Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №20**

Дисциплина: «Основы теории алгоритмов и структуры данных»  
Тема: Графы

Вариант 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2021

**Цель работы**

1) Получить практические навыки по работе с графами и алгоритмом Дейкстры.

2) Продемонстрировать работу с графическим интерфейсом средствами Qt при выводе графа.

**Постановка задачи**

1) Реализовать алгоритм Дейкстры для графа, состоящего из шести узлов.

2) Визуализировать граф и результат применения алгоритма Дейкстры.

**Анализ задачи**

Для реализации задачи необходимо:

1. Создать двумерный размерностью 6х6 и заполнить его значениями из варианта.
2. Реализовать алгоритм Дейкстры.
3. Визуализировать граф.

**Реализация задачи на базе Qt**

1. Заполнение массива:

struct **Uzel**{

int \*way;

bool take=0;

int from=1000;

};

Uzel uzel[6];

Graf::**Graf**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::Graf)

{

ui->setupUi(this);

ui->button->setShortcut(Qt::Key\_Return);

uzel[0].way= new int[6]{0,2,0,0,0,57};

uzel[1].way= new int[6]{2,0,3,8,0,13};

uzel[2].way= new int[6]{0,3,0,5,0,0};

uzel[3].way= new int[6]{0,8,5,0,34,21};

uzel[4].way= new int[6]{0,0,0,34,0,45};

uzel[5].way= new int[6]{57,13,0,21,45,0};

1. Функция находящая кратчайшее расстояние между узлами с помощью алгоритма Дейкстры:

void **Process**(Uzel &cur){

int mini=1000;

for (int i=0;i<6;i++){

if (cur.way[i]&&cur.way[i]+cur.from<uzel[i].from) uzel[i].from=cur.way[i]+cur.from;

if (cur.way[i]&&!uzel[i].take&&cur.way[i]<mini) mini=cur.way[i];

}

cur.take=1;

for (int i=0;i<6;i++) if (cur.way[i]==mini&&!uzel[i].take)Process(*uzel[i]*);

}

1. Функция, рисующая граф в окне программы:

void Graf::***paintEvent***(QPaintEvent \*)

{

QPainter paint;

QPen pen;

pen.setColor(Qt::green);

pen.setWidth(5);

paint.begin(this);

paint.setPen(pen);

paint.drawEllipse(200,10,80,80);

paint.drawEllipse(50,300,80,80);

paint.drawEllipse(20,500,80,80);

paint.drawEllipse(400,600,80,80);

paint.drawEllipse(900,400,80,80);

paint.drawEllipse(800,100,80,80);

paint.drawLine(240,90,130,340);

paint.drawLine(240,90,800,140);

paint.drawLine(130,340,100,540);

paint.drawLine(130,340,440,600);

paint.drawLine(130,340,800,140);

paint.drawLine(100,540,440,600);

paint.drawLine(440,600,900,440);

paint.drawLine(440,600,800,140);

paint.drawLine(800,140,900,440);

paint.end();

}

**Полный код программы**

1. Заголовочный файл:

#ifndef GRAF\_H

#define GRAF\_H

#include <QPainter>

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **Graf**; }

QT\_END\_NAMESPACE

class **Graf** : public QMainWindow

{

Q\_OBJECT

public:

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

~***Graf***();

private:

Ui::Graf \*ui;

protected:

void ***paintEvent***(QPaintEvent\*) override;

private slots:

void **on\_button\_clicked**();

};

#endif // GRAF\_H

1. Файл-источник:

#include "graf.h"

#include "ui\_graf.h"

struct **Uzel**{

int \*way;

bool take=0;

int from=1000;

};

Uzel uzel[6];

Graf::**Graf**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::Graf)

{

ui->setupUi(this);

ui->button->setShortcut(Qt::Key\_Return);

uzel[0].way= new int[6]{0,2,0,0,0,57};

uzel[1].way= new int[6]{2,0,3,8,0,13};

uzel[2].way= new int[6]{0,3,0,5,0,0};

uzel[3].way= new int[6]{0,8,5,0,34,21};

uzel[4].way= new int[6]{0,0,0,34,0,45};

uzel[5].way= new int[6]{57,13,0,21,45,0};

}

Graf::~***Graf***()

{

delete ui;

