Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

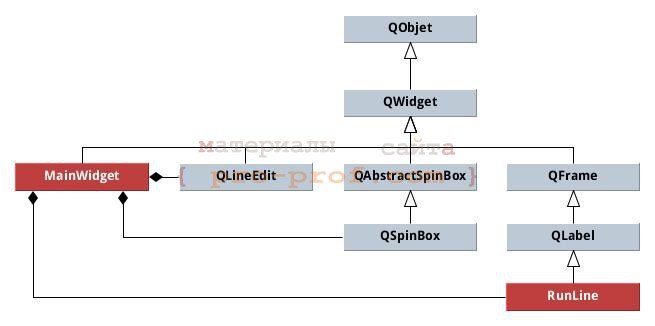
Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

Создать приложение калькулятор и сделать задачу коммивояжера.

Анализ задачи калькулятор:

1. Применены стандартные математические операции, конвертация чисел в строки, и строки в числа. Запись значений в label.
2. В качестве инструмента визуализации выбрана платформа QT.
3. Пришлось столкнуться с трудностями конвертации данных в другой тип.
4. В данной работе автор гордится дизайном и тем, что калькулятор может считать различные компоненты кредита.
5. Визуализация сделана при помощи стандартных шаблонных кнопок и label-ов, а дизайн наследовался от заднего фона. Первая ассоциация, связанная с денежными средствами – мистер Крабс из Губки Боба.
6. 

**Анализ задачи коммивояжера:**

1. Применен метод ветвей и границ для нахождения кратчайшего маршрута.

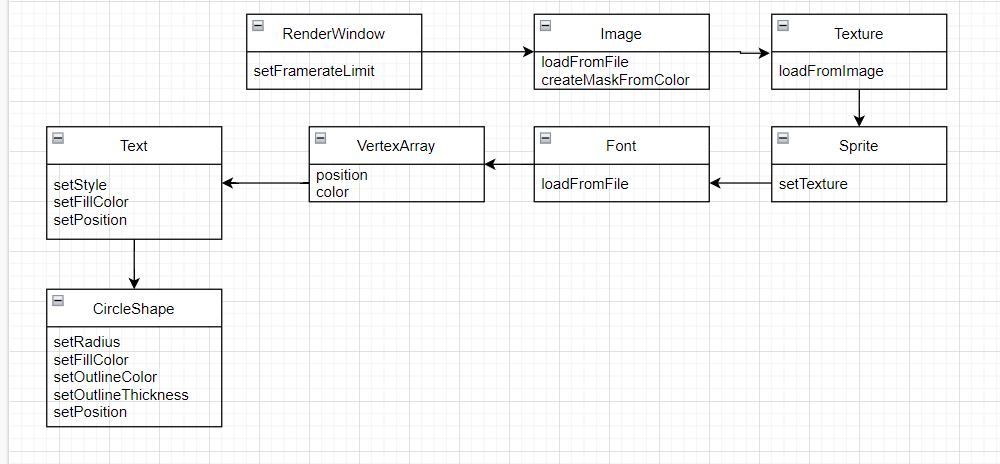
2. Инструмент визуализации – sfml.

3. Трудности возникали с оценкой нулей после редуцирования строк и столбцов. Также возникли трудности с визуальным представлением графа и обработкой событий.

4. К достоинствам следует отнести простоту.

5. Визуализация сделана с помощью стандартных классов фигур пространства имен sf.

6. Дизайн выбран менее яркий, самый простой.

7. 

Код программы калькулятор:

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include<QString>

#include<QPushButton>

#include <QMessageBox>

#include<QPixmap>

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

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~**MainWindow**()

{

delete ui;

}

void MainWindow::**on\_Calculate\_clicked**()

{

QString **tmp**;

double **Stavka**,**Sum**,**Months**;double **Pay\_Months**;

tmp=ui->Stavka->text(); Stavka=tmp.toDouble();

tmp=ui->Sum->text();Sum=tmp.toDouble();

tmp=ui->Months->text();Months=tmp.toDouble();

Stavka=(Stavka/100)+1;

Pay\_Months=((Sum\*Stavka)/(Months));

QString **new\_Pay**;

new\_Pay.setNum(Pay\_Months);

ui->Pay->setText(new\_Pay);

}

void MainWindow::**on\_Delete\_all\_clicked**()

{

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

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

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

ui->Pay->clear();

}

void MainWindow::**on\_plata\_clicked**()

{

ui->label\_SUM->setText("Сумма кредита(руб): ");

ui->label\_ST->setText("Годовая ставка(%)");

ui->label\_SROK->setText("На какой срок(мес): ");

ui->label\_ITOG->setText("Ежемесячный платеж: ");

}

void MainWindow::**on\_stavvvka\_clicked**()

{

ui->label\_SUM->setText("Сумма кредита(руб): ");

ui->label\_ST->setText("Ежемесячный платеж: ");

ui->label\_SROK->setText("На какой срок(мес): ");

ui->label\_ITOG->setText("Годовая ставка(%)");

}

void MainWindow::**on\_Calculate\_2\_clicked**()

{

QMessageBox::information(this,"Всплывающее окно", "Пасхалка найдена!");

}

void MainWindow::**on\_Summa\_clicked**()

{

ui->label\_SUM->setText("Ежемесячный платеж:");

ui->label\_ST->setText("Годовая ставка(%)");

ui->label\_SROK->setText("На какой срок(мес): ");

ui->label\_ITOG->setText("Сумма кредита(руб):");

}

void MainWindow::**on\_Srok\_clicked**()

