**Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования**

**Московский технический университет связи и информатики**

**КАФЕДРА ИНФОРМАТИКИ**

**Лабораторная работа по теме 2  
Компьютерное моделирование моделей распространения инфекций**

**Выполнила:**

**Студентка группы БВТ1701**

**Сенькова Ольга**

**Вариант 22**

**Москва 2019**

**Цель работы:** реализовать модель распространения инфекции SIS/SIR.

**1 Формулировка задания на лабораторную работу**

Для сети, спроектированной в лабораторной работе №1, реализовать модель распространения инфекций SIS/SIR. Результаты работы, оформить в виде отчета со скриншотами работы модели и кодом основных функций.

**2 Техническое задание**

-Параметры ввода: InfactionRate – вероятность заражения, RecoveryTime – время восстановления, timer1.Interva - скорость моделирования, i- заражение узла по X, j- заражение узла по Y/

-Параметры вывода: наглядная картина заражения и восстановления SIS/SIR.

-Класс Node: класс узлов, состоящий из i- номер узла по вертикали, j- номер узла по горизонтали, X- координаты узла по оси X, Y- координаты узла по осиY

**1. SIR модель распространения инфекции**

SIR (Susceptible–Infected–Removed) модель включает разделение населения на три группы: S — количество лиц восприимчивых к заболеванию, I — количество инфицированных и R — количество лиц, которые выздоровели и имеют иммунитет. Впервые эта модель была упомянута в трудах британских ученых Kermack W.O. и McKendrick A.G., опубликованных в 1927.

Эта модель не является сложной при решении и одновременно позволяет строить распространения многих инфекционных заболеваний, в том числе против кори, эндемического паротита и краснухи.

Для того чтобы, показать, что значение S, I, R меняются со временем (даже если общая численность населения остается неизменной), мы будем обозначать их как функции, зависящие от времени S (t), I (t) и R (t). Эти функции будут менять в зависимости от заболевания и популяции, чтобы иметь возможность спрогнозировать возможные вспышки и взять их под контроль.

**2. SIS модель распространения инфекции**

Некоторые инфекции, например, простуда и грипп, не дают длительного иммунитета. Такие инфекции не дают иммунизации после выздоровления от инфекции, и люди снова становятся восприимчивыми.

Такая модель получила название SIS (Susceptible–Infected- Susceptible).

**3 Функции программы**

-private void button3\_Click(object sender, EventArgs e)

Основная функция при нажатии устанавливает узлы заражения, а после при помощи вероятности заражает соседние узлы, меняет состояние времени восстановления

**4 Код программы**

InfactionRate = Convert.ToInt32(textBox5.Text);

RecoveryTime = Convert.ToInt32(textBox6.Text);

Random rnd = new Random();

int Time = 0;

for (int j = 0; j < Nodes.Count; j++)

{

if (Nodes[j].I == true)

{

Nodes[j].RecoveryTime = RecoveryTime;

Nodes[j].CountR++;

}

}

timer1.Interval = Convert.ToInt32(textBox7.Text);

timer1.Start();

timer1.Tick += new EventHandler((o, ev) =>

{

listBox1.Items.Clear();

Time++;

for (int j = 0; j < Nodes.Count; j++)

{

if (Nodes[j].I == true)

{

for (int k = 0; k < Nodes[j].Near.Count; k++)

{

if (Nodes[j].Near[k].S == true && Nodes[j].Near[k].Iflag != 1 && Nodes[j].Near[k].Rem == false)

{

if (rnd.Next(0, 100) < InfactionRate)

{

Nodes[j].Near[k].Iflag = 1;

NI++;

NS--;

Nodes[j].Near[k].CountR++;

}

}

}

}

if (Nodes[j].RecoveryTime > 0)

{

Nodes[j].RecoveryTime -= 1;

}

else

{

if (Nodes[j].RecoveryTime == 0)

{

Nodes[j].CountTime = Time;

Nodes[j].RecoveryTime = -1;

Nodes[j].Rflag = 1;

NI--;

if (radioButton2.Checked)

NR++;

else if (radioButton1.Checked)

NS++;

}

}

}

SolidBrush mySolidBrush1 = new SolidBrush(Color.Red);

SolidBrush mySolidBrush2 = new SolidBrush(Color.Green);

SolidBrush mySolidBrush3 = new SolidBrush(Color.Yellow);

var a = CreateGraphics();

for (int j = 0; j < Nodes.Count; j++)

{

if (radioButton2.Checked)

{

if (Nodes[j].Iflag == 1)

{

Nodes[j].S = false;

Nodes[j].I = true;

Nodes[j].Iflag = 0;

