МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра информационных систем

Отчет по дисциплине Основы алгоритмизации и программирование

Домашняя работа №1

Выполнил студент гр. ИСЭбд-11

Гапон Николай Иванович

Преподаватель

Власенко Олег Федосович

ССЫЛКА НА GIT https://github.com/NikGapon/1year/tree/main/lab5

Домашнее задание

• Доделать задачи 1-2, которые не успели сделать в классе.

• Задача 3. Создать ЕЩЕ одну (вторую) собственную фигуру. И сделать при помощи неё – горизонтальный ряд, вертикальный ряд, диагональный ряд, много рядов.

• Задача 4\*. Создать ЕЩЕ одну (третью) собственную фигуру. И сделать при помощи неё – горизонтальный ряд, вертикальный ряд, диагональный ряд, много рядов.

• Задача 5\*\*. Нарисовать при помощи вашей фигуры узор в виде пирамиды:

Всё всё так же сделал в функциях

void Cor(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 cor\_gor(HDC hdc){

{

int r = 0;

int x = 50;

int sizex = 10;

while (x < 2100) {

Cor(hdc, x, 50, sizex,20, RGB(r, 0, 0));

x = x + 100;

sizex = sizex + 3;

r += 20;

}

}

}

void cor\_ver(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

while (y < 2100) {

Cor(hdc, x, y, sizex, sizey, RGB(r, 0, 0));

y = y + 100;

sizey = sizey + 3;

r += 20;

}

}

}

void cor\_riad(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

int b = 0;

while (y < 2100) {

while (x < 2100) {

x = x + 100;

sizex = sizex + 1;

b += 1;

Cor(hdc, x, y, sizex, sizey, RGB(r, 0, b));

}

x = 100;

y = y + 100;

sizey = sizey + 1;

r += 20;

}

}

}

void cor\_diag(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

while (y < 2100) {

Cor(hdc, x, y, sizex, sizey, RGB(r, 0, 0));

y = y + 100;

sizey = sizey + 3;

x = x + 100;

sizex = sizex + 3;

r += 20;

}

}

}

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

POINT l[] = {

cx, cy,

cx + sizeX, cy + sizeY,

cx, cy + sizeY \* 2,

cx - sizeX, cy + sizeY,

cx, cy

};

HPEN hPen;

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

SelectObject(hdc, hPen);

Polyline(hdc, l, 5);

DeleteObject(hPen);

}

void f\_gor(HDC hdc) {

{

int r = 0;

int x = 50;

int sizex = 10;

while (x < 2100) {

MyFigure(hdc, x, 50, sizex, 20, RGB(r, 0, 0));

x = x + 100;

sizex = sizex + 3;

r += 20;

}

}

}

void f\_riad(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

int b = 0;

while (y < 2100) {

while (x < 2100) {

x = x + 100;

sizex = sizex + 1;

b += 1;

MyFigure(hdc, x, y, sizex, sizey, RGB(r, 0, b));

}

x = 100;

y = y + 100;

sizey = sizey + 1;

r += 20;

}

}

}

void f\_diag(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

while (y < 2100) {

MyFigure(hdc, x, y, sizex, sizey, RGB(r, 0, 0));

y = y + 100;

sizey = sizey + 3;

x = x + 100;

sizex = sizex + 3;

r += 20;

}

}

}

void f\_ver(HDC hdc) {

{

int r = 0;

int x = 100;

int sizex = 10;

int y = 50;

int sizey = 10;

while (y < 2100) {

MyFigure(hdc, x, y, sizex, sizey, RGB(r, 0, 0));

y = y + 100;

sizey = sizey + 3;

r += 20;

}

}

}

void f\_p(HDC hdc) {

{

int r = 0;

int x = 50;

int sizex = 10;

int y = 50;

int sizey = 0;

while (y < 2100) {

MyFigure(hdc, x, y, sizex, sizey, RGB(r, 0, 0));

y = y + 100;

x = x + 100;

sizex = sizex +1;

sizey = sizey + 1;

r += 20;

}

}

}

void Claus(HDC hdc, int x, int y) {

MoveToEx(hdc, 20 + x, 0 + y, NULL);

LineTo(hdc, 30 + x, 20 + y);

LineTo(hdc, 10 + x, 20 + y);

LineTo(hdc, 20 + x, 0 + y);

MoveToEx(hdc, 20 + x, 10 + y, NULL);

LineTo(hdc, 30 + x, 40 + y);

LineTo(hdc, 10 + x, 40 + y);