{

ui->label\_SUM->setText("Сумма кредита(руб): ");

ui->label\_ST->setText("Годовая ставка(%)");

ui->label\_SROK->setText("Ежемесячный платеж: ");

ui->label\_ITOG->setText("На какой срок(мес): ");

}

void MainWindow::**on\_Calculate\_ST\_clicked**()

{

QString **tmp**;

double **Stavka**,**Sum**,**Months**, **Pay\_Months**;

tmp=ui->Stavka->text(); Stavka=tmp.toDouble();

tmp=ui->Sum->text();Sum=tmp.toDouble();

tmp=ui->Months->text();Months=tmp.toDouble();

Pay\_Months=(((Stavka\*Months)/Sum)-1)\*100;

QString **new\_Pay**;

new\_Pay.setNum(Pay\_Months);

ui->Pay->setText(new\_Pay);

}

void MainWindow::**on\_Calculate\_SUM\_2\_clicked**()

{

QString **tmp**;

double **Sum**,**Months**, **Pay\_Months**;

tmp=ui->Sum->text();Sum=tmp.toDouble();

tmp=ui->Months->text();Months=tmp.toDouble();

Pay\_Months=Sum\*Months;

QString **new\_Pay**;

new\_Pay.setNum(Pay\_Months);

ui->Pay->setText(new\_Pay);

}

void MainWindow::**on\_Calculate\_srok\_clicked**()

{

QString **tmp**;

double **Stavka**,**Sum**,**Months**, **Pay\_Months**;

tmp=ui->Stavka->text(); Stavka=tmp.toDouble();

tmp=ui->Sum->text();Sum=tmp.toDouble();

tmp=ui->Months->text();Months=tmp.toDouble();

Stavka=(Stavka)/100;

Pay\_Months=((Stavka\*Sum)+Sum)/Months;

QString **new\_Pay**;

new\_Pay.setNum(Pay\_Months);

ui->Pay->setText(new\_Pay);

}

Код программы коммивояжер:

#include<iostream>

#include<string>

#include <SFML/Graphics.hpp>

using namespace std;

string st;

struct pare

{

int main;

int dop;

int os\_x;

int os\_y;

};

int made\_arr(int(&arr)[9][8])

{

arr[0][0] = -1;

arr[0][1] = 6;

arr[0][2] = 6;

arr[0][3] = 4;

arr[0][4] = 10;

arr[0][5] = -1;

arr[0][6] = -1;

arr[0][7] = 0;

arr[1][0] = 4;

arr[1][1] = -1;

arr[1][2] = -1;

arr[1][3] = -1;

arr[1][4] = 2;

arr[1][5] = -1;

arr[1][6] = -1;

arr[1][7] = 0;

arr[2][0] = 8;

arr[2][1] = -1;

arr[2][2] = -1;

arr[2][3] = 2;

arr[2][4] = -1;

arr[2][5] = 1;

arr[2][6] = -1;

arr[2][7] = 0;

arr[3][0] = 2;

arr[3][1] = -1;

arr[3][2] = 1;

arr[3][3] = -1;

arr[3][4] = -1;

arr[3][5] = 10;

arr[3][6] = -1;

arr[3][7] = 0;

arr[4][0] = 7;

arr[4][1] = 5;

arr[4][2] = -1;

arr[4][3] = -1;

arr[4][4] = -1;

arr[4][5] = 2;

arr[4][6] = 7;

arr[4][7] = 0;

arr[5][0] = -1;

arr[5][1] = -1;

arr[5][2] = 0;

arr[5][3] = 8;

arr[5][4] = 3;

arr[5][5] = -1;

arr[5][6] = 2;

arr[5][7] = 0;

arr[6][0] = -1;

arr[6][1] = -1;

arr[6][2] = -1;

arr[6][3] = -1;

arr[6][4] = 4;

arr[6][5] = 6;

arr[6][6] = -1;

arr[6][7] = 0;

arr[7][0] = 0;

arr[7][1] = 0;

arr[7][2] = 0;

arr[7][3] = 0;

arr[7][4] = 0;

arr[7][5] = 0;

arr[7][6] = 0;

arr[7][7] = 0;

arr[8][7] = 0;

return arr[9][8];

}

void show\_arr(int(&arr)[9][8])

{

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

if (arr[i][j] < 0)

{

cout << "M";

cout << "\t";

}

else

{

cout << arr[i][j];

cout << "\t";

}

}

cout << endl;

}

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

}

int min\_line(int(&arr)[9][8], int i)

{

int min = 100;

for (int j = 0; j < 7; j++)

{

if ((arr[i][j] < min) && (arr[i][j] >= 0))

{

min = arr[i][j];

}

}

return min;

}

int min\_column(int(&arr)[9][8], int j)

{

int min = 100;

for (int i = 0; i < 7; i++)

{

if ((arr[i][j] < min) && (arr[i][j] >= 0))

{

min = arr[i][j];

}

}

return min;

}

int reduction\_lines(int(&arr)[9][8])

{

for (int i = 0; i < 7; i++)

{

arr[i][7] = min\_line(arr, i);

}

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

arr[i][j] = arr[i][j] - arr[i][7];

}

}

return arr[9][8];

}

int reduction\_columns(int(&arr)[9][8])

{

for (int i = 0; i < 7; i++)

{

arr[7][i] = min\_column(arr, i);

}

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

arr[i][j] = arr[i][j] - arr[7][j];

}

}

return arr[9][8];

}

int mark(int(&arr)[9][8], int a, int b)

