Министерство Образования республики Беларусь

Учреждение Образования

«Брестский Государственный технический университет»

Кафедра ИИТ

# Лабораторная работа №5

По дисциплине «Традиционные и интеллектуальные информационные технологии»

Тема: «Имитационное моделирование»

**Выполнил:**

Студент 1 курса

Группы ИИ-23

Макаревич Никита Русланович

**Проверила**:

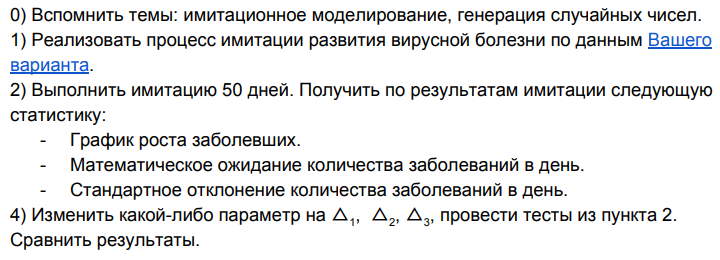
Анфилец С.В.

Брест 2023

**Цель работы:**

Реализовать макромодель развития вируса.

**Задание:**



**Код программы:**

**import java.io.BufferedWriter;**

**import java.io.FileWriter;**

**import java.io.IOException;**

**import java.util.Random;**

**import java.util.Scanner;**

**public class Main {**

**static class Patient{**

**boolean isIll = false;**

**boolean isHealing = false;**

**boolean isAlive = true;**

**boolean bitten = false;**

**boolean recoveredToday = false;**

**int healingDays;**

**int socialContacts;**

**double responsibility;**

**double health;**

**double age;**

**}**

**public static int recoveryTime(double treatmentEf, Patient infected){**

**Random random = new Random();**

**double n = 2 + (30 - 2.2 \* infected.health - 0.8 \* treatmentEf)**

**+ random.nextGaussian() \* Math.sqrt(4 - 0.15 + infected.health - 0.12 \* treatmentEf);**

**if(n > 0)**

**return (int)(n + 0.5);**

**else return 0;**

**}**

**public static double deathRate(double treatmentEf, Patient infected){**

**Random random = new Random();**

**double n = infected.age \* 0.05 + random.nextDouble(15 - 1.5 \* infected.health - 0.5 \* treatmentEf,**

**3 - 0.25 \* infected.health - 0.1 \* treatmentEf);**

**if(n > 0)**

**return n;**

**else return 0;**

**}**

**public static void giveStats(Patient[] city, double countryAge){**

**Random random = new Random();**

**for (int i = 0; i < city.length; i++) {**

**city[i].responsibility = 2 + random.nextGaussian() \* Math.sqrt(0.1);**

**city[i].health = 10 + random.nextGaussian() \* Math.sqrt(1.5);**

**city[i].age = random.nextDouble(countryAge - 30,countryAge + 30);**

**}**

**}**

**public static void dayOfInfected(Patient[] city, Patient infected, double infectiousness){**

**Random random = new Random();**

**int bites = 0; //сколько человек заразится**

**for (int k = 0; k < infected.socialContacts; k++) {**

**double chance = random.nextDouble(0,1);**

**if (chance < infectiousness)**

**bites++;**

**}**

**for (int k = 0; k < city.length && bites > 0; k++) {**

**if(!city[k].isIll && !city[k].recoveredToday){**

**city[k].isIll = true;**

**city[k].bitten = true;**

**bites--;**

**}**

**}**

**}**

**public static void testChance(Patient infected, double testPercent, double treatmentEf){**

**Random random = new Random();**

**double testChance = random.nextDouble(0,100);**

**if (testChance <= testPercent){**

**infected.isHealing = true;**

**infected.healingDays = recoveryTime(treatmentEf, infected);**

**}**

**}**

**public static void deathChance(Patient infected, double treatmentEf){**

**Random random = new Random();**

**double deathRate = deathRate(treatmentEf, infected);**

**double deathChance = random.nextDouble(0, 1.0);**

**if(deathChance <= deathRate)**