}

void **Process**(Uzel &cur){

int mini=1000;

for (int i=0;i<6;i++){

if (cur.way[i]&&cur.way[i]+cur.from<uzel[i].from) uzel[i].from=cur.way[i]+cur.from;

if (cur.way[i]&&!uzel[i].take&&cur.way[i]<mini) mini=cur.way[i];

}

cur.take=1;

for (int i=0;i<6;i++) if (cur.way[i]==mini&&!uzel[i].take)Process(*uzel[i]*);

}

void Graf::***paintEvent***(QPaintEvent \*)

{

QPainter paint;

QPen pen;

pen.setColor(Qt::green);

pen.setWidth(5);

paint.begin(this);

paint.setPen(pen);

paint.drawEllipse(200,10,80,80);

paint.drawEllipse(50,300,80,80);

paint.drawEllipse(20,500,80,80);

paint.drawEllipse(400,600,80,80);

paint.drawEllipse(900,400,80,80);

paint.drawEllipse(800,100,80,80);

paint.drawLine(240,90,130,340);

paint.drawLine(240,90,800,140);

paint.drawLine(130,340,100,540);

paint.drawLine(130,340,440,600);

paint.drawLine(130,340,800,140);

paint.drawLine(100,540,440,600);

paint.drawLine(440,600,900,440);

paint.drawLine(440,600,800,140);

paint.drawLine(800,140,900,440);

paint.end();

}

void Graf::**on\_button\_clicked**()

{

for (int i=0;i<6;i++){

uzel[i].from=1000;

uzel[i].take=0;

}

int k = ui->line->text().toInt()-1;

if (k<0||k>5){

ui->line->setText("");

ui->statusbar->showMessage("Номер вершины должен быть в диапазоне от 1 до 6! ");

}

else{

ui->statusbar->showMessage("");

uzel[k].from=0;

Process(*uzel[k]*);

QString s = QString::number(uzel[0].from);

ui->u->setText(s);

s = QString::number(uzel[1].from);

ui->u\_2->setText(s);

s = QString::number(uzel[2].from);

ui->u\_3->setText(s);

s = QString::number(uzel[3].from);

ui->u\_4->setText(s);

s = QString::number(uzel[4].from);

ui->u\_5->setText(s);

s = QString::number(uzel[5].from);

ui->u\_6->setText(s);

}

}

1. Файл, содержащий функцию main():