{

int sum;

int line = a;

int column = b;

int min\_line = 100;

int min\_column = 100;

for (int i = 0; i < 7; i++)

{

if ((arr[line][i] < min\_line) && (i != b) && (arr[line][i] >= 0))

{

min\_line = arr[line][i];

}

}

for (int i = 0; i < 7; i++)

{

if ((arr[i][column] < min\_column) && (i != a) && (arr[i][column] >= 0))

{

min\_column = arr[i][column];

}

}

sum = min\_column + min\_line;

return sum;

}

int reduction\_arr(int(&arr)[9][8], int(&arr1)[9][8])

{

int result;

for (int i = 0; i < 7; i++)

{

int a;

int b;

int temp1 = 0;

int temp2 = 0;

reduction\_lines(arr);

reduction\_columns(arr);

pare mas[8][8];

int max\_secret = 0;

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

mas[i][j].main = arr[i][j];

mas[i][j].dop = 0;

}

}

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

if (arr[i][j] == 0)

{

mas[i][j].dop = mark(arr, i, j);

}

}

}

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 7; j++)

{

if (mas[i][j].main == 0)

{

if (mas[i][j].dop > max\_secret)

{

max\_secret = mas[i][j].dop;

temp1 = i;

temp2 = j;

}

}

}

}

for (int i = 0; i < 7; i++)

{

arr[temp1][i] = -1;

arr[i][temp2] = -1;

}

arr[temp2][temp1] = -1;

show\_arr(arr);

for (int i = 0; i < 8; i++)

{

arr[7][i] = 0;

}

for (int i = 0; i < 8; i++)

{

arr[i][7] = 0;

}

cout << temp1 + 1 << "->" << temp2 + 1 << endl;

arr[8][7] += arr1[temp1][temp2];

}

st = to\_string(arr[8][7]);

return arr[8][7];

}

int main()

{

setlocale(LC\_ALL, "rus");

sf::RenderWindow window(sf::VideoMode(800, 600), "Salesman problem");

window.setFramerateLimit(60);

sf::Image image;

image.loadFromFile("C:/Users/79024/Desktop/Fon.jpg");

image.createMaskFromColor(image.getPixel(0, 0));

sf::Texture texture;

texture.loadFromImage(image);

sf::Sprite sprite;

sprite.setTexture(texture);

sf::Font font;

if (!font.loadFromFile("C:/Github/komi/komi/arialmt.ttf"))

{

return 2;

}

//создаем массив точек, по которым будут рисоваться линии:

sf::VertexArray lines\_1\_2\_1(sf::Lines, 16 /\*количество точек\*/);

//далее для каждой из точек задаем свою позицию:

lines\_1\_2\_1[0].position = sf::Vector2f(40, 200);

lines\_1\_2\_1[1].position = sf::Vector2f(230, 50);

//и т.д.

//далее для каждой точки указываем цвет(так можно создавать градиенты):

lines\_1\_2\_1[0].color = sf::Color::White;

lines\_1\_2\_1[1].color = sf::Color::White;

sf::VertexArray lines\_1\_2\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_1\_2\_2[0].position = sf::Vector2f(41, 201);

lines\_1\_2\_2[1].position = sf::Vector2f(231, 51);

lines\_1\_2\_2[0].color = sf::Color::White;

lines\_1\_2\_2[1].color = sf::Color::White;

sf::VertexArray lines\_2\_5\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_2\_5\_1[0].position = sf::Vector2f(230, 50);

lines\_2\_5\_1[1].position = sf::Vector2f(430, 50);

lines\_2\_5\_1[0].color = sf::Color::White;

lines\_2\_5\_1[1].color = sf::Color::White;

sf::VertexArray lines\_2\_5\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_2\_5\_2[0].position = sf::Vector2f(231, 51);

lines\_2\_5\_2[1].position = sf::Vector2f(431, 51);

lines\_2\_5\_2[0].color = sf::Color::White;

lines\_2\_5\_2[1].color = sf::Color::White;

sf::VertexArray lines\_5\_7\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_5\_7\_1[0].position = sf::Vector2f(430, 50);

lines\_5\_7\_1[1].position = sf::Vector2f(630, 50);

lines\_5\_7\_1[0].color = sf::Color::White;

lines\_5\_7\_1[1].color = sf::Color::White;

sf::VertexArray lines\_5\_7\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_5\_7\_2[0].position = sf::Vector2f(431, 51);

lines\_5\_7\_2[1].position = sf::Vector2f(631, 51);

lines\_5\_7\_2[0].color = sf::Color::White;

lines\_5\_7\_2[1].color = sf::Color::White;

sf::VertexArray lines\_7\_6\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_7\_6\_1[0].position = sf::Vector2f(630, 50);

lines\_7\_6\_1[1].position = sf::Vector2f(630, 200);

lines\_7\_6\_1[0].color = sf::Color::White;

lines\_7\_6\_1[1].color = sf::Color::White;

sf::VertexArray lines\_7\_6\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_7\_6\_2[0].position = sf::Vector2f(631, 51);

lines\_7\_6\_2[1].position = sf::Vector2f(631, 201);

lines\_7\_6\_2[0].color = sf::Color::White;

lines\_7\_6\_2[1].color = sf::Color::White;

sf::VertexArray lines\_6\_4\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_6\_4\_1[0].position = sf::Vector2f(630, 200);

lines\_6\_4\_1[1].position = sf::Vector2f(480, 330);

lines\_6\_4\_1[0].color = sf::Color::White;

