



МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технологический университет

«СТАНКИН»

(ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт
информационных
технологий**

**Кафедра
информационных систем**

Основная образовательная программа 09.03.02

«Информационные системы и технологии»

Отчет по дисциплине «Компьютерная геометрия и графика»

по лабораторной работе № 1

Студент
группы ИДБ-21-06

Музафаров.К.Р

Преподаватель

к.т.н. , доц. Разумовский А.И

Москва, 2022 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ЗАДАНИЕ 1.....	4
ЗАДАНИЕ 2.....	6
Вывод.....	9

ВВЕДЕНИЕ

В этой лабораторной работе будут рассмотрены следующие необходимые аспекты:

1. Создание окна
2. Базовые элементы
3. Запуск приложения

Создание окна. Реализация базовых графических объектов.

Взаимодействие между приложением и системой происходит с помощью набора сообщений, например, можно создать отклик на нажатие клавиши мыши в окне. Если сообщения не описаны программистом, то реакция системы на их возникновение будет стандартной.

Библиотека функций, при помощи которых происходит взаимодействие приложения с ОС, это функции Программного Интерфейса Приложений (Application Program Interface, API). Подмножество этих функций, предназначенных для графического вывода на экран, представляет собой Интерфейс Графических Устройств (Graphics Device Interface, GDI).

Приложение реализуется в среде Visual C++.

ЗАДАНИЕ 1

Ввести и отладить в среде Visual Studio код каркаса Windows-приложения.

