МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Основы алгоритмизации и программирования»

Лабораторная работа №5

Вложенные циклы

Выполнил:

студент гр. ПИбд-12

Дозорова Алена

Ульяновск, 2021 г

Задание лабораторной работы

1. Нарисовать корону
   1. Вывести короны в горизонтальный ряд (с изменением ширины и с изменением цвета). Какой цвет менять – на ваш выбор (RED, GREEN, BLUE) .
   2. Вывести короны в вертикальный ряд (с изменением высоты и с изменением цвета – отличного от изменения по горизонтали). Какой цвет менять – на ваш выбор (RED, GREEN, BLUE) .
   3. Вывести короны в диагональный ряд (с одновременным изменением ширины, высоты и изменением двух цветов).
   4. Вывести короны в виде блока (несколько рядов по несколько корон).
2. Создать функцию MyFigure() для отрисовки своей собственной фигуры – с возможностью изменения ширины, высоты и цвета.
   1. Вывести MyFigure() *(созданный в задаче 2.0)* в горизонтальный ряд (с изменением ширины и с изменением цвета). Какой цвет менять – на ваш выбор (RED, GREEN, BLUE) .
   2. Вывести MyFigure() *(созданный в задаче 2.0)* в вертикальный ряд (с изменением высоты и с изменением цвета – отличного от изменения по горизонтали). Какой цвет менять – на ваш выбор (RED, GREEN, BLUE)
   3. Вывести MyFigure() *(созданный в задаче 2.0)* в диагональный ряд (с одновременным изменением ширины, высоты и изменением двух цветов).
   4. Вывести MyFigure() *(созданный в задаче 2.0)* в виде блока (несколько рядов по несколько фигур).
3. Создать ЕЩЕ одну (вторую) собственную фигуру. И сделать при помощи неё – горизонтальный ряд, вертикальный ряд, диагональный ряд, много рядов.
4. Создать ЕЩЕ одну (третью) собственную фигуру. И сделать при помощи неё – горизонтальный ряд, вертикальный ряд, диагональный ряд, много рядов.
5. Нарисовать при помощи вашей фигуры узор в виде пирамиды:

Код программы

int number = 0;

int num2 = 0;

void Сrown(HDC hdc, int cx, int cy, int sizeX, int sizeY, COLORREF color) {

POINT p[] = {

cx, cy - sizeY,

cx + sizeX / 2, cy,

cx + sizeX, cy - sizeY,

cx + sizeX, cy + sizeY,

cx - sizeX, cy + sizeY,

cx - sizeX, cy - sizeY,

cx - sizeX / 2, cy,

cx, cy - sizeY

};

HPEN hPen;

hPen = CreatePen(PS\_SOLID, 3, color);

SelectObject(hdc, hPen);

Polyline(hdc, p, 8);

DeleteObject(hPen);

}

void BlockСrown1(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (x < 900) {

Сrown(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

g += 30;

}

}

void BlockСrown2(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

Сrown(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

r += 30;

}

}

void BlockСrown3(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

Сrown(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

}

void BlockСrown4(HDC hdc) {

int y = 50;

int sizeY = 5;

while (y < 600)

{

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 5;

while (x < 600) {

Сrown(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

y += 80;

sizeY += 5;

}

}

void Arrow(HDC hdc, int cx, int cy, int sizeX, int sizeY, COLORREF color) {

POINT p[] = {

cx-sizeX, cy - sizeY,

cx, cy - sizeY,

cx + sizeX, cy,

cx, cy+sizeY,

cx - sizeX, cy + sizeY,

cx - sizeX / 2, cy,

cx - sizeX, cy - sizeY,

};

HPEN hPen;

hPen = CreatePen(PS\_SOLID, 3, color);

SelectObject(hdc, hPen);

Polyline(hdc, p, 7);

DeleteObject(hPen);

}

void BlockArrow1(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (x < 900) {

Arrow(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

g += 30;

}

}

void BlockArrow2(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

Arrow(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

r += 30;

}

}

void BlockArrow3(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

Arrow(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

}

void BlockArrow4(HDC hdc) {

int y = 50;

int sizeY = 5;

while (y < 600)

{

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 5;

while (x < 600) {

Arrow(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

y += 80;

sizeY += 5;