**infected.isAlive = false;**

**}**

**public static void main(String[] args) {**

**Random random = new Random();**

**Scanner sc = new Scanner(System.in);**

**double countryAge = 30 + random.nextGaussian() \* Math.sqrt(2.5);**

**double treatmentEf = random.nextDouble(7.5,12.5);**

**double testPercent = random.nextDouble(1.5, 4.5);**

**int numOfInfected = random.nextInt(5,11);**

**int numOfDead = 0;**

**int numOfHealing = 0;**

**double infectiousness = random.nextExponential();**

**if(infectiousness >= 1)**

**infectiousness = 0.99;**

**System.out.println("Введите кол-во жителей: ");**

**int population = sc.nextInt();**

**System.out.println(numOfInfected + " начальных зараженных");**

**Patient[] city = new Patient[population];**

**for (int i = 0; i < population; i++) {**

**city[i] = new Patient();**

**}**

**giveStats(city, countryAge);**

**for (int i = 0; i < numOfInfected; i++) {**

**city[i].isIll = true;**

**}**

**try {**

**BufferedWriter writerInfected = new BufferedWriter(new FileWriter("output.txt"));**

**BufferedWriter writerDead = new BufferedWriter(new FileWriter("output2.txt"));**

**BufferedWriter writerHealing = new BufferedWriter(new FileWriter("output3.txt"));**

**for (int i = 0; i < 50; i++) { // в течении 50 дней**

**for (int j = 0; j < population; j++) { //проход по всему городу**

**if (city[j].isAlive){**

**if(city[j].isIll && !city[j].bitten){ // если болен и не заболел на текущей итерации**

**numOfInfected++;**

**int m = random.nextInt(0,51);**

**city[j].socialContacts = random.nextInt(m - 10,m + 10);**