Код программы:

```
#include <windows.h>
#include <tchar.h>
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd, int mode)
{
    HWND hWnd; //Дескриптор главного окна программы
    MSG msg; //Структура для хранения сообщений
    WNDCLASS wc; // Класс окна
    // Определение класса окна
    wc.hInstance = This;
    wc.lpszClassName = WinName; // Имя класса окна
    wc.lpfnWndProc = WndProc; // Функция окна
    wc.style = CS_HREDRAW | CS_VREDRAW; //Стиль окна
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION); //Стандартная иконка
    wc.hCursor = LoadCursor(NULL, IDC_ARROW); //Стандартный курсор
    wc.lpszMenuName = NULL; // Нет меню
    wc.cbClsExtra = 0; // Нет дополнительных данных класса
    wc.cbWndExtra = 0; //Нет дополнительных данных окна

    // Заполнение окна белым цветом
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1); //установка цвета фона
    if (!RegisterClass(&wc)) return 0; // Регистрация класса окна

    // Создание окна
    hWnd = CreateWindow(WinName, _T("Karkas"), // Заголовок окна
        WS_OVERLAPPEDWINDOW, // Стиль окна
        CW_USEDEFAULT, // X
        CW_USEDEFAULT, // Y
        CW_USEDEFAULT, // Width
        CW_USEDEFAULT, // Height
        HWND_DESKTOP, // Дескриптор родительского окна
        NULL, // Нет меню
        This, // Дескриптор приложения
        NULL); // Дополнительной информации нет
    ShowWindow(hWnd, mode); // Показать окно

    // Цикл обработки сообщений
    while (GetMessage(&msg, NULL, 0, 0)) //цикл получения сообщений
    {
        TranslateMessage(&msg); // Функция трансляции кодов нажатой клавиши
        DispatchMessage(&msg); // Посылает сообщение функции WndProc0
    } return 0;
}

// 1 при положительном завершении программы в главную функцию
//возвращается «0»
//Оконная функция вызываемая операционной системой
//и получает сооошения из очереди оля оанного приложения
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{ //Обработчик сообщений
    PAINTSTRUCT ps;
```

```

HDC hdc;
int x, y;
static int sx, sy;
switch (message)
{
case WM_SIZE:
    sx = LOWORD(lParam); //Ширина окна
    sy = HIWORD(lParam); //Высота окна
    break;
case WM_PAINT:
    hdc = BeginPaint(hWnd, &ps);
    for(x=0; x<sx; x+=sx/10)
    {
        MoveToEx(hdc, x, 0, NULL);
        LineTo(hdc, x, sy);
    }
    for (y = 0; y < sy; y += sy / 10)
    {
        MoveToEx(hdc, 0, y, NULL);
        LineTo(hdc, sx, y);
    }
    EndPaint(hWnd, &ps);
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break; // Завершение программы
    // Обработка сообщения по умолчанию
default:
    // возвращение стандартных параметров по умолчанию
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат работы программы:

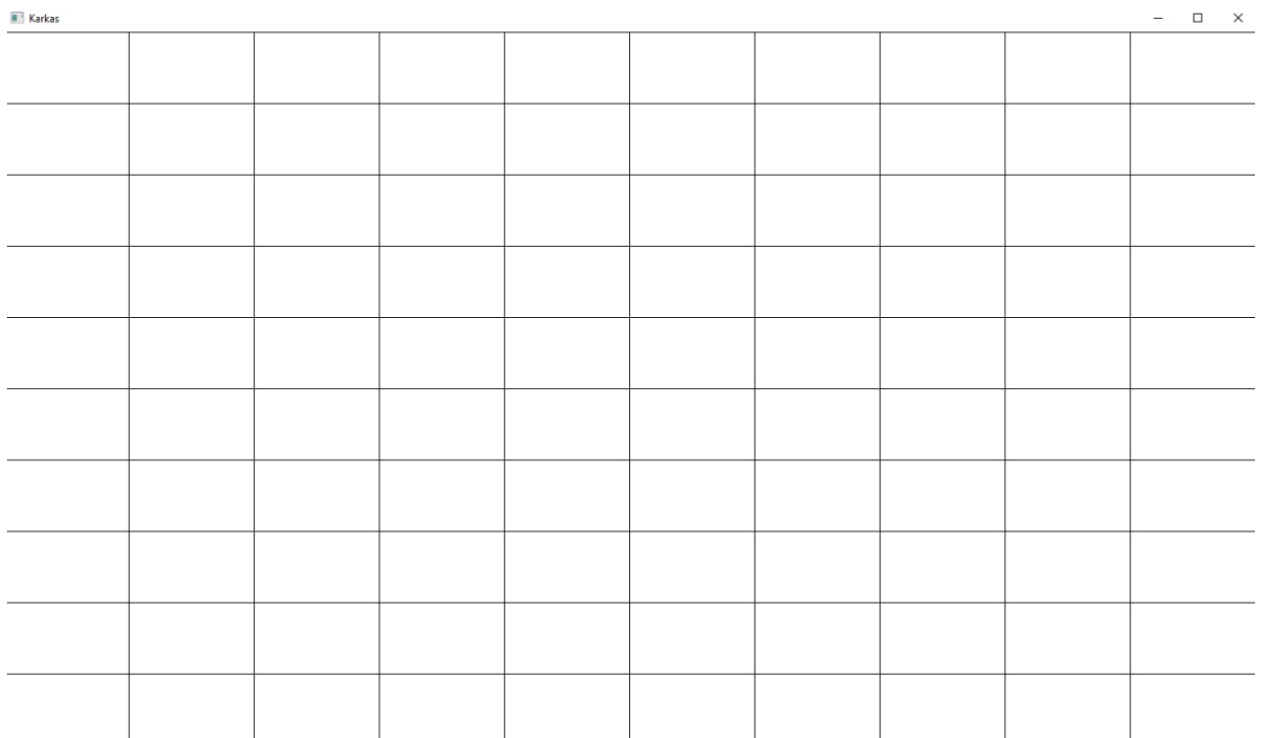


Рис.1. Сетка

ЗАДАНИЕ 2

Используя код для построения сети, отрисовать указанный вариант фигуры. Используя линию, точку, прямоугольник и т.п., выполнить построение фигур по варианту. Обязательно соблюдая цвета рисуемых объектов.

Вариант №12.

Код программы:

```
#include <windows.h>
#include <tchar.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");

#define MAX_LOADSTRING 100

int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd, int
mode)
{
    HWND hWnd;
    MSG msg;
    WNDCLASS wc;
    wc.hInstance = This;
    wc.lpszClassName = WinName;
    wc.lpfnWndProc = WndProc;
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.lpszMenuName = NULL;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;

    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
    if (!RegisterClass(&wc)) return 0;

    hWnd = CreateWindow(WinName, _T("Каркас Windows-приложения"),
WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT,
CW_USEDEFAULT, CW_USEDEFAULT, HWND_DESKTOP, NULL, This,
NULL);
```