}

}

void MyFigure1(HDC hdc, int cx, int cy, int sizeX, int sizeY, COLORREF color) {

POINT p[] = {

cx, cy - sizeY,

cx + sizeX/3, cy - sizeY/3,

cx + sizeX, cy,

cx, cy + sizeY,

cx - sizeX/3, cy + sizeY/3,

cx - sizeX, cy,

cx, cy - sizeY,

};

HPEN hPen;

hPen = CreatePen(PS\_SOLID, 3, color);

SelectObject(hdc, hPen);

Polyline(hdc, p, 7);

DeleteObject(hPen);

}

void BlockMyFigure11(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (x < 900) {

MyFigure1(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

g += 30;

}

}

void BlockMyFigure12(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

MyFigure1(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

r += 30;

}

}

void BlockMyFigure13(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

MyFigure1(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

}

void BlockMyFigure14(HDC hdc) {

int y = 50;

int sizeY = 5;

while (y < 600)

{

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 5;

while (x < 600) {

MyFigure1(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

y += 80;

sizeY += 5;

}

}

void MyFigure2(HDC hdc, int cx, int cy, int sizeX, int sizeY, COLORREF color) {

POINT p[] = {

cx, cy - sizeY,

cx + sizeX / 4, cy - sizeY / 4,

cx + sizeX, cy+sizeY/2,

cx, cy + sizeY,

cx - sizeX / 6, cy + sizeY / 4,

cx - sizeX / 6, cy - sizeY / 4,

cx, cy - sizeY,

};

HPEN hPen;

hPen = CreatePen(PS\_SOLID, 3, color);

SelectObject(hdc, hPen);

Polyline(hdc, p, 7);

DeleteObject(hPen);

}

void BlockMyFigure21(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (x < 900) {

MyFigure2(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

g += 30;

}

}

void BlockMyFigure22(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

MyFigure2(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

r += 30;

}

}

void BlockMyFigure23(HDC hdc) {

int y = 50;

int sizeY = 10;

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 10;

while (y < 600) {

MyFigure2(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

y += 80;

sizeY += 5;

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

}

void BlockMyFigure24(HDC hdc) {

int y = 50;

int sizeY = 5;

while (y < 600)

{

int r = 0;

int b = 0;

int g = 0;

int x = 50;

int sizeX = 5;

while (x < 600) {

MyFigure2(hdc, x, y, sizeX, sizeY, RGB(r, g, b));

x += 100;

sizeX += 5;

b += 30;

g += 30;

}

y += 80;

sizeY += 5;

}

}

void task1(int n, HDC hdc)

{

if (n == 1)

{

//task1

BlockСrown1(hdc);

//task 2

BlockСrown2(hdc);

//task 3

BlockСrown3(hdc);

}

else {

BlockСrown4(hdc);

}

}

void task2(int n, HDC hdc)

{

if (n == 1)

{

//task1

BlockArrow1(hdc);

//task 2

BlockArrow2(hdc);

//task 3

BlockArrow3(hdc);

}

else {

BlockArrow4(hdc);

}

}

void task3(int n, HDC hdc)

{

if (n == 1)

{

//task1

BlockMyFigure11(hdc);

//task 2

BlockMyFigure12(hdc);

//task 3

BlockMyFigure13(hdc);

}

else {

BlockMyFigure14(hdc);

}

}

void task4(int n, HDC hdc)

{

if (n == 1)

{

//task1

BlockMyFigure21(hdc);

//task 2

BlockMyFigure22(hdc);

//task 3

BlockMyFigure23(hdc);

}

else {

BlockMyFigure24(hdc);

}

}

void rec(int x, int y, int sizex, int sizey, HDC hdc, COLORREF color)

{

MyFigure1(hdc, x + sizex, y, sizex, sizey, color);

MyFigure1(hdc, x - sizex, y, sizex, sizey, color);

}

void task5(HDC hdc)

{

int x = 500;

int y = 50;

int r = 0; int g = 0, b = 0;

int sizex = 5;

int sizey = 5;

for (int i = 0; i < 13; i++)

{

int j = 0;

while (j <= i)

{

MyFigure1(hdc, x-j\*20, y, sizex, sizey, RGB(r, g, b));

MyFigure1(hdc, x + j \* 20, y, sizex, sizey, RGB(r, g, b));

j += 1;

