|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

«Определение среднего относительного времени пребывания системы в предельном стационарном состоянии»

Студент ИУ7-68Б(В) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Бахолдин

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель темы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Рудаков

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

Рассмотрим математическое описание марковского случайного процесса с дискретными состояниями и непрерывным временем. Будем полагать, что все переходы системы из состояния Si в состояние Sj происходят под воздействием простейших потоков событий с интенсивностями 𝜆 {i,j=l,2,3,4,5}.

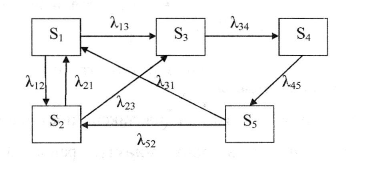
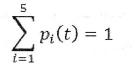
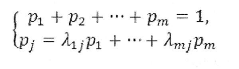


Рис.1 Граф состояний с интенсивностями переходов из состояния в состояние.

Вероятностью i-го состояния называется вероятность Pi(t) того, что в момент t система будет находиться в состоянии Si. Для любого момента времени t сумма всех вероятностей pi(t) равна единице:

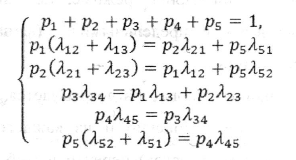


Вероятности системы pi(t) в предельном стационарном режиме, т.е. при t->∞, называются предельными (финальными) вероятностями состояний. В теории случайных процессов доказывается, что если число состояний системы конечно и из каждого из них можно (за конечное число шагов) перейти в любое другое состояние, то предельные вероятности существуют. Для определения предельных вероятностей необходимо решить систему линейных уравнений:



j = 1,2,3,…,m

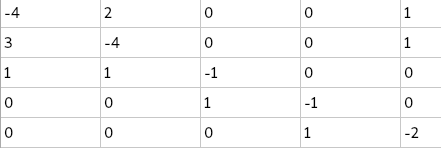
Рассмотрим Рис.1 Граф состояний, число состояний в которое может перейти система равно 5, поэтому нужно решить СЛАУ:



Предельная вероятность состояния Si имеет смысл: она показывает среднее относительное время пребывания системы в этом состоянии. Предположим что для графа на Рис.1 интенсивности заданы следующим образом :

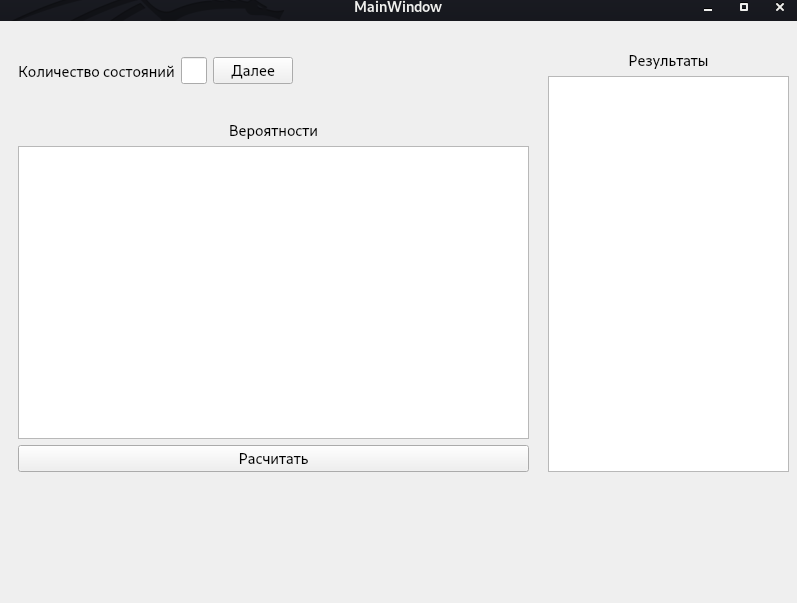


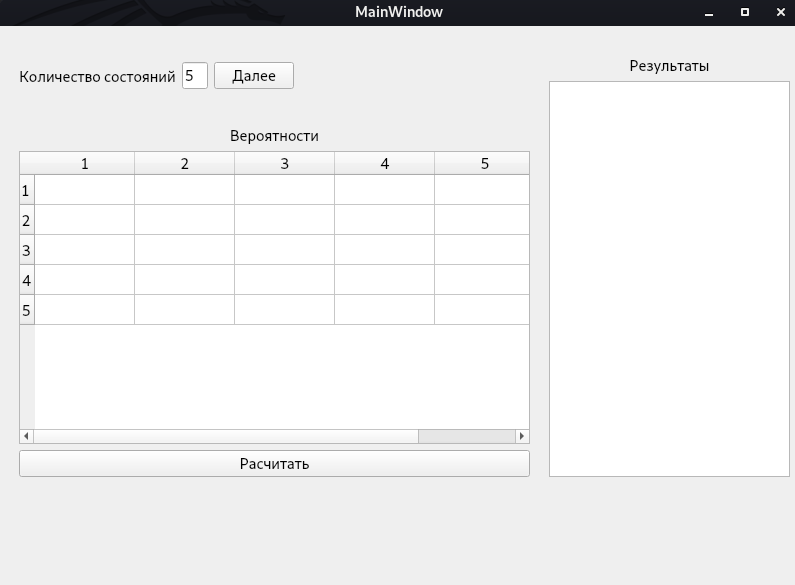
Тогда матрица интенсивности переходов будет выглядить следующим образом:



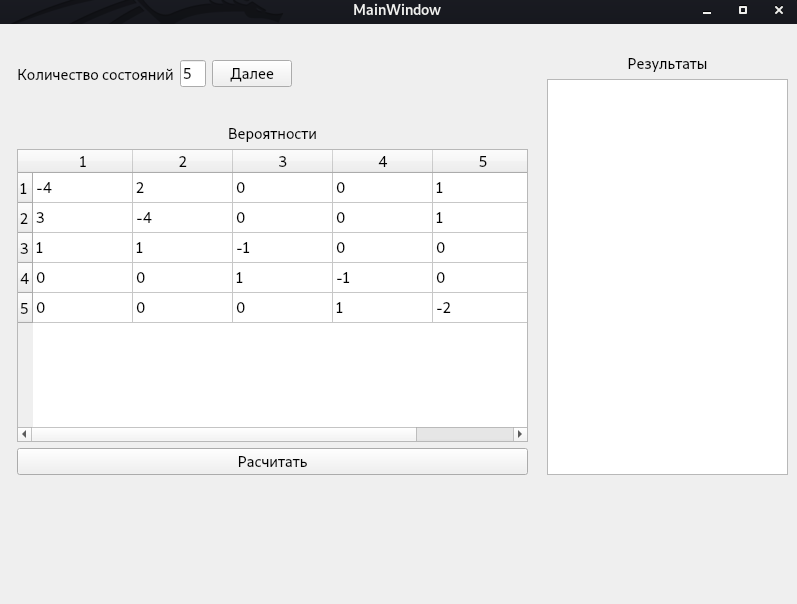
Для выполнения работы использовалась среда разработки Qt Creator 4.11.0 Based on Qt 5.12.5 ( GCC 9.2.1 20200104, 64 bit ), язык программирования Qt C++( стандарт языка С++17 ). Программа собрана под ОС Linux Debian 5.4.8-1kali1 ( 2020-01-06 ) x86\_64.

На первом шаге пользователю предлагаюется ввести количество состояний системы и нажать кнопку далее:

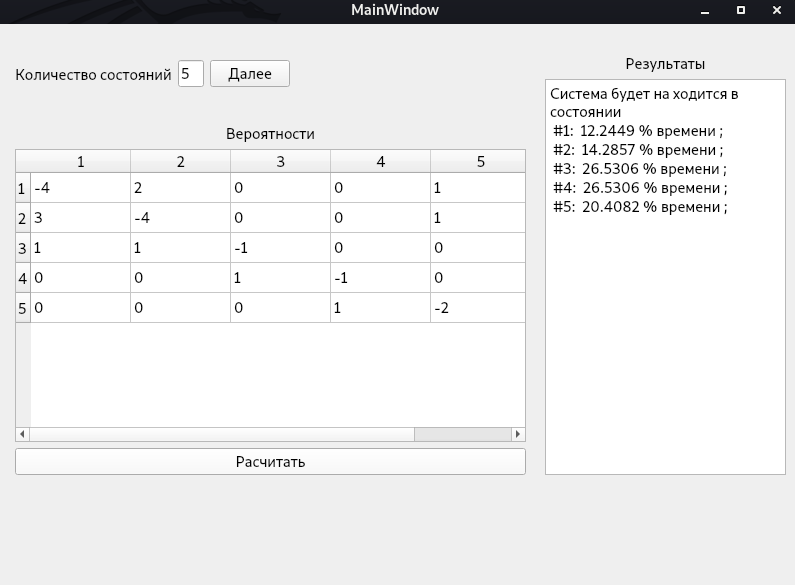




Далее пользователь вводит матрицу интенсивности и нажимает кнопку расчитать:



В окне результаты появляются результаты расчетов:



Листинг программы.