lines\_6\_4\_1[1].color = sf::Color::White;

sf::VertexArray lines\_6\_4\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_6\_4\_1[0].position = sf::Vector2f(631, 201);

lines\_6\_4\_1[1].position = sf::Vector2f(481, 331);

lines\_6\_4\_1[0].color = sf::Color::White;

lines\_6\_4\_1[1].color = sf::Color::White;

sf::VertexArray lines\_4\_3\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_4\_3\_1[0].position = sf::Vector2f(480, 330);

lines\_4\_3\_1[1].position = sf::Vector2f(330, 200);

lines\_4\_3\_1[0].color = sf::Color::White;

lines\_4\_3\_1[1].color = sf::Color::White;

sf::VertexArray lines\_4\_3\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_4\_3\_2[0].position = sf::Vector2f(481, 331);

lines\_4\_3\_2[1].position = sf::Vector2f(331, 201);

lines\_4\_3\_2[0].color = sf::Color::White;

lines\_4\_3\_2[1].color = sf::Color::White;

sf::VertexArray lines\_6\_3\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_6\_3\_1[0].position = sf::Vector2f(630, 200);

lines\_6\_3\_1[1].position = sf::Vector2f(330, 200);

lines\_6\_3\_1[0].color = sf::Color::White;

lines\_6\_3\_1[1].color = sf::Color::White;

sf::VertexArray lines\_6\_3\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_6\_3\_2[0].position = sf::Vector2f(631, 201);

lines\_6\_3\_2[1].position = sf::Vector2f(331, 201);

lines\_6\_3\_2[0].color = sf::Color::White;

lines\_6\_3\_2[1].color = sf::Color::White;

sf::VertexArray lines\_3\_1\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_3\_1\_1[0].position = sf::Vector2f(330, 200);

lines\_3\_1\_1[1].position = sf::Vector2f(40, 200);

lines\_3\_1\_1[0].color = sf::Color::White;

lines\_3\_1\_1[1].color = sf::Color::White;

sf::VertexArray lines\_3\_1\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_3\_1\_2[0].position = sf::Vector2f(331, 201);

lines\_3\_1\_2[1].position = sf::Vector2f(41, 201);

lines\_3\_1\_2[0].color = sf::Color::White;

lines\_3\_1\_2[1].color = sf::Color::White;

sf::VertexArray lines\_1\_4\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_1\_4\_1[0].position = sf::Vector2f(40, 200);

lines\_1\_4\_1[1].position = sf::Vector2f(480, 330);

lines\_1\_4\_1[0].color = sf::Color::White;

lines\_1\_4\_1[1].color = sf::Color::White;

sf::VertexArray lines\_1\_4\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_1\_4\_2[0].position = sf::Vector2f(41, 201);

lines\_1\_4\_2[1].position = sf::Vector2f(481, 331);

lines\_1\_4\_2[0].color = sf::Color::White;

lines\_1\_4\_2[1].color = sf::Color::White;

sf::VertexArray lines\_1\_5\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_1\_5\_1[0].position = sf::Vector2f(40, 200);

lines\_1\_5\_1[1].position = sf::Vector2f(430, 50);

lines\_1\_5\_1[0].color = sf::Color::White;

lines\_1\_5\_1[1].color = sf::Color::White;

sf::VertexArray lines\_1\_5\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_1\_5\_2[0].position = sf::Vector2f(41, 201);

lines\_1\_5\_2[1].position = sf::Vector2f(431, 51);

lines\_1\_5\_2[0].color = sf::Color::White;

lines\_1\_5\_2[1].color = sf::Color::White;

sf::VertexArray lines\_5\_6\_1(sf::Lines, 16 /\*количество точек\*/);

lines\_5\_6\_1[0].position = sf::Vector2f(430, 50);

lines\_5\_6\_1[1].position = sf::Vector2f(630, 200);

lines\_5\_6\_1[0].color = sf::Color::White;

lines\_5\_6\_1[1].color = sf::Color::White;

sf::VertexArray lines\_5\_6\_2(sf::Lines, 16 /\*количество точек\*/);

lines\_5\_6\_2[0].position = sf::Vector2f(431, 51);

lines\_5\_6\_2[1].position = sf::Vector2f(631, 201);

lines\_5\_6\_2[0].color = sf::Color::White;

lines\_5\_6\_2[1].color = sf::Color::White;

// Создаём текст

sf::Text text\_1("1", font,30);

text\_1.setStyle(sf::Text::Bold);

text\_1.setFillColor(sf::Color::White);

text\_1.setPosition(30, 175);

sf::CircleShape circle\_1;

circle\_1.setRadius(30);

circle\_1.setFillColor(sf::Color::Blue);

circle\_1.setOutlineColor(sf::Color::Blue);

circle\_1.setOutlineThickness(5);

circle\_1.setPosition(10, 170);

//пара

sf::Text text\_2("2", font, 30);

text\_2.setStyle(sf::Text::Bold);

text\_2.setFillColor(sf::Color::White);

text\_2.setPosition(225, 25);

sf::CircleShape circle\_2;

circle\_2.setRadius(30);

circle\_2.setFillColor(sf::Color::Blue);

circle\_2.setOutlineColor(sf::Color::Blue);

circle\_2.setOutlineThickness(5);

circle\_2.setPosition(200, 20);

//пара

sf::Text text\_5("5", font, 30);

text\_5.setStyle(sf::Text::Bold);

text\_5.setFillColor(sf::Color::White);

text\_5.setPosition(425, 25);

sf::CircleShape circle\_5;

circle\_5.setRadius(30);

circle\_5.setFillColor(sf::Color::Blue);

circle\_5.setOutlineColor(sf::Color::Blue);

circle\_5.setOutlineThickness(5);

circle\_5.setPosition(400, 20);