}

y += 50;

r += 30;

sizey+= 5;

}

}

case WM\_KEYDOWN:

switch (wParam)

{

case VK\_F1:

number = 1;

num2 = 1;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F2:

number = 1;

num2 = 2;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F3:

number = 2;

num2 = 1;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F4:

number = 2;

num2 = 2;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F5:

number = 3;

num2 = 1;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F6:

number = 3;

num2 = 2;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F7:

number = 4;

num2 = 1;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F8:

number = 4;

num2 = 2;

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F9:

number = 5;

InvalidateRect(hWnd, NULL, TRUE);

break;

}

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

if (number == 1)

{

task1(num2, hdc);

}

if (number == 2)

{

task2(num2, hdc);

}

if (number == 3)

{

task3(num2, hdc);

}

if (number == 4)

{

task4(num2, hdc);

}

if (number == 5)

{

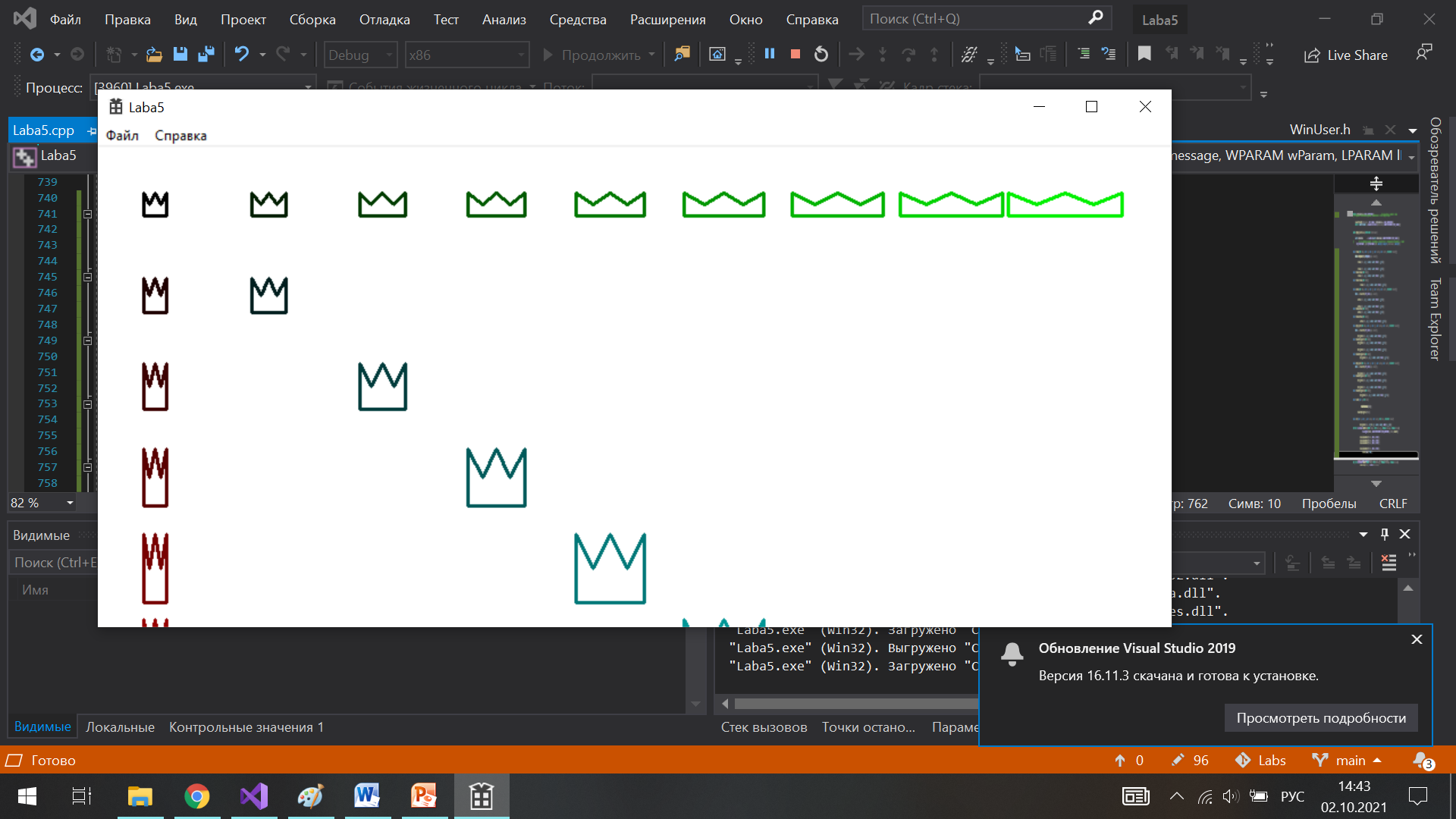
task5(hdc);