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include<QLineEdit>

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

*namespace* Ui { *class* MainWindow; }

QT\_END\_NAMESPACE

*class* MainWindow : *public* QMainWindow

{

Q\_OBJECT

*public*:

MainWindow(QWidget \*parent = *nullptr*);

~*MainWindow*();

*private* slots:

void on\_state\_next\_clicked();

void on\_state\_next\_2\_clicked();

*private*:

Ui::MainWindow \*ui;

};

#endif *//* *MAINWINDOW\_H*

***mainwindow.cpp***

*#include<QMessageBox>*

*#include<QDebug>*

*#include<QTableWidget>*

*#include<QTableView>*

*#include<QVariant>*

*#include<QModelIndex>*

*#include<QString>*

*#include<sstream>*

*#include "mainwindow.h"*

*#include "ui\_mainwindow.h"*

*#include "matrix.h"*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*MainWindow::MainWindow( QWidget \*parent )*

*: QMainWindow( parent ), ui(new Ui::MainWindow) {*

*ui->setupUi( this ) ;*

*}*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*MainWindow::~MainWindow() { delete ui ; }*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*void MainWindow::on\_state\_next\_clicked() {*

*int count\_state = (ui->txt\_count\_state->text()).toInt() ;*

*if( count\_state <= 0 || count\_state > 11 ) QMessageBox::information( NULL, QObject::tr( "Информация" ),*

*tr("Количество состояний должно быть больше 0 и меньше 11") ) ;*

*else {*

*ui->tableWidget->clear();*

*ui->tableWidget->setRowCount(count\_state);*

*ui->tableWidget->setColumnCount(count\_state);*

*ui->tableWidget->horizontalHeader();*

*ui->tableWidget->show();*

*}*

*}*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*void MainWindow::on\_state\_next\_2\_clicked() {*

*size\_t row = ui->tableWidget->rowCount() ;*

*size\_t col = ui->tableWidget->columnCount() ;*

*if( row <=0 || col <=0 ) QMessageBox::information( NULL, QObject::tr( "Информация" ),*

*tr("Сначало задайте количество состояний!") ) ;*

*else {*

*double \*\*v = new double \* [row] ;*

*for( size\_t i = 0; i < row; i++ ) v[i] = new double [col];*

*double \*y = new double[row];*

*QVariant myData ;*

*QModelIndex myIndex ;*

*for( size\_t i = 0 ; i < row ; ++i ) {*

*if( i == (row - 1) ) y[i] = 1 ;*

*else y[i] = 0 ;*

*for( size\_t j = 0 ; j < col ; ++j ) {*

*if( i == (row - 1) ) v[i][j] = 1 ;*

*else{*

*myIndex = ui->tableWidget->model()->index( i, j, QModelIndex() ) ;*

*myData = ui->tableWidget->model()->data( myIndex, Qt::DisplayRole ) ;*

*v[i][j] = myData.toDouble() ;*

*}*

*}*

*}*

*MTRX::Matrix< double > \*matrix = new MTRX::Matrix< double >( row, col, v, y ) ;*

*std::stringstream ss;*

*double \*x = matrix->GaussMeth();*

*/\*double res = 1 ;*

*for( size\_t i = 0 ; i < row ; i++ ) res -= x[i] ;*

*x[row-1] = res;\*/*

*ss << "Сиcтема будет на ходится в состоянии \n" ;*

*for( size\_t i = 0; i < col; i++ )*

*//ss << "x[" << i << "]=" << x[i] << " ; ";*

*ss <<" #" << i + 1<<": " << x[i] \* 100 <<" % времени"<< " ; \n" ;*

*QString s = QString::fromStdString(ss.str());*

*ui->textBrowser->setText( s ) ;*

*delete matrix ;*

*for( size\_t i = 0; i < row; i++ ) delete v[i] ;*

*delete [] v ;*

*}*

*}*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

***matrix.h***

*#ifndef MATRIX\_H*

*#define MATRIX\_H*

*#include<QVector>*

*#include<QString>*

*#include<algorithm>*

*namespace MTRX {*

*//--------------------------------------------------------------------------------------------------------------------//*

*template< class Tdata >*

*class Matrix {*

*private :*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*Tdata \*\*matrix ;*

*Tdata \*y ;*

*size\_t row ;*

*size\_t col ;*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*public :*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*Matrix() = delete ;*

*Matrix( Matrix< Tdata > &\_copy ) = delete ;*

*Matrix< Tdata >& operator=( Matrix< Tdata > &matrix ) = delete ;*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*Matrix( const size\_t &row, const size\_t &col, Tdata \*\*v, Tdata \*y )*

*: matrix( v ), y(y), row( row ), col( col ) {}*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*virtual ~Matrix() { }*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*double \* GaussMeth() {*

