



МИНОБРНАУКИ РОССИИ

**федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«Московский государственный технологический университет
«СТАНКИН»**

(ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт
информационных технологий**

**Кафедра
Инженерной Графики**

**Основная образовательная программа 09.03.02
«Информационные системы и технологии»**

**Отчет по дисциплине «Компьютерная геометрия и графика»
по лабораторной работе № 2**

**Студент
группы ИДБ-22-06**

Мустафаева П.М.

Преподаватель

к.т.н. , доц. Разумовский А.И.

Москва, 2023

ВВЕДЕНИЕ

В этой лабораторной работе будут рассмотрены понятия:

1. Функция
2. График
3. Циклы

Функция может возвращать значение, тип которого в большинстве случаев аналогично типу самой функции. Если функция не возвращает никакого значения, то она должна иметь тип `void` (такие функции иногда называются процедурами).

При объявлении функции, после ее типа должно находиться имя функции и две круглые скобки – открывающаяся и закрывающаяся, внутри которых могут находиться один или несколько аргументов функции, которых также может не быть вообще.

ЗАДАНИЕ 1

Используя исходный текст первого задания «Каркас приложения» дополните его недостающим кодом для построения графика «Синус».

Ниже представлен код для выполнения:

```
#include <windows.h>
#include <tchar.h>
#include <cmath>
#define _USE_MATH_DEFINES
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
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("Karkas"),
        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;
    static int sx, sy;
    static HPEN hpen1, hpen2;
    int a, b, x_scr, y_scr; //экранные координаты
    double x, h; //физические координаты
    const double PI = acos(-1.0);
    switch (message) {
    case WM_CREATE:
        hpen1 = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
        hpen2 = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));
        break;
    case WM_SIZE:
        sx = LOWORD(lParam);
        sy = HIWORD(lParam);
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        a = sx / 2; //координаты центра окна
        b = sy / 2;
        SelectObject(hdc, hpen1); //синее перо
        MoveToEx(hdc, 0, b, NULL);
        LineTo(hdc, sx, b);
        MoveToEx(hdc, a, 0, NULL);
        LineTo(hdc, a, sy);
        MoveToEx(hdc, 0, b, NULL);
        SelectObject(hdc, hpen2);
        h = 3 * PI / a;
        for (x = -PI, x_scr = 0; x < PI; x += h) {
            x_scr = (x + PI) * a / PI;
            y_scr = b - b * sin(x);
            LineTo(hdc, x_scr, y_scr);
        }
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY:
        DeleteObject(hpen1);
        DeleteObject(hpen2);
        PostQuitMessage(0);
        break;
    default:

```

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

Результат выполнения программы представлен ниже (рис. 1).