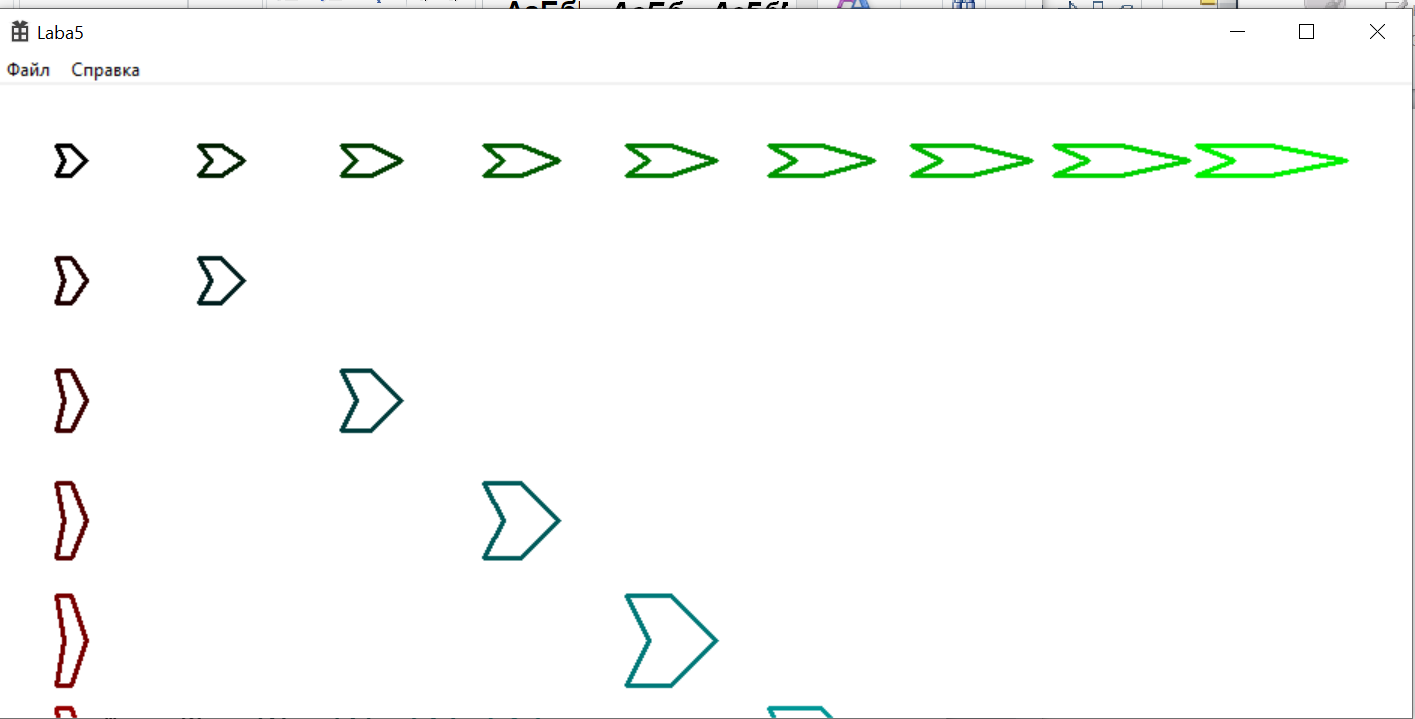
}