**if(city[j].isHealing){**

**numOfHealing++;**

**city[j].socialContacts /= city[j].responsibility;**

**city[j].healingDays--;**

**if(city[j].healingDays <= 0){**

**numOfHealing--;**

**city[j].isHealing = false;**

**city[j].recoveredToday = true;**

**city[j].isIll = false;**

**}**

**}**

**dayOfInfected(city, city[j], infectiousness);**

**testChance(city[j], testPercent, treatmentEf);**

**deathChance(city[j], treatmentEf);**

**}**

**city[j].bitten = false;**

**city[j].recoveredToday = false;**

**}**

**else numOfDead++;**

**}**

**//запись в файла для построения графиков**

**writerInfected.write("(" + i + ";" + numOfInfected +")");**

**writerDead.write("(" + i + ";" + numOfDead +")");**

**writerHealing.write("(" + i + ";" + numOfHealing +")");**

**numOfInfected = numOfDead = numOfHealing = 0;**

**}**

**writerInfected.close();**

**writerDead.close();**

**writerHealing.close();**

**} catch (IOException e) {**

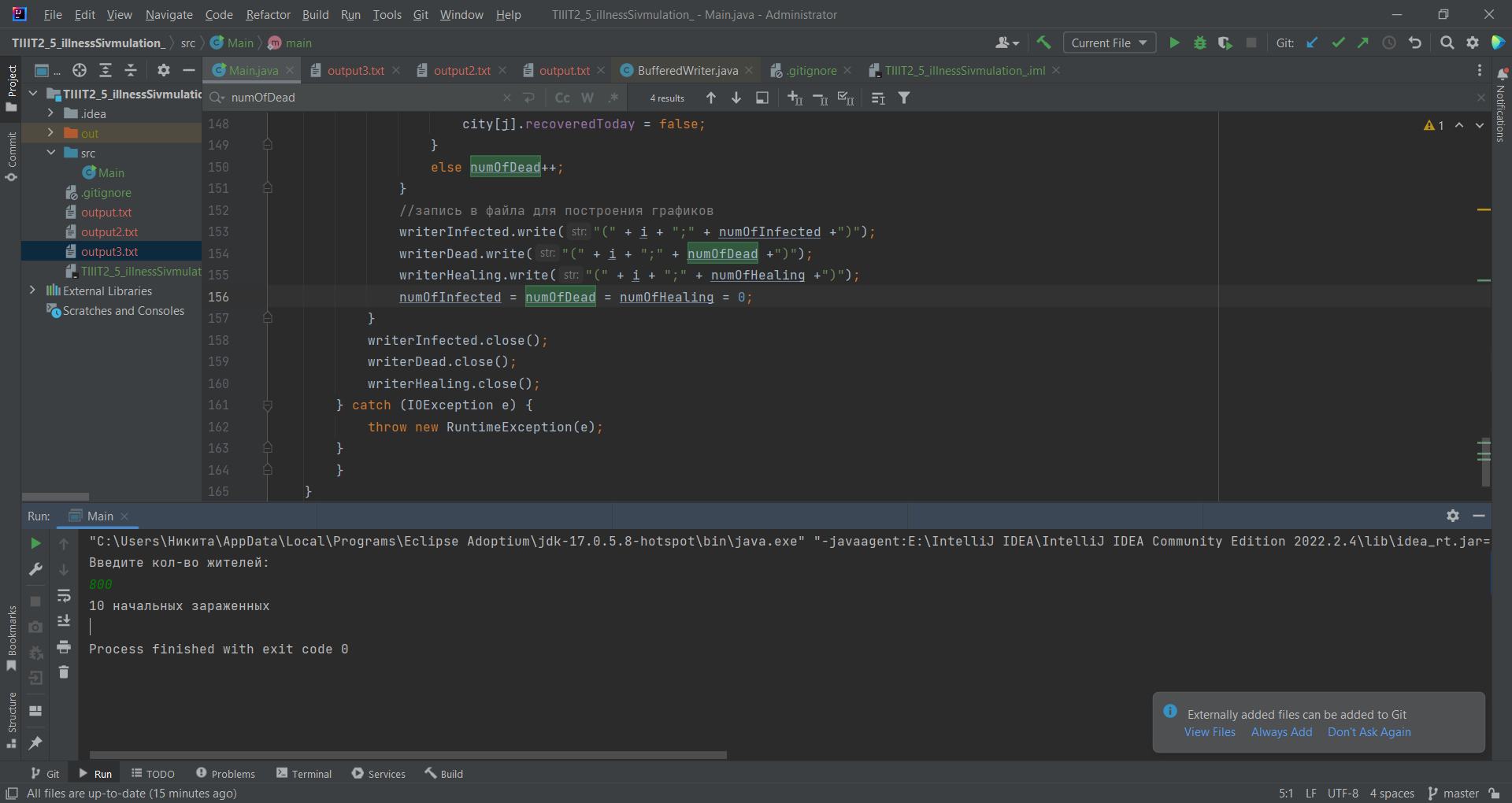
**throw new RuntimeException(e);**

**}**

**}**

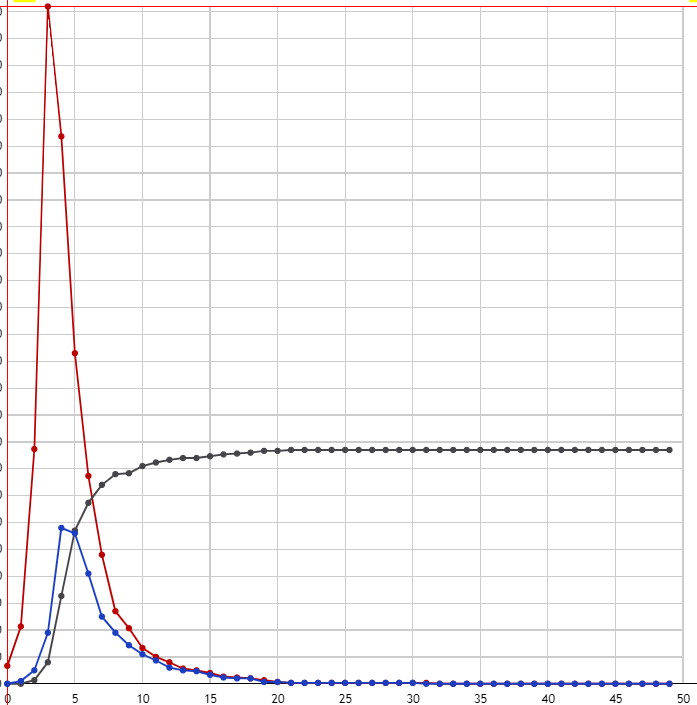
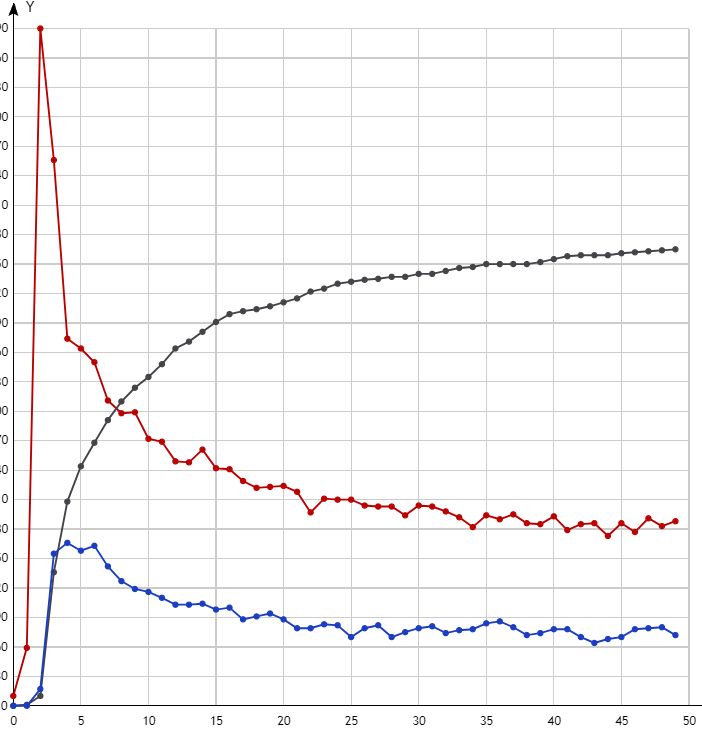
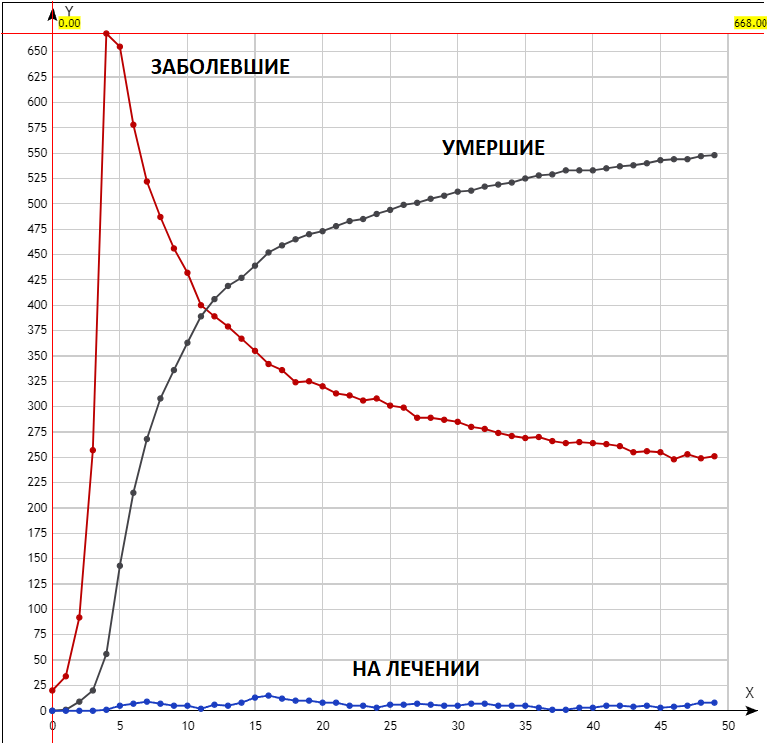
**}**

**Результат программы:**



ГРАФИКИ:

1. Параметры по варианту
2. Параметры по варианту + вручную поднял шанс успешного теста на болезнь
3. Параметры по варианту + поднял шанс на успешный тест + все кто переболел получают имунитет



**Вывод:**

В ходе лабораторной реализовал макромодель развития вируса. Для изучения данных методов использовался язык программирования Java, для построения графиков использовался сайт http://yotx.ru