Nodes[j].RecoveryTime = RecoveryTime;

a.FillEllipse(mySolidBrush1, (int)Nodes[j].X - 50 / m, (int)Nodes[j].Y - 50 / m, (50 / m) \* 2, (50 / m) \* 2);

}

if (Nodes[j].Rflag == 1)

{

Nodes[j].I = false;

Nodes[j].R = true;

Nodes[j].Rflag = 0;

a.FillEllipse(mySolidBrush2, (int)Nodes[j].X - 50 / m, (int)Nodes[j].Y - 50 / m, (50 / m) \* 2, (50 / m) \* 2);

}

}

else if (radioButton1.Checked)

{

if (Nodes[j].Iflag == 1)

{

Nodes[j].S = false;

Nodes[j].I = true;

Nodes[j].Iflag = 0;

Nodes[j].RecoveryTime = RecoveryTime;

a.FillEllipse(mySolidBrush1, (int)Nodes[j].X - 50 / m, (int)Nodes[j].Y - 50 / m, (50 / m) \* 2, (50 / m) \* 2);

}

if (Nodes[j].Rflag == 1)

{

Nodes[j].I = false;

Nodes[j].S = true;

Nodes[j].Rflag = 0;

a.FillEllipse(mySolidBrush3, (int)Nodes[j].X - 50 / m, (int)Nodes[j].Y - 50 / m, (50 / m) \* 2, (50 / m) \* 2);

}

}

}

mySolidBrush1.Dispose();

mySolidBrush2.Dispose();

mySolidBrush3.Dispose();

});

}

**5 Результат программы**

На рисунке 1 и 2 изображены итоги программы моделей SIS и SIR соответственно.

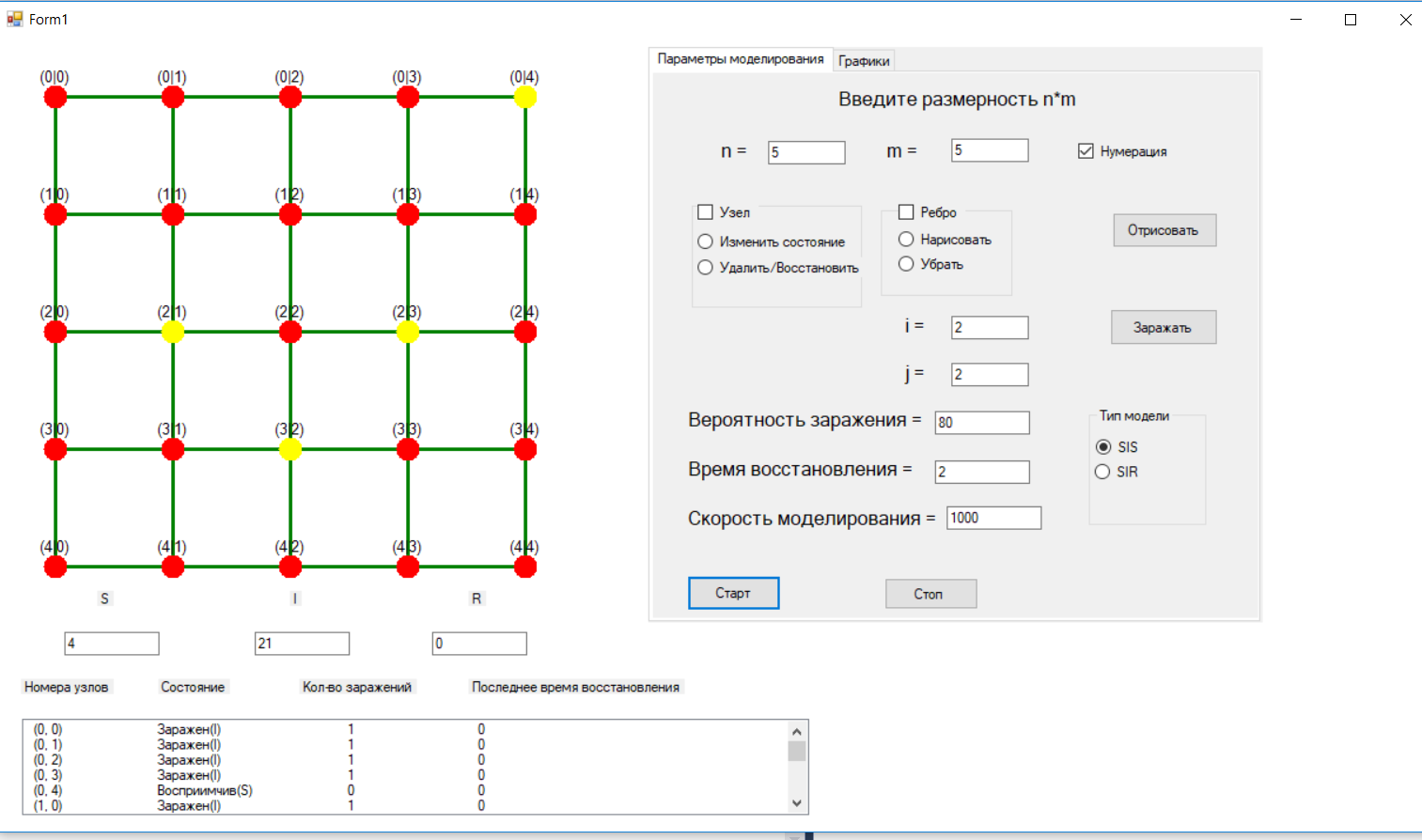


Рисунок 1 – SIS.