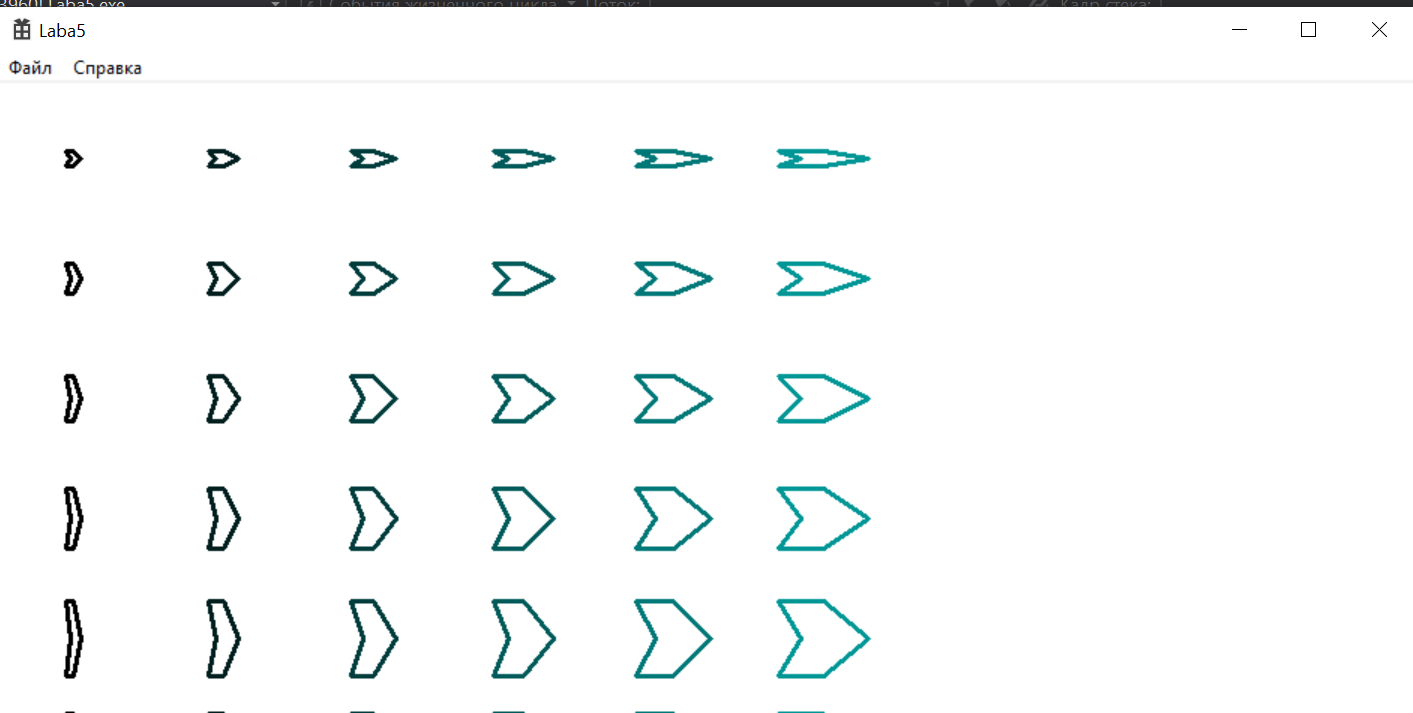
EndPaint(hWnd, &ps);