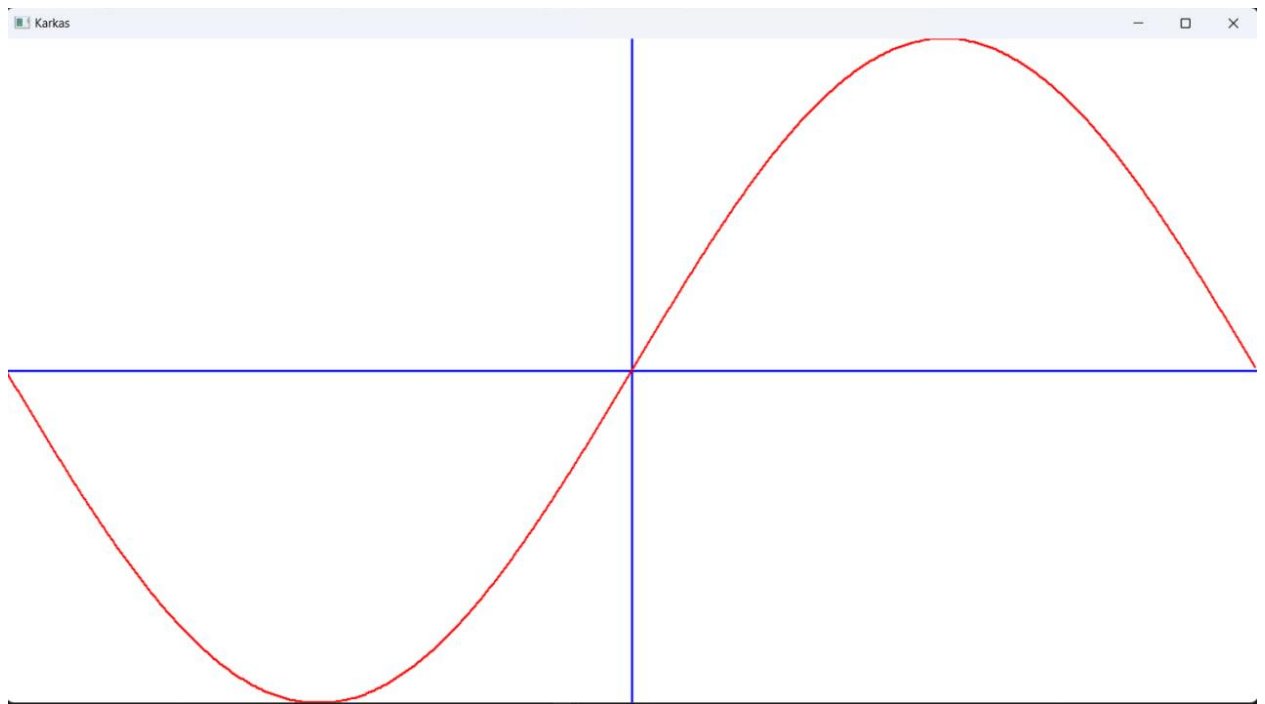


Рисунок 1. График «синус».

ЗАДАНИЕ 2

На основе полученного кода самостоятельно построить функцию «парабола».

Ниже представлена программа:

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam) {
    PAINTSTRUCT ps;
    HDC hdc;
    static int sx, sy;
    static HPEN hpen1, hpen2;
    int a, b; //экранные координаты
    double x, y, Ky, Kx, x_min, x_max, y_min, y_max, x1, y1; //физические
координаты
    switch (message) {
    case WM_CREATE:
        hpen1 = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
        hpen2 = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));
        break;
    case WM_SIZE:
        sx = LOWORD(lParam);
        sy = HIWORD(lParam);
        break;
    case WM_PAINT: {
        hdc = BeginPaint(hWnd, &ps);
        a = sx / 2; //координаты центра окна
        b = sy / 2;
        SelectObject(hdc, hpen1); //синее перо
        MoveToEx(hdc, 0, b, NULL);
        LineTo(hdc, sx, b);
        MoveToEx(hdc, a, 0, NULL);
        LineTo(hdc, a, sy);
        MoveToEx(hdc, 0, b, NULL);
        SelectObject(hdc, hpen2);
        x_min = -40;
        x_max = 40;
        y_min = -40;
        y_max = 40;
        Kx = sx / (x_max - x_min);
        Ky = sy / (y_max - y_min);
        double xphy = x_min * Kx + a;
        double yphy = -x_min * x_min * Ky + b;
```

```

MoveToEx(hdc, xphy, yphy, NULL);
for (x = x_min; x < x_max; x += 0.01) {
    xphy = x * Kx + a;
    yphy = -x * x * Ky + b;
    LineTo(hdc, xphy, yphy);
}

SelectObject(hdc, hpen1);
for (int i = 0; i <= 20; i++) {
    MoveToEx(hdc, a - 40 * i, b - 3, NULL);
    LineTo(hdc, a - 40 * i, b + 3);
    MoveToEx(hdc, a + 40 * i, b - 3, NULL);
    LineTo(hdc, a + 40 * i, b + 3);
}

for (int i = 0; i <= 15; i++) {
    MoveToEx(hdc, a - 3, b - 40 * i, NULL);
    LineTo(hdc, a + 3, b - 40 * i);
    MoveToEx(hdc, a - 3, b + 40 * i, NULL);
    LineTo(hdc, a + 3, b + 40 * i);
}

EndPaint(hWnd, &ps);
}

    break;
case WM_DESTROY:
    DeleteObject(hpen1);
    DeleteObject(hpen2);
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат выполнения программы представлен ниже (рис. 2).