#include "graf.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

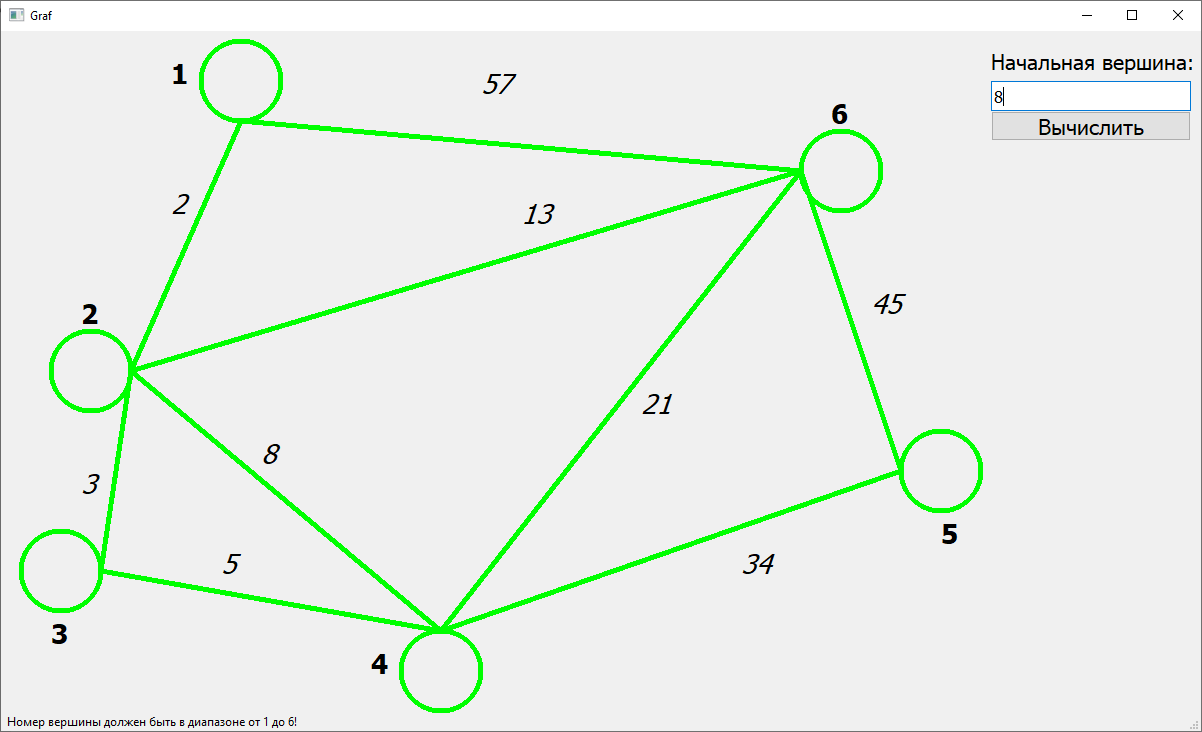