}

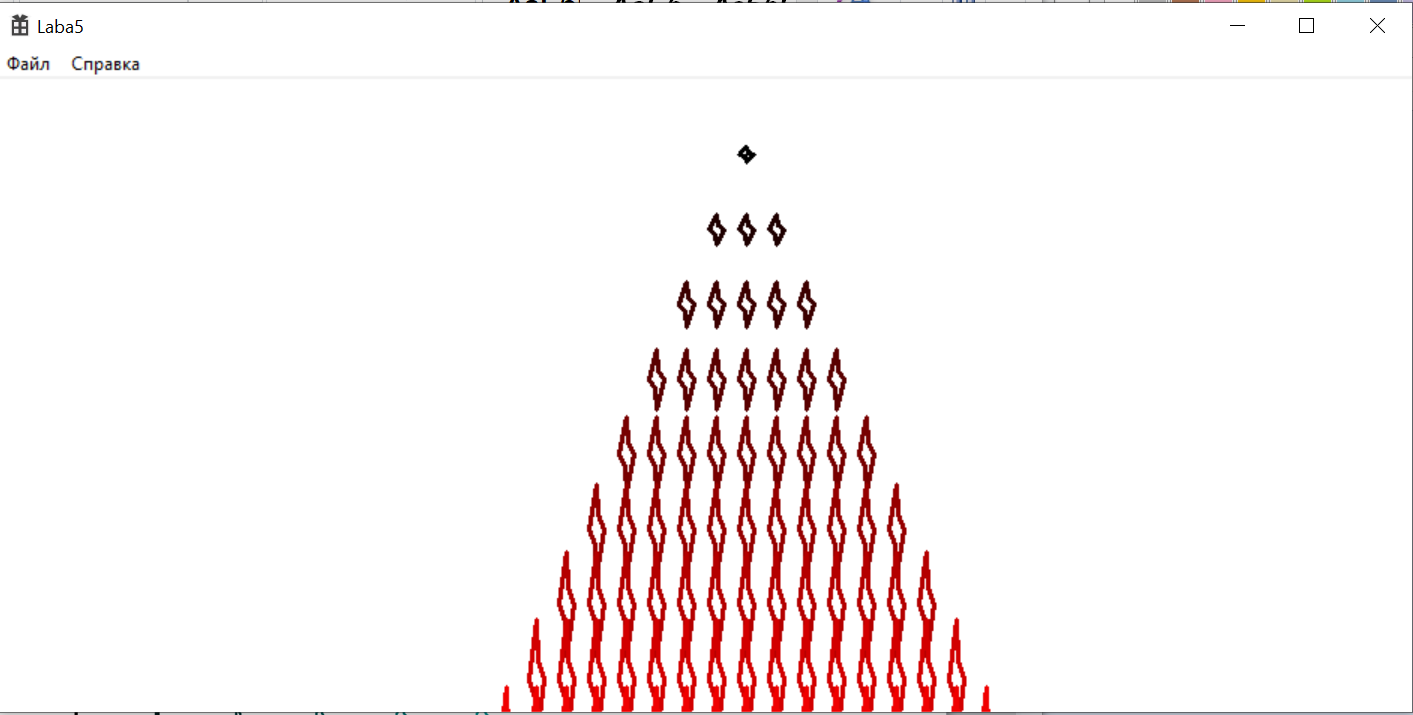
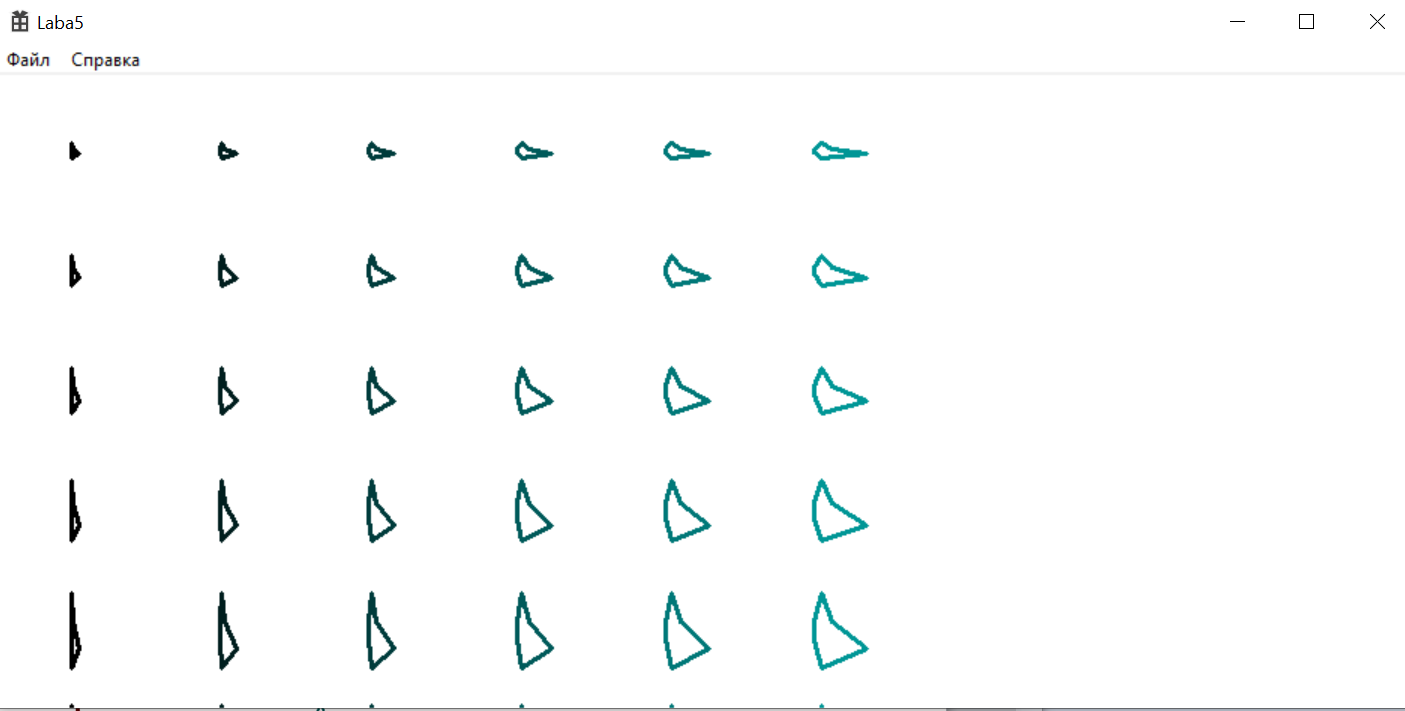
Скриншоты работающей программы



