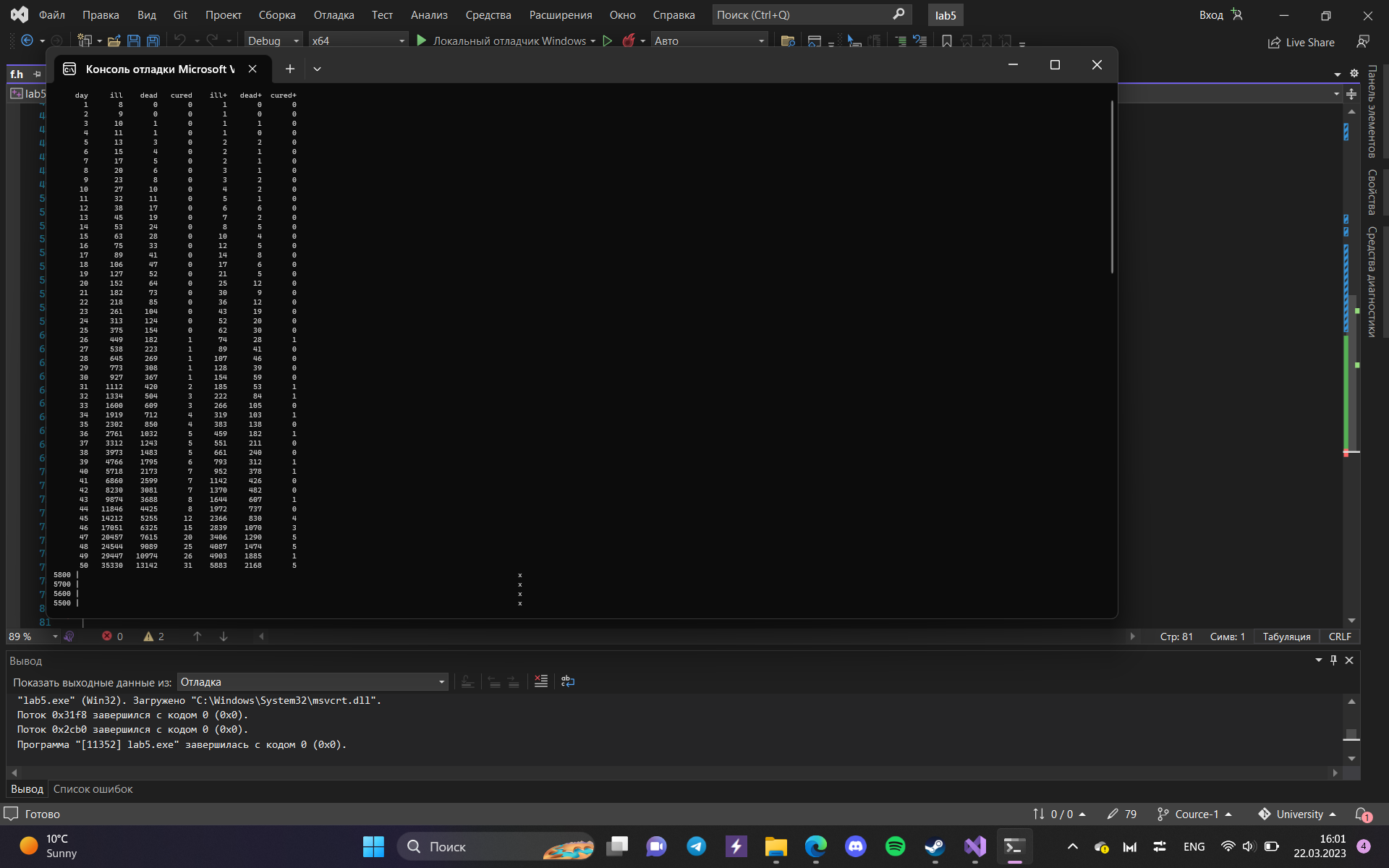
double HA;