//пара

sf::Text text\_7("7", font, 30);

text\_7.setStyle(sf::Text::Bold);

text\_7.setFillColor(sf::Color::White);

text\_7.setPosition(625, 25);

sf::CircleShape circle\_7;

circle\_7.setRadius(30);

circle\_7.setFillColor(sf::Color::Blue);

circle\_7.setOutlineColor(sf::Color::Blue);

circle\_7.setOutlineThickness(5);

circle\_7.setPosition(600, 20);

//пара

sf::Text text\_6("6", font, 30);

text\_6.setStyle(sf::Text::Bold);

text\_6.setFillColor(sf::Color::White);

text\_6.setPosition(625, 175);

sf::CircleShape circle\_6;

circle\_6.setRadius(30);

circle\_6.setFillColor(sf::Color::Blue);

circle\_6.setOutlineColor(sf::Color::Blue);

circle\_6.setOutlineThickness(5);

circle\_6.setPosition(600, 170);

//пара

sf::Text text\_3("3", font, 30);

text\_3.setStyle(sf::Text::Bold);

text\_3.setFillColor(sf::Color::White);

text\_3.setPosition(325, 175);

sf::CircleShape circle\_3;

circle\_3.setRadius(30);

circle\_3.setFillColor(sf::Color::Blue);

circle\_3.setOutlineColor(sf::Color::Blue);

circle\_3.setOutlineThickness(5);

circle\_3.setPosition(300, 170);

//пара

sf::Text text\_4("4", font, 30);

text\_4.setStyle(sf::Text::Bold);

text\_4.setFillColor(sf::Color::White);

text\_4.setPosition(475, 305);

sf::CircleShape circle\_4;

circle\_4.setRadius(30);

circle\_4.setFillColor(sf::Color::Blue);

circle\_4.setOutlineColor(sf::Color::Blue);

circle\_4.setOutlineThickness(5);

circle\_4.setPosition(450, 300);

//пара

sf::Text ves\_1\_2("6", font, 20);

ves\_1\_2.setStyle(sf::Text::Bold);

ves\_1\_2.setFillColor(sf::Color::White);

ves\_1\_2.setPosition(75, 125);

sf::Text ves\_1\_5("10", font, 20);

ves\_1\_5.setStyle(sf::Text::Bold);

ves\_1\_5.setFillColor(sf::Color::White);

ves\_1\_5.setPosition(105, 145);

sf::Text ves\_1\_3("6", font, 20);

ves\_1\_3.setStyle(sf::Text::Bold);

ves\_1\_3.setFillColor(sf::Color::White);

ves\_1\_3.setPosition(105, 175);

sf::Text ves\_1\_4("4", font, 20);

ves\_1\_4.setStyle(sf::Text::Bold);

ves\_1\_4.setFillColor(sf::Color::White);

ves\_1\_4.setPosition(105, 198);

sf::Text ves\_2\_1("4", font, 20);

ves\_2\_1.setStyle(sf::Text::Bold);

ves\_2\_1.setFillColor(sf::Color::White);

ves\_2\_1.setPosition(150, 70);

sf::Text ves\_2\_5("2", font, 20);

ves\_2\_5.setStyle(sf::Text::Bold);

ves\_2\_5.setFillColor(sf::Color::White);

ves\_2\_5.setPosition(280, 25);

sf::Text ves\_5\_2("5", font, 20);

ves\_5\_2.setStyle(sf::Text::Bold);

ves\_5\_2.setFillColor(sf::Color::White);

ves\_5\_2.setPosition(380, 25);

sf::Text ves\_5\_1("7", font, 20);

ves\_5\_1.setStyle(sf::Text::Bold);

ves\_5\_1.setFillColor(sf::Color::White);

ves\_5\_1.setPosition(350, 50);

sf::Text ves\_5\_7("7", font, 20);

ves\_5\_7.setStyle(sf::Text::Bold);

ves\_5\_7.setFillColor(sf::Color::White);

ves\_5\_7.setPosition(480, 25);

sf::Text ves\_5\_6("2", font, 20);

ves\_5\_6.setStyle(sf::Text::Bold);

ves\_5\_6.setFillColor(sf::Color::White);

ves\_5\_6.setPosition(480, 60);

sf::Text ves\_7\_5("4", font, 20);

ves\_7\_5.setStyle(sf::Text::Bold);

ves\_7\_5.setFillColor(sf::Color::White);

ves\_7\_5.setPosition(580, 25);

sf::Text ves\_7\_6("6", font, 20);

ves\_7\_6.setStyle(sf::Text::Bold);

ves\_7\_6.setFillColor(sf::Color::White);

ves\_7\_6.setPosition(640, 85);

sf::Text ves\_6\_3("0", font, 20);

ves\_6\_3.setStyle(sf::Text::Bold);

ves\_6\_3.setFillColor(sf::Color::White);

ves\_6\_3.setPosition(555, 170);

sf::Text ves\_6\_5("3", font, 20);

ves\_6\_5.setStyle(sf::Text::Bold);

ves\_6\_5.setFillColor(sf::Color::White);

ves\_6\_5.setPosition(555, 120);

sf::Text ves\_6\_4("8", font, 20);

ves\_6\_4.setStyle(sf::Text::Bold);

ves\_6\_4.setFillColor(sf::Color::White);