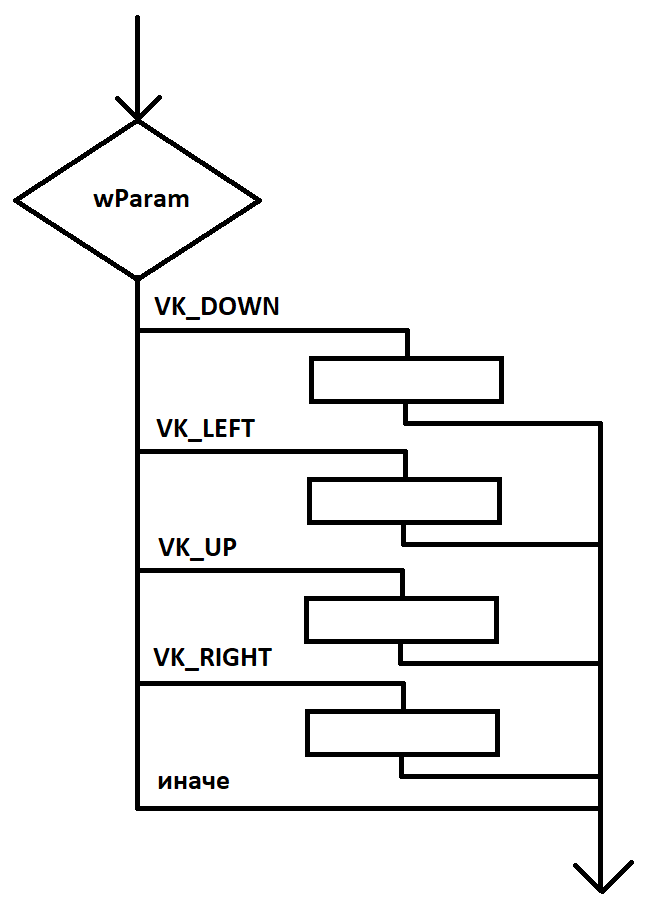
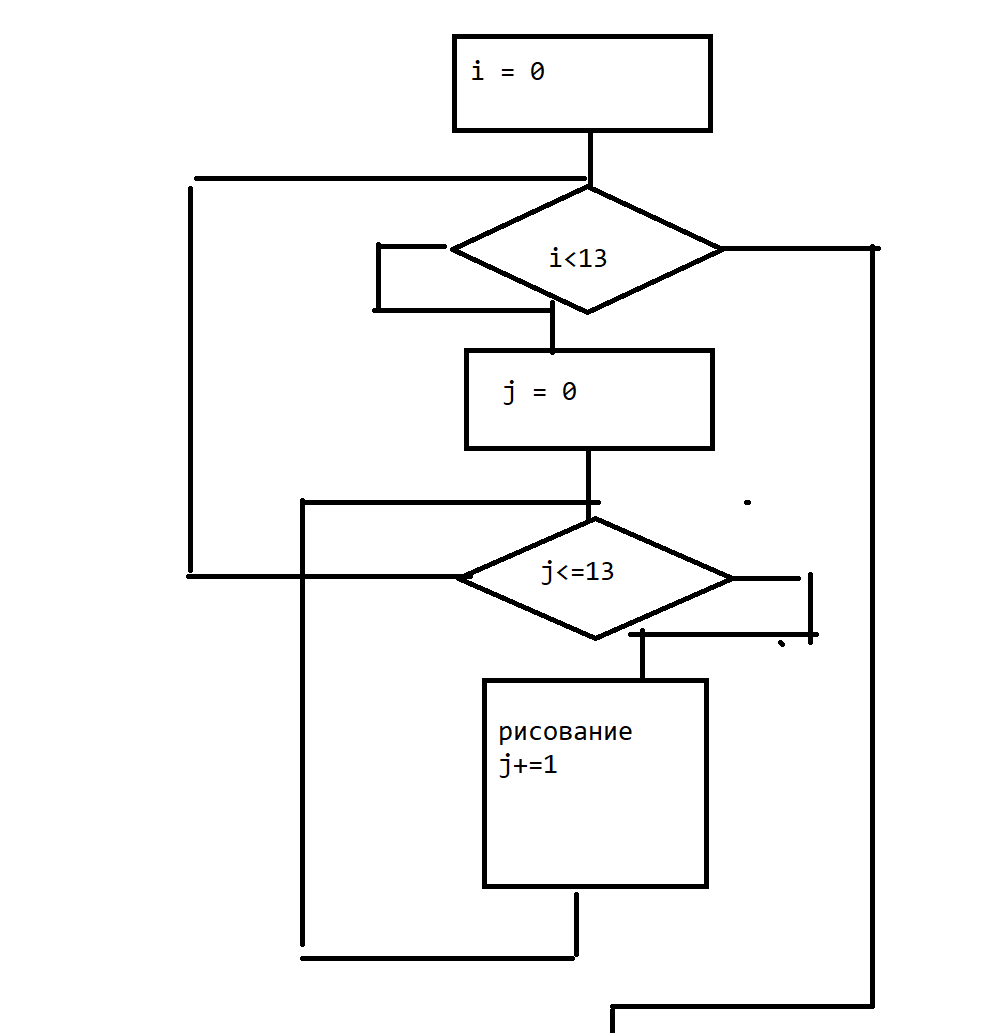