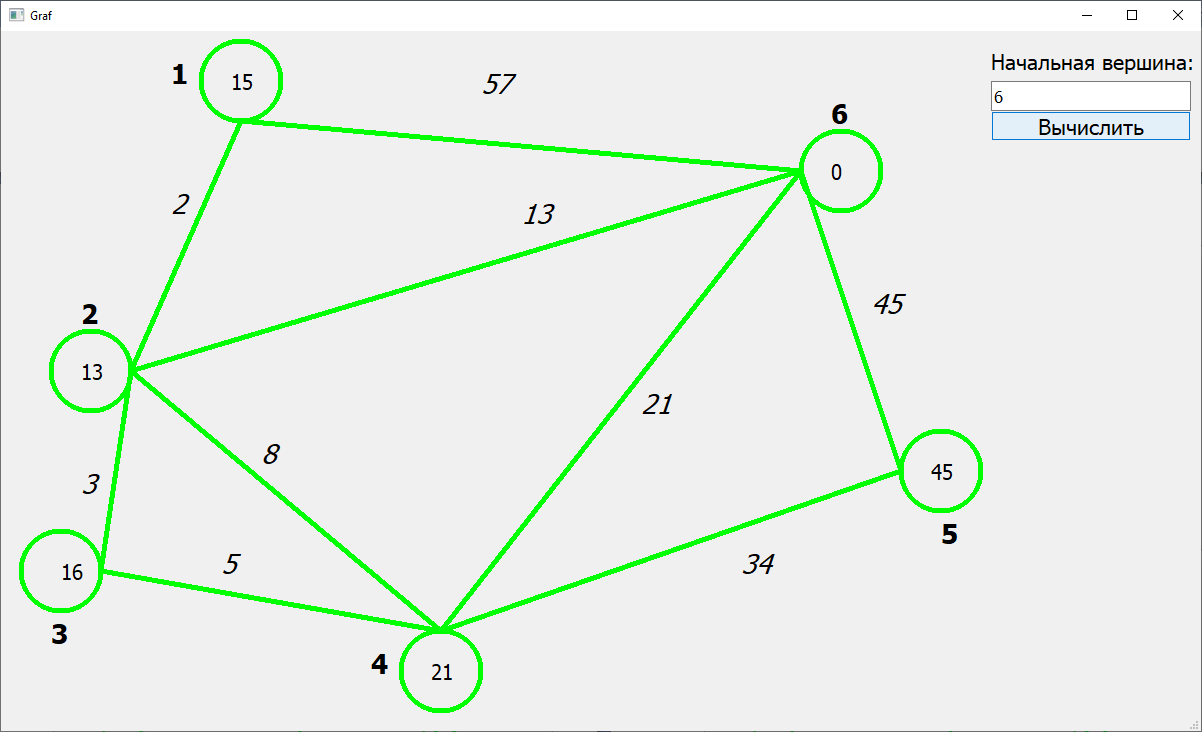
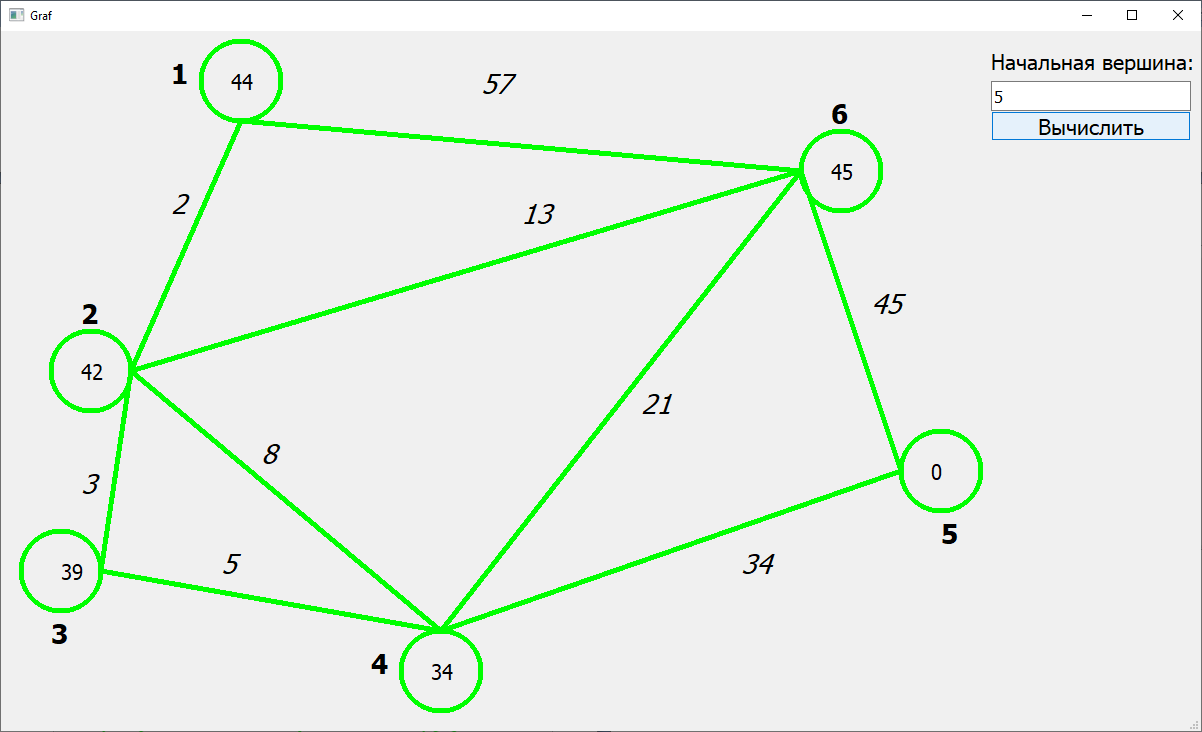