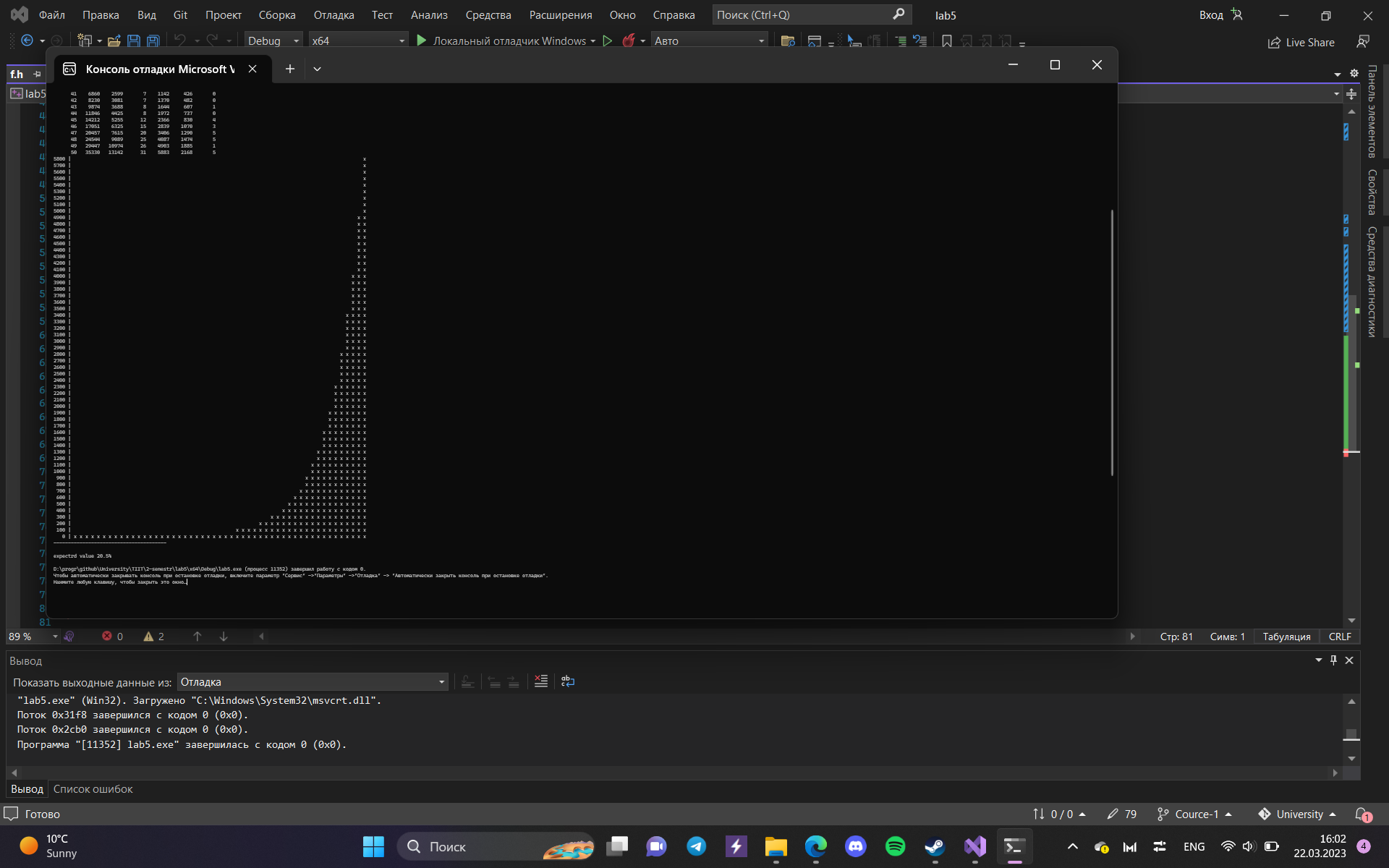
double A;

int RT;

double DR;

};





Вывод:Смоделировал развитие вируса.