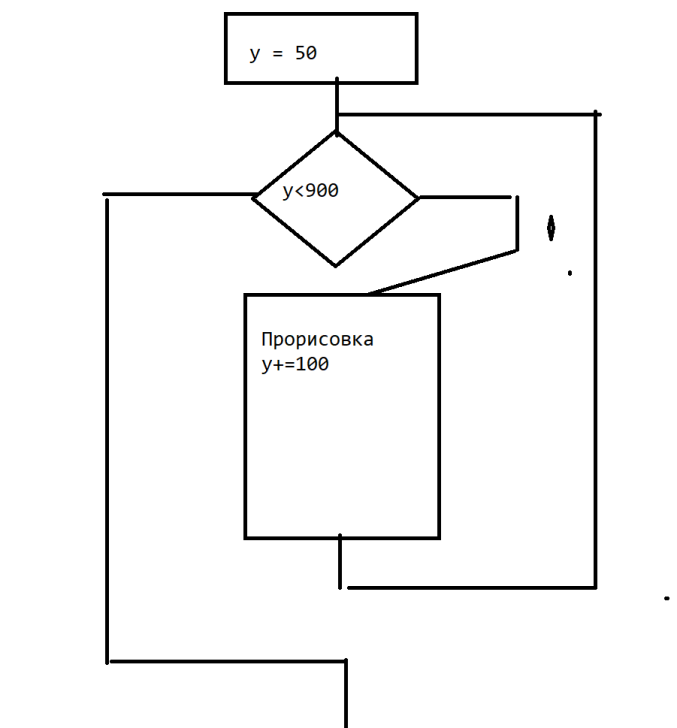






Блок схемы вложенных циклов



Заключение

В ходе этой лабораторной я научилась работать со вложенными циклами, меняя несколько параметров объекта, который прорисовывается на экране.