ves\_6\_4.setPosition(565, 225);

sf::Text ves\_4\_6("10", font, 20);

ves\_4\_6.setStyle(sf::Text::Bold);

ves\_4\_6.setFillColor(sf::Color::White);

ves\_4\_6.setPosition(500, 265);

sf::Text ves\_4\_3("1", font, 20);

ves\_4\_3.setStyle(sf::Text::Bold);

ves\_4\_3.setFillColor(sf::Color::White);

ves\_4\_3.setPosition(440, 265);

sf::Text ves\_4\_1("2", font, 20);

ves\_4\_1.setStyle(sf::Text::Bold);

ves\_4\_1.setFillColor(sf::Color::White);

ves\_4\_1.setPosition(400, 280);

sf::Text ves\_3\_1("8", font, 20);

ves\_3\_1.setStyle(sf::Text::Bold);

ves\_3\_1.setFillColor(sf::Color::White);

ves\_3\_1.setPosition(260, 170);

sf::Text ves\_3\_6("1", font, 20);

ves\_3\_6.setStyle(sf::Text::Bold);

ves\_3\_6.setFillColor(sf::Color::White);

ves\_3\_6.setPosition(380, 170);

sf::Text ves\_3\_4("2", font, 20);

ves\_3\_4.setStyle(sf::Text::Bold);

ves\_3\_4.setFillColor(sf::Color::White);

ves\_3\_4.setPosition(380, 215);

sf::Text key("Press Enter to run OR Press Escape to exit", font, 20);

key.setStyle(sf::Text::Bold);

key.setFillColor(sf::Color::White);

key.setPosition(200, 400);

sf::Text marsh(" Route ", font, 20);

marsh.setStyle(sf::Text::Bold);

marsh.setFillColor(sf::Color::White);

marsh.setPosition(200, 500);

sf::Text len(" WAY ", font, 20);

len.setStyle(sf::Text::Bold);

len.setFillColor(sf::Color::White);

len.setPosition(400, 550);

sf::Event event;

while (window.isOpen())

{

while (window.pollEvent(event))

{

if (event.type == sf::Event::Closed) window.close();

if (event.type == sf::Event::KeyPressed)

{

// Получаем нажатую клавишу - выполняем соответствующее действие

if (event.key.code == sf::Keyboard::Escape) window.close();

}

if (event.type == sf::Event::KeyPressed) {

switch (event.key.code) {

case sf::Keyboard::Enter: // обработка нажатия Escape

int arr[9][8];

int arr1[9][8];

arr[9][8] = made\_arr(arr);

arr1[9][8] = made\_arr(arr1);

show\_arr(arr);

cout << reduction\_arr(arr, arr1);

marsh.setString("6->7->5->2->1->4->3->6");

len.setString(st);

break;

}

}

}

// Выполняем необходимые действия по отрисовке

window.clear();

window.draw(sprite);

window.draw(lines\_1\_2\_1);

window.draw(lines\_1\_2\_2);

window.draw(lines\_2\_5\_1);

window.draw(lines\_2\_5\_2);

window.draw(lines\_5\_7\_1);

window.draw(lines\_5\_7\_2);

window.draw(lines\_7\_6\_1);

window.draw(lines\_7\_6\_2);

window.draw(lines\_6\_4\_1);

window.draw(lines\_6\_4\_2);

window.draw(lines\_4\_3\_1);

window.draw(lines\_4\_3\_2);

window.draw(lines\_6\_3\_1);

window.draw(lines\_6\_3\_2);

window.draw(lines\_3\_1\_1);

window.draw(lines\_3\_1\_2);

window.draw(lines\_1\_4\_1);

window.draw(lines\_1\_4\_2);

window.draw(lines\_1\_5\_1);

window.draw(lines\_1\_5\_2);

window.draw(lines\_5\_6\_1);

window.draw(lines\_5\_6\_2);

window.draw(ves\_1\_2);

window.draw(ves\_1\_5);

window.draw(ves\_1\_3);

window.draw(ves\_1\_4);

window.draw(ves\_2\_1);

window.draw(ves\_2\_5);

window.draw(ves\_5\_2);

window.draw(ves\_5\_1);

window.draw(ves\_5\_7);

window.draw(ves\_5\_6);

window.draw(ves\_7\_5);

window.draw(ves\_7\_6);

window.draw(ves\_6\_3);

window.draw(ves\_6\_5);

window.draw(ves\_6\_4);

window.draw(ves\_4\_6);

window.draw(ves\_4\_3);

window.draw(ves\_4\_1);

window.draw(ves\_3\_1);

window.draw(ves\_3\_6);

window.draw(ves\_3\_4);

window.draw(circle\_1);

window.draw(text\_1);

window.draw(circle\_2);

window.draw(text\_2);

window.draw(circle\_3);

window.draw(text\_3);

window.draw(circle\_4);

window.draw(text\_4);

window.draw(circle\_5);

window.draw(text\_5);

window.draw(circle\_6);

window.draw(text\_6);

window.draw(circle\_7);

window.draw(text\_7);

window.draw(key);

window.draw(marsh);