```

ShowWindow(hWnd, mode);

while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
return 0;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc; int x, y; static int startx, starty, step, lenght , sx, sy;
    switch (message)
    {
    case WM_SIZE:
        sx = LOWORD(wParam);
        sy = LOWORD(lParam);
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        startx = LOWORD(600);
        starty = LOWORD(200);
        lenght = LOWORD(300);
        step = LOWORD(100);
        for (int i = 0; i <= 1; i++)
        {
            MoveToEx(hdc, startx - step * i, starty + step * i, NULL);
            LineTo(hdc, startx - step * i + lenght, starty + step * i + lenght);
        }
        startx = LOWORD(800);
        for (int i = 0; i <= 1; i++)
        {
            MoveToEx(hdc, startx + step * i, starty + step * i, NULL);
            LineTo(hdc, startx + step * i - lenght, starty + step * i + lenght);
        }
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY: PostQuitMessage(0);
        break;
    }
}

```

```
default: return DefWindowProc(hWnd, message, wParam, lParam);  
}  
return 0;  
}
```

Результат выполнения программы:

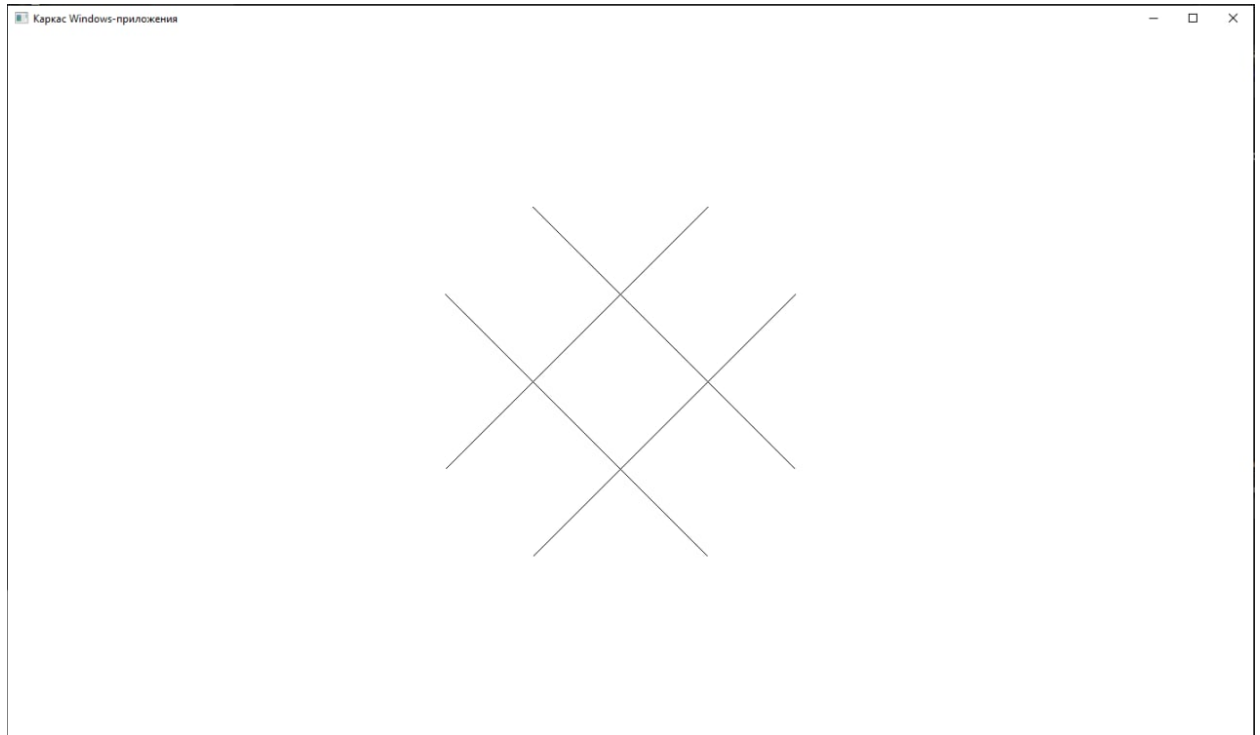


Рис.2. Поле для игры в крестики нолики под углом 45 градусов

ВЫВОД

В этой лабораторной работе были созданы 2 полноценные конструкции с реализацией графических объектов. В каждой программе используются различные функции, позволяющие графически изобразить задачу. Освоены новые навыки работы с базовыми элементами библиотеки.