LineTo(hdc, 20 + x, 10 + y);

MoveToEx(hdc, 20 + x, 30 + y, NULL);

LineTo(hdc, 40 + x, 80 + y);

LineTo(hdc, 0 + x, 80 + y);

LineTo(hdc, 20 + x, 30 + y);

}

void el(HDC hdc) {

{

int y = 0;

int x = 0;

while (y < 1100) {

Claus(hdc, x, y);

x = x + 50;

y = y + 80;

}

}

}

void was(HDC hdc) {

{

int y = 0;

int x = 500;

int p = 0;

int k = 1;

int l = 0;

while (y < 1100) {

p = (x - 50 \* (k/3)) ;

while (l < k) {

Claus(hdc, p, y);

p = p + 50;

l = l + 1;

}

l = 0;

k = k \* 3;

y = y + 80;

x = 500;

}

}

}

void j(HDC hdc) {

if (KArtin == 0) {

f\_gor(hdc);

}

if (KArtin == 2) {

f\_ver(hdc);

}

if (KArtin == 3) {

f\_diag(hdc);

}

if (KArtin == 4) {

f\_riad(hdc);

}

if (KArtin == 5) {

f\_p(hdc);

}

if (KArtin == 6) {

el(hdc);

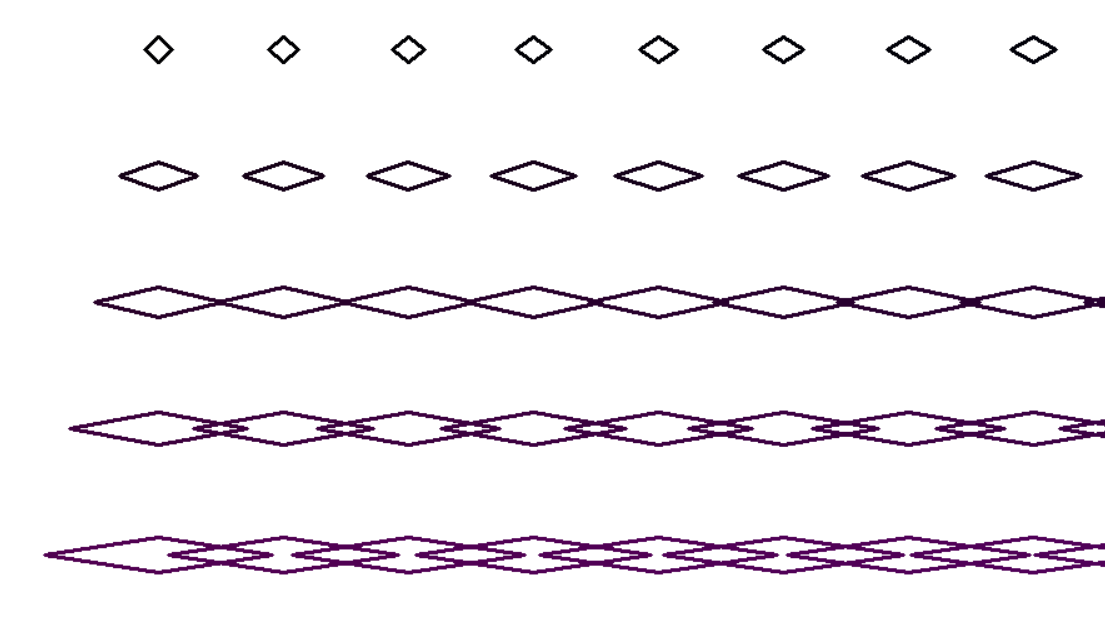
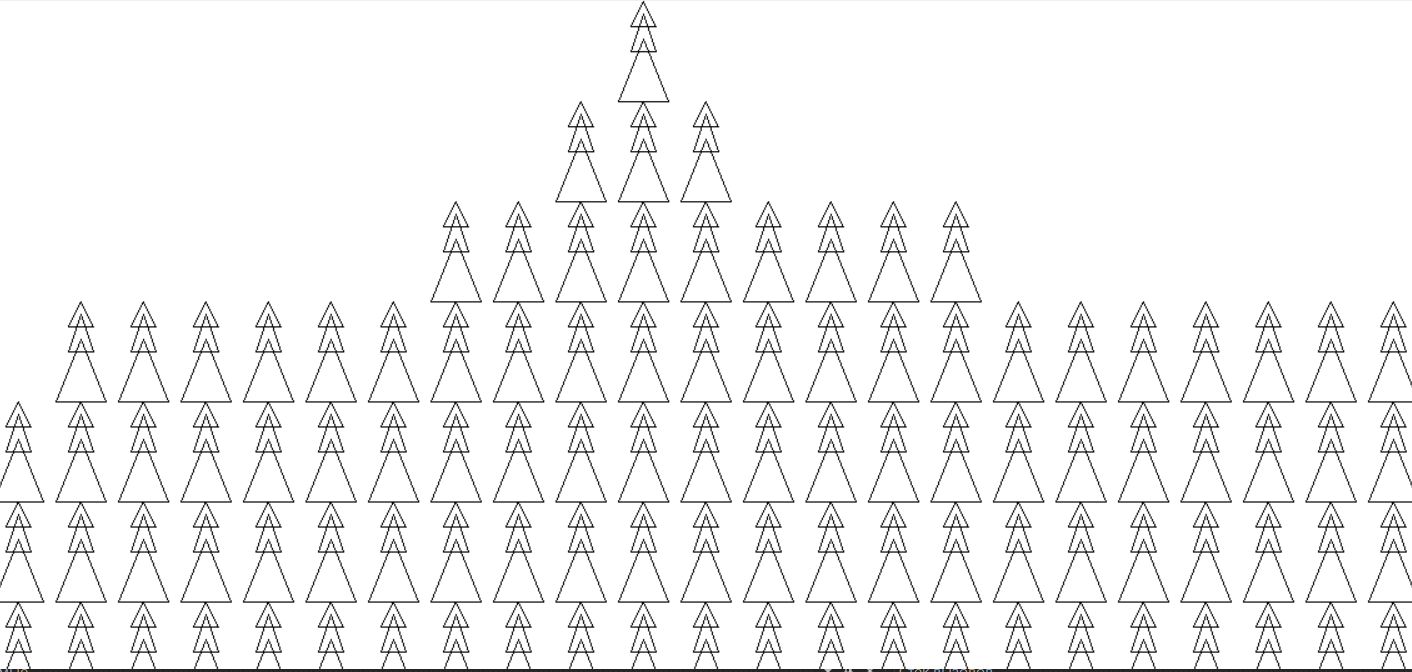
}