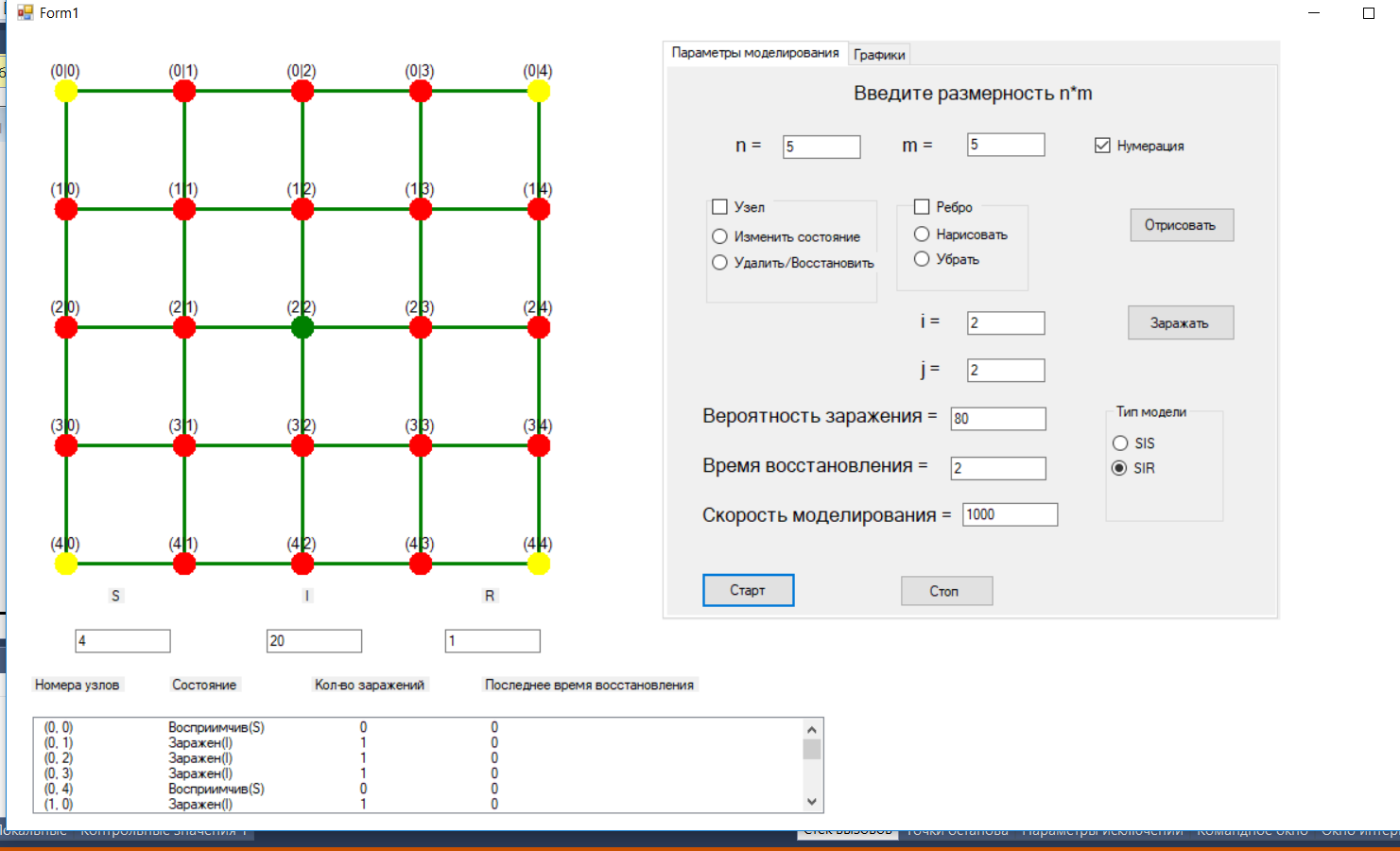


Рисунок 2 - SIR.

**Вывод:** на основе первой программы удалось смоделировать наглядное представления заражение узлов, научились работать с моделями SIS/SIR.