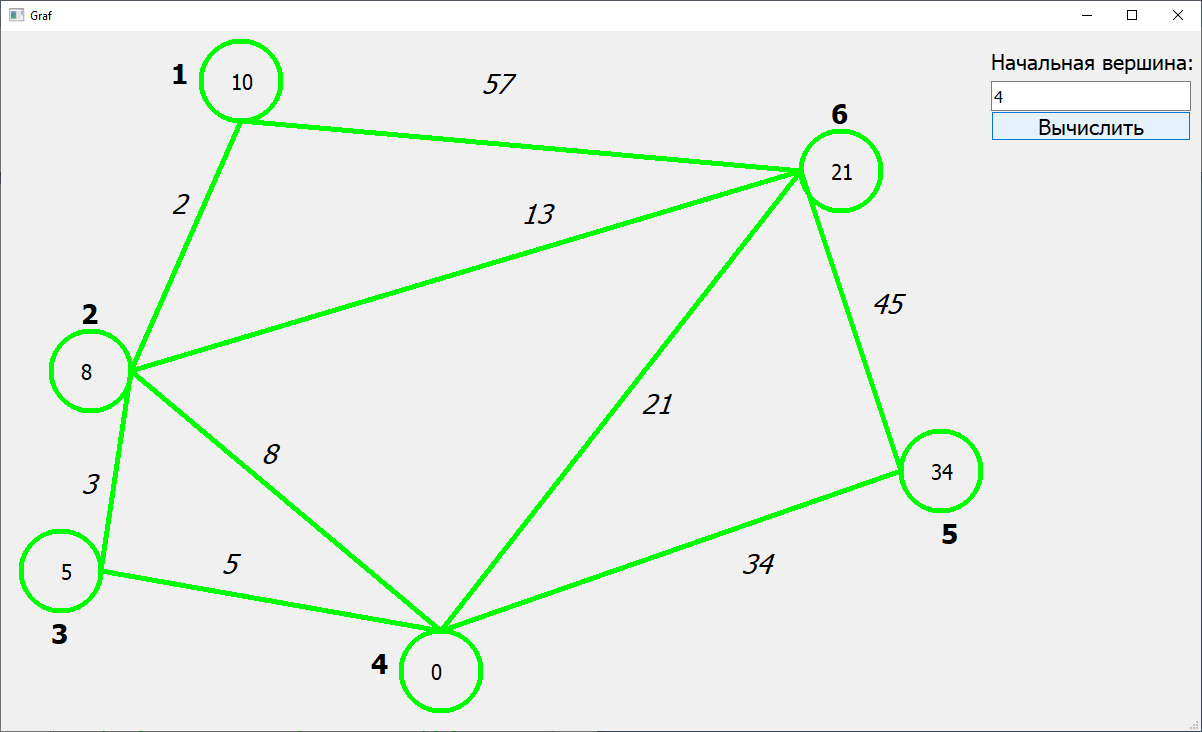
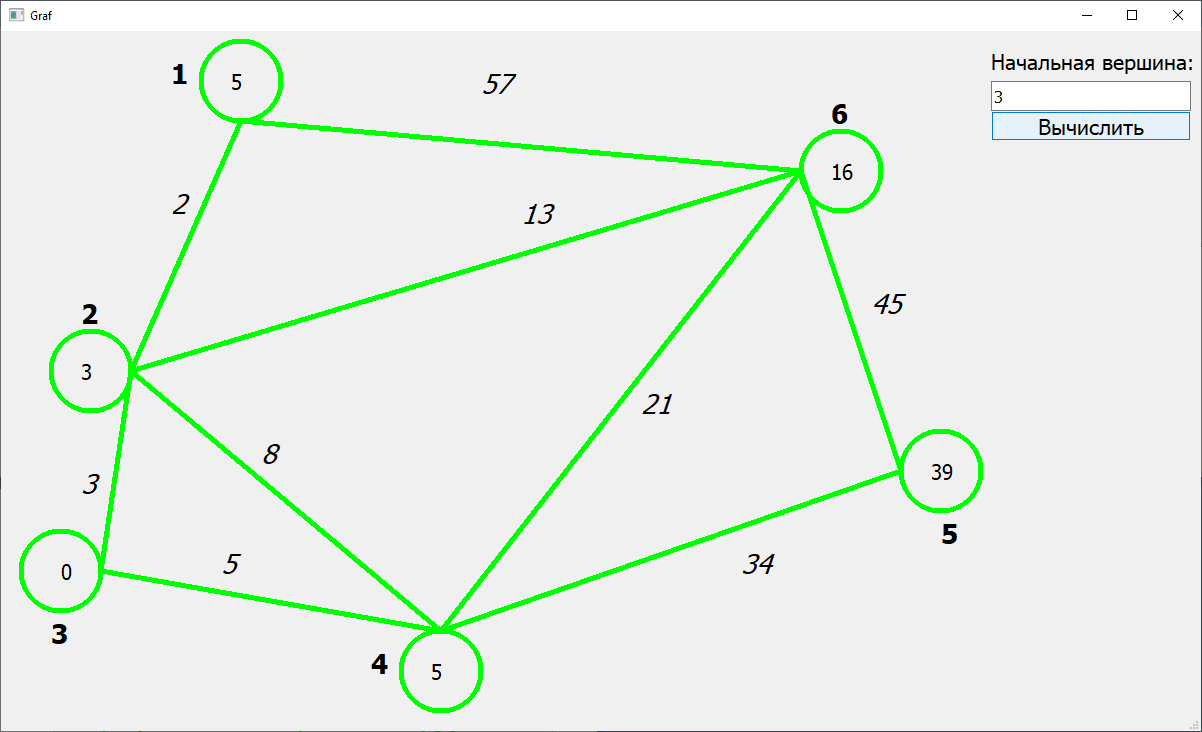
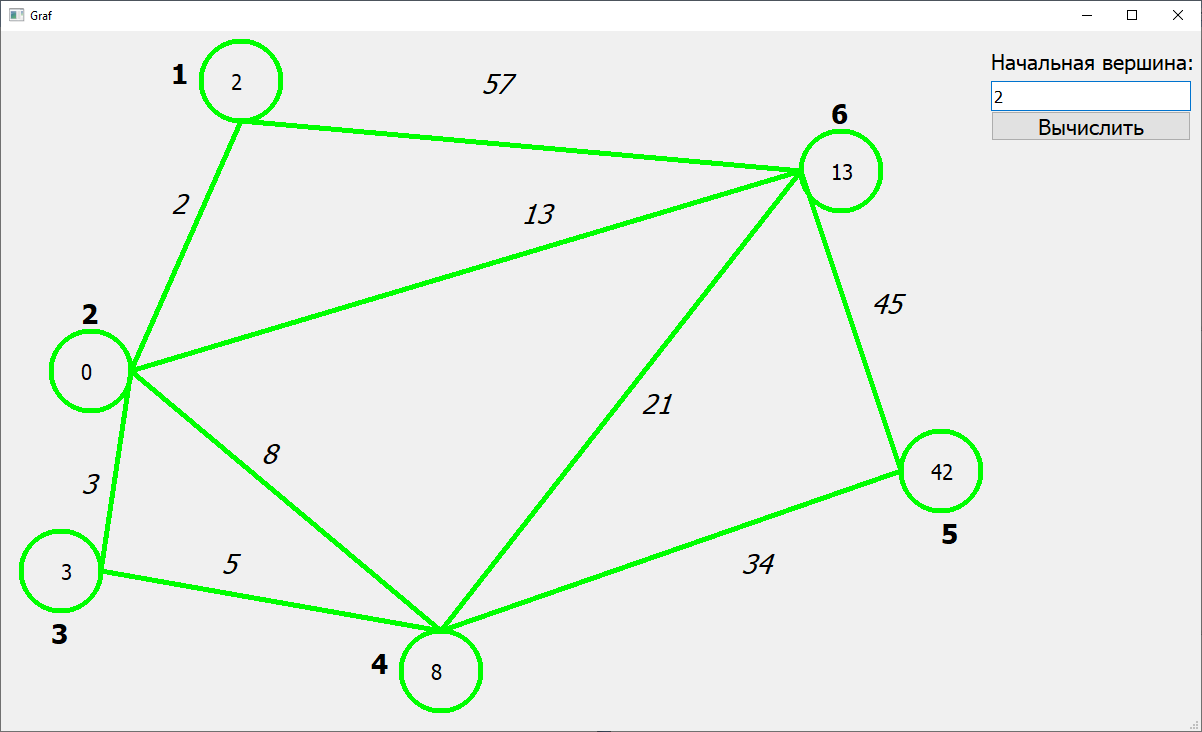