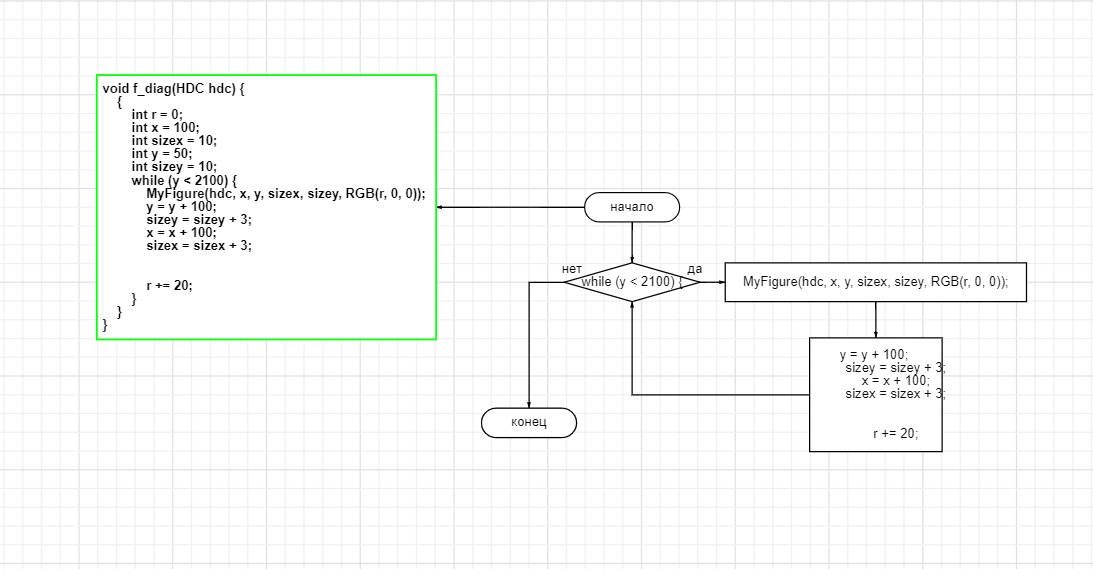
if (KArtin == 7) {

was(hdc);

}

}





ВЫВОД

В данной работе я продолжил изучать рекурсию

А так же выучил работу с цветами