*double \*\*a = this->matrix ;*

*double \*y = this->y ;*

*int n = col ;*

*double \*x, max ;*

*int k, index ;*

*const double eps = 0.01 ; // точность*

*x = new double[n] ;*

*k = 0 ;*

*while( k < n ) {*

*max = abs( a[k][k] ) ;*

*index = k ;*

*for( int i = k + 1 ; i < n ; i++ ) {*

*if( abs(a[i][k]) > max ) {*

*max = abs( a[i][k] ) ;*

*index = i ;*

*}*

*}*

*for( int j = 0 ; j < n ; j++ ) {*

*double temp = a[k][j] ;*

*a[k][j] = a[index][j] ;*

*a[index][j] = temp ;*

*}*

*double temp = y[k] ;*

*y[k] = y[index] ;*

*y[index] = temp ;*

*for( int i = k ; i < n ; i++ ) {*

*double temp = a[i][k] ;*

*if( abs(temp) < eps ) continue ;*

*for( int j = 0 ; j < n ; j++ ) a[i][j] = a[i][j] / temp ;*

*y[i] = y[i] / temp ;*

*if( i == k ) continue ;*

*for(int j = 0; j < n; j++) a[i][j] = a[i][j] - a[k][j] ;*

*y[i] = y[i] - y[k] ;*

*}*

*k++ ;*

*} // while*

*for( k = n - 1 ; k >= 0 ; k-- ) {*

*x[k] = y[k] ;*

*for( int i = 0 ; i < k ; i++) y[i] = y[i] - a[i][k] \* x[k] ;*

*}*

*return x;*

*}*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//*

*} ;*

*//--------------------------------------------------------------------------------------------------------------------//*

*} // namespace MTRX*

*#endif // MATRIX\_H*

***main.cpp***

*#include "mainwindow.h"*

*#include <QApplication>*

*int main(int argc, char \*argv[])*

*{*

*QApplication a(argc, argv);*

*MainWindow w;*

*w.show();*

*return a.exec();*

*}*

***mainwindow.ui***

*<?xml version="1.0" encoding="UTF-8"?>*

*<ui version="4.0">*

*<class>MainWindow</class>*

*<widget class="QMainWindow" name="MainWindow">*

*<property name="geometry">*

*<rect>*

*<x>0</x>*

*<y>0</y>*

*<width>800</width>*

*<height>600</height>*

*</rect>*

*</property>*

*<property name="windowTitle">*

*<string>MainWindow</string>*

*</property>*

*<widget class="QWidget" name="centralwidget">*

*<widget class="QWidget" name="horizontalLayoutWidget">*

*<property name="geometry">*

*<rect>*

*<x>20</x>*

*<y>10</y>*

*<width>275</width>*

*<height>80</height>*

*</rect>*

*</property>*

*<layout class="QHBoxLayout" name="horizontalLayout">*

*<item>*

*<widget class="QLabel" name="label">*

*<property name="text">*

*<string>Количество состояний</string>*

*</property>*

*</widget>*

*</item>*

*<item>*

*<widget class="QLineEdit" name="txt\_count\_state"/>*

*</item>*

*<item>*

*<widget class="QPushButton" name="state\_next">*

*<property name="text">*

*<string>Далее</string>*

*</property>*

*</widget>*

*</item>*

*</layout>*

*</widget>*

*<widget class="QWidget" name="verticalLayoutWidget">*

*<property name="geometry">*

*<rect>*

*<x>20</x>*

*<y>100</y>*

*<width>511</width>*

*<height>351</height>*

*</rect>*

*</property>*

*<layout class="QVBoxLayout" name="verticalLayout">*

*<item>*

*<widget class="QLabel" name="label\_2">*

*<property name="text">*

*<string>Вероятности</string>*

*</property>*

*<property name="alignment">*

*<set>Qt::AlignCenter</set>*

*</property>*

*</widget>*

*</item>*

*<item>*

*<widget class="QTableWidget" name="tableWidget"/>*

*</item>*

*<item>*

*<widget class="QPushButton" name="state\_next\_2">*

*<property name="text">*

*<string>Расчитать</string>*

*</property>*

*</widget>*

*</item>*

*</layout>*

*</widget>*

*<widget class="QWidget" name="verticalLayoutWidget\_2">*

*<property name="geometry">*

*<rect>*

*<x>550</x>*

*<y>30</y>*

*<width>241</width>*

*<height>421</height>*

*</rect>*

*</property>*

*<layout class="QVBoxLayout" name="verticalLayout\_2">*

*<item>*

*<widget class="QLabel" name="label\_3">*

*<property name="text">*

*<string>Результаты</string>*

*</property>*

*<property name="alignment">*

*<set>Qt::AlignCenter</set>*

*</property>*

*</widget>*

*</item>*

*<item>*

*<widget class="QTextBrowser" name="textBrowser"/>*

*</item>*

*</layout>*

*</widget>*

*</widget>*

*<widget class="QMenuBar" name="menubar">*

*<property name="geometry">*

*<rect>*

*<x>0</x>*

*<y>0</y>*

*<width>800</width>*

*<height>24</height>*

*</rect>*

*</property>*

*</widget>*

*<widget class="QStatusBar" name="statusbar"/>*

*</widget>*

*<resources/>*

*<connections/>*

*</ui>*