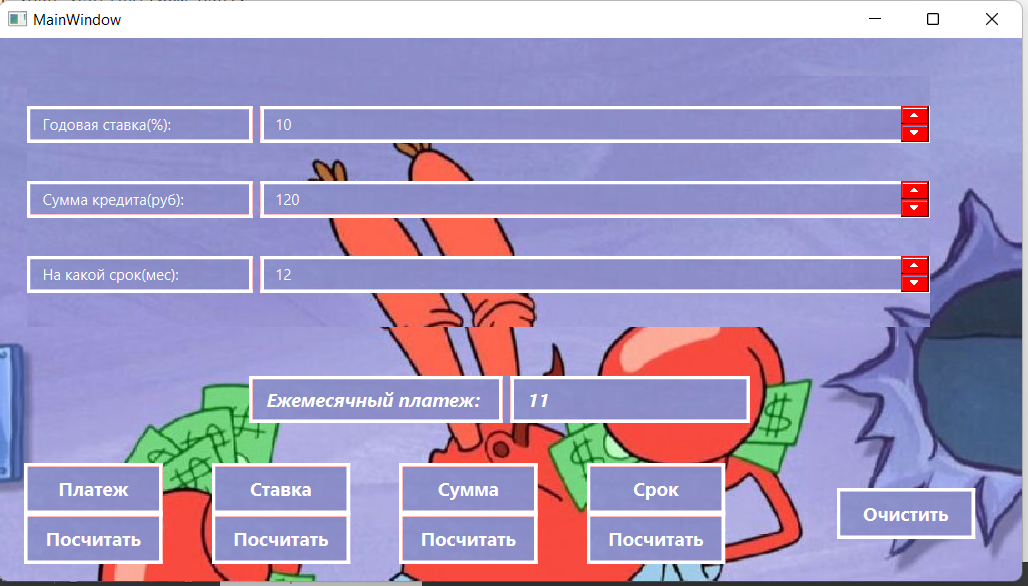
window.draw(len);

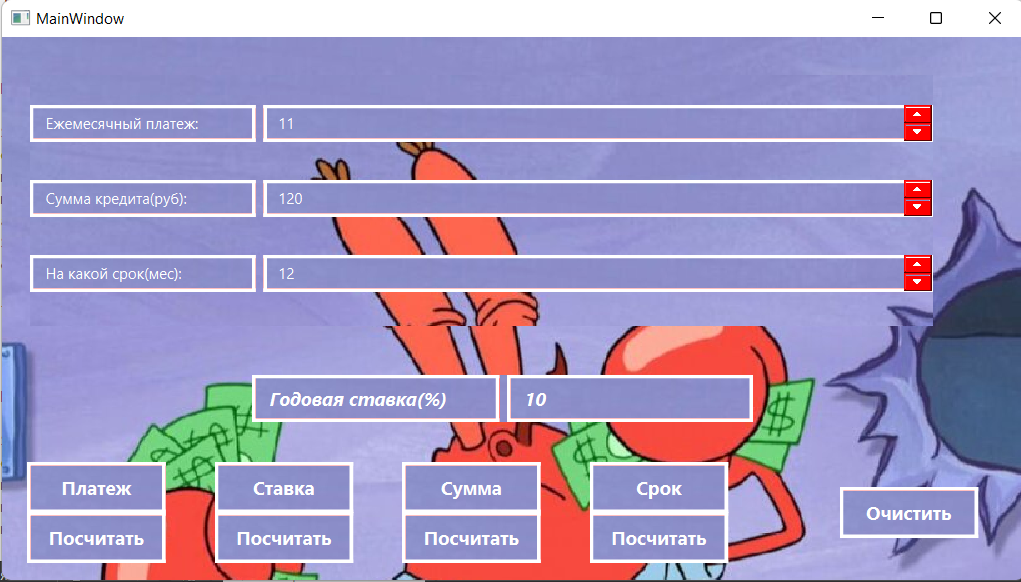
window.display();