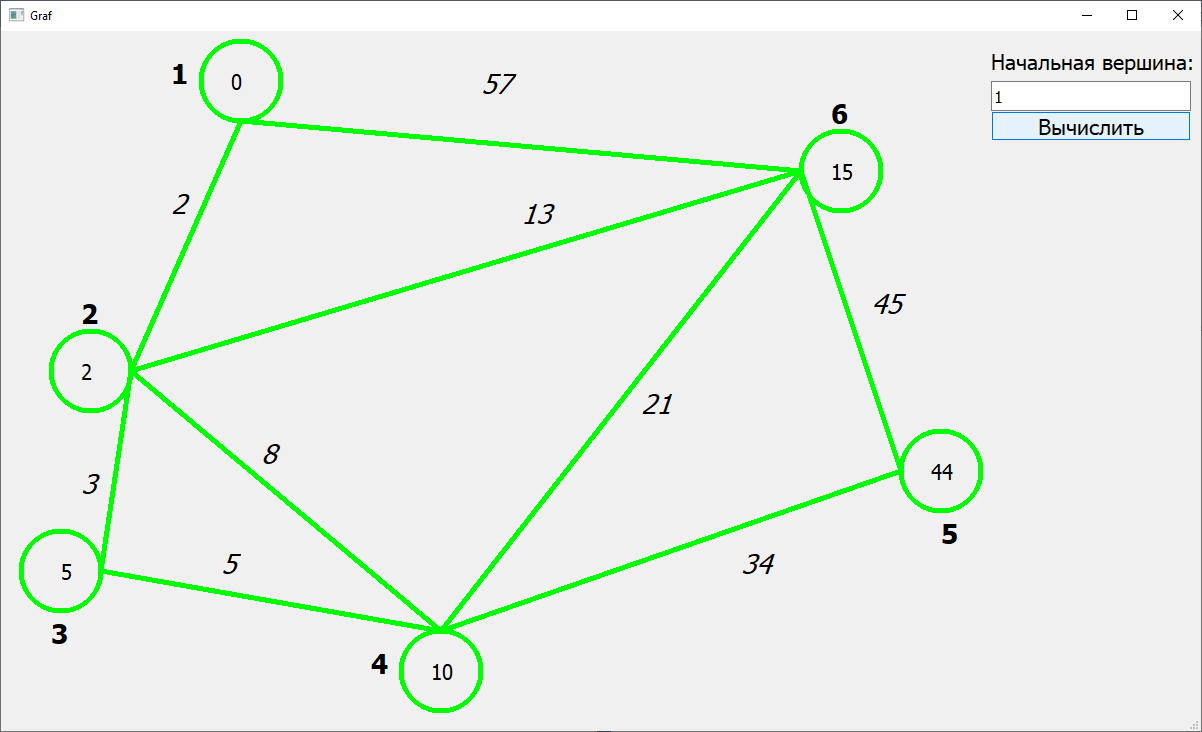
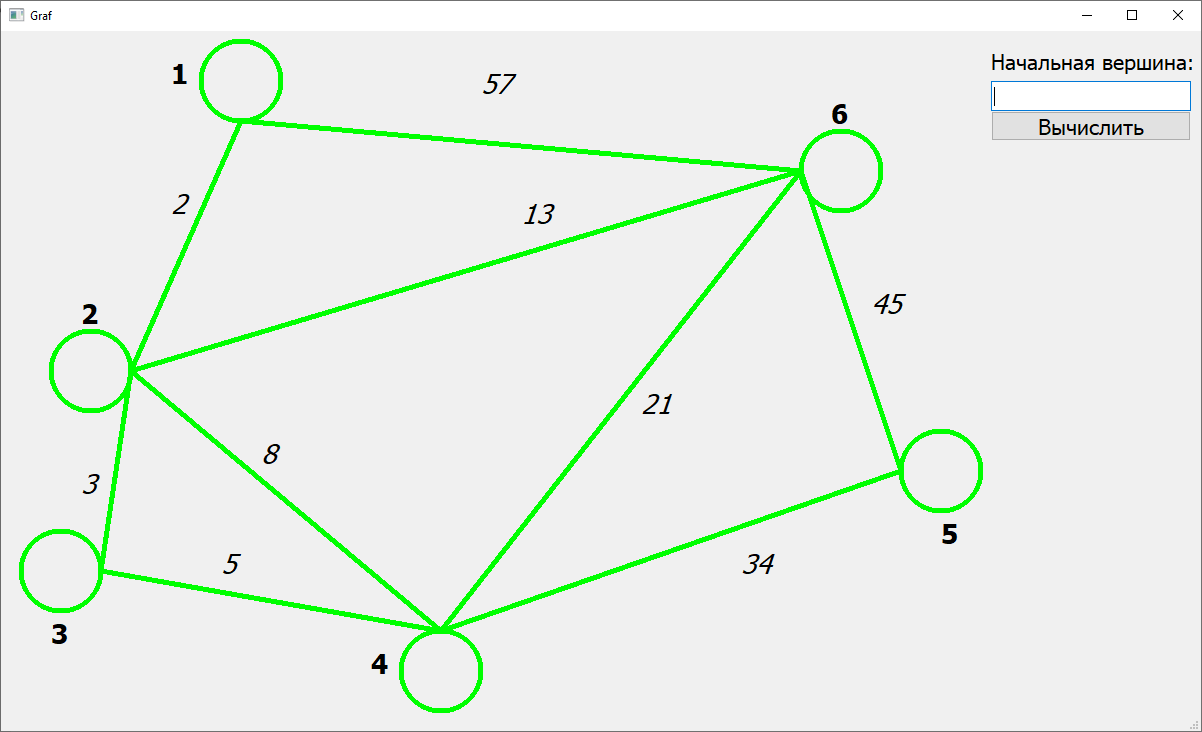
Graf w;

w.show();

return a.exec();

}

**Результат работы программы**



**Диаграмма классов**