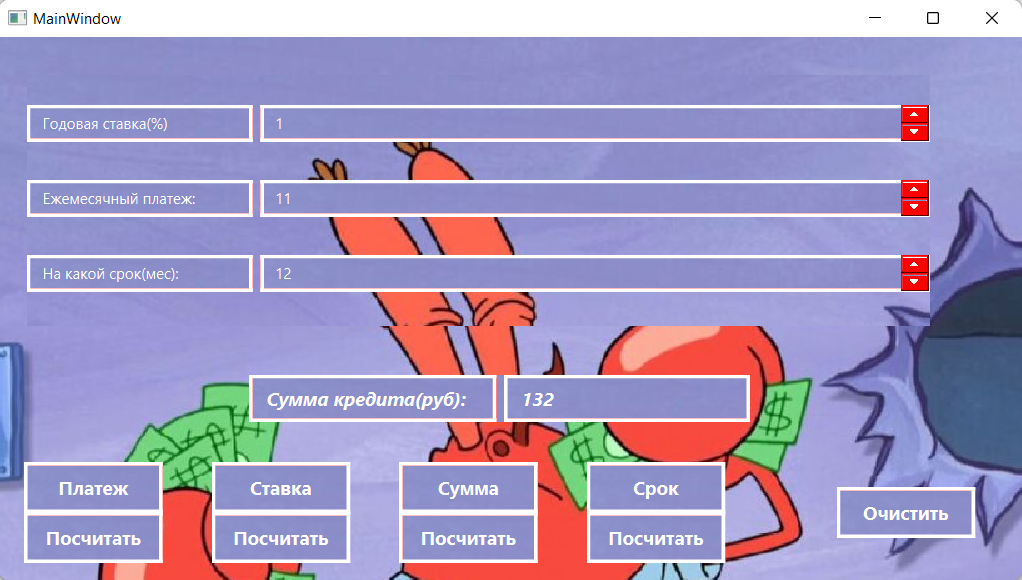
}

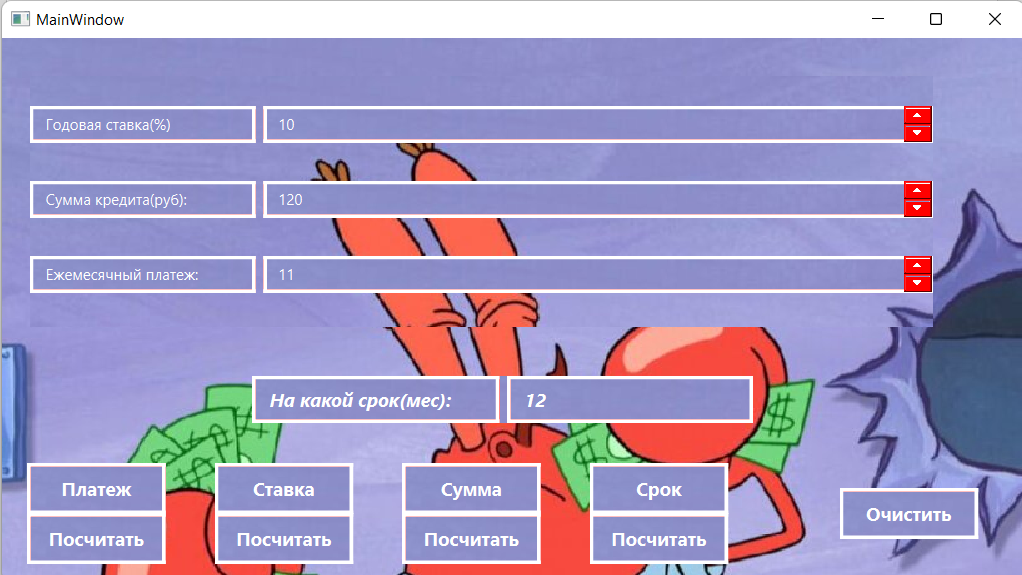
}

Скриншоты результатов калькулятор:





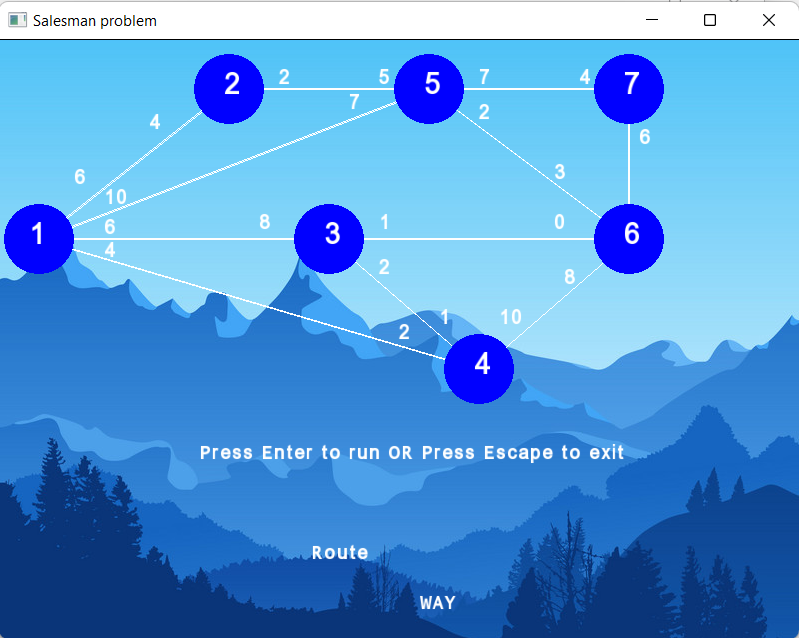


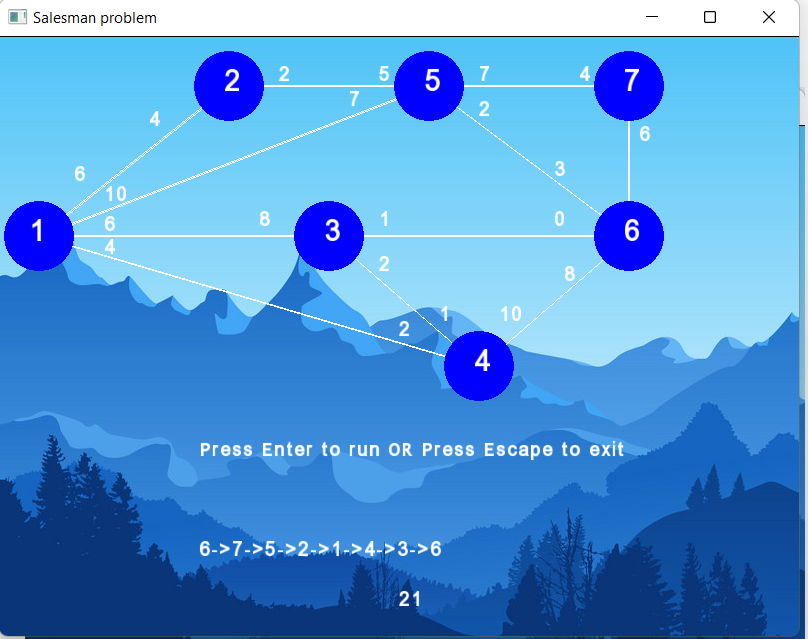


Анализ результатов:

Программа работает корректно. Все значения считаются правильно.

Скриншоты результатов коммивояжер:





Анализ результатов:

Программа работает корректно. Выводится кратчайший маршрут и его длина.