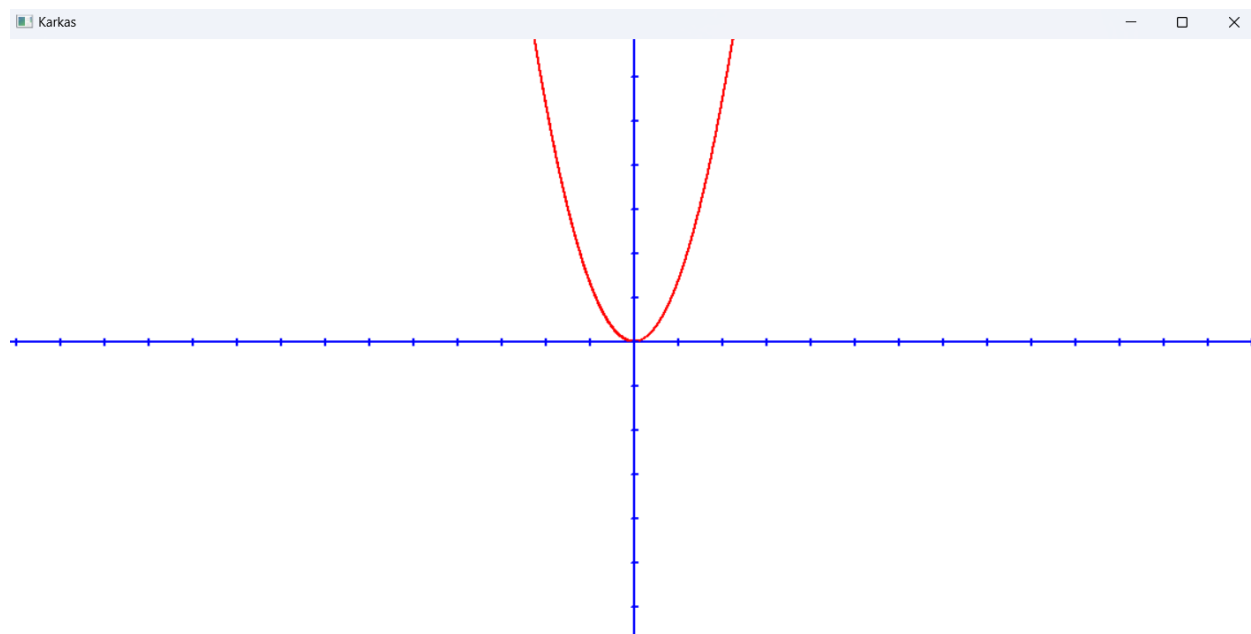


Рисунок 2. Функция «парабола».

ЗАДАНИЕ 3

Дополнить код «Каркас» фрагментами кода для построения поверхности функции.

Ниже представлена программа:

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam) {
    PAINTSTRUCT ps;
    HDC hdc;
    static HPEN hpen; //тип структуры пера
    static HPEN hpen1;
    static int sx, sy;
    int a, b, n, x_scr, y_scr, z_scr; //экранные координаты
    double x, y, z, x_min, x_max, y_min, y_max, z_min, z_max, Kx, Ky; //физ
координаты, габариты, коэфф масштабирования
    double M_PI = 3.14;
    switch (message) {
    case WM_CREATE:
        hpen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
        break;
    case WM_SIZE:
        sx = LOWORD(lParam);
        sy = HIWORD(lParam);
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        a = sx / 2;
        b = sy / 2;
        x_min = -100;
        y_min = -100;
        x_max = 100;
        y_max = 100;
        Kx = (sx - 220) / (x_max - x_min);
        Ky = (sy - 220) / (y_max - y_min);
        SelectObject(hdc, hpen);
        MoveToEx(hdc, 0, b, NULL);
        LineTo(hdc, sx, b);
        MoveToEx(hdc, a, 0, NULL);
        LineTo(hdc, a, sy);
        MoveToEx(hdc, 20, sy - 20, NULL);
        LineTo(hdc, sx - 20, 20); n = y_max - y_min;
        for (y = y_min; y < y_max; y += 5) {
```

```

        hpen1 = CreatePen(PS_SOLID, 2, RGB(255, 255 - 255 / n * (y + y_min),
255 - 255 / n * (y + y_min)));
        SelectObject(hdc, hpen1);
        z = b - 100 * sin(M_PI / 50 * sqrt(x_min * x_min + y * y));
        MoveToEx(hdc, x_min * Kx + a + y, z + y, NULL);
        for (x = x_min; x < x_max; x += 5) {
            z = b - 100 * sin(M_PI / 50 * sqrt(x * x + y * y));
            LineTo(hdc, x * Kx + a + y, z + y);
        }
    }
    EndPaint(hWnd, &ps);
    break;
case WM_DESTROY:
    DeleteObject(hpen);
    DeleteObject(hpen1);
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат выполнения программы представлен ниже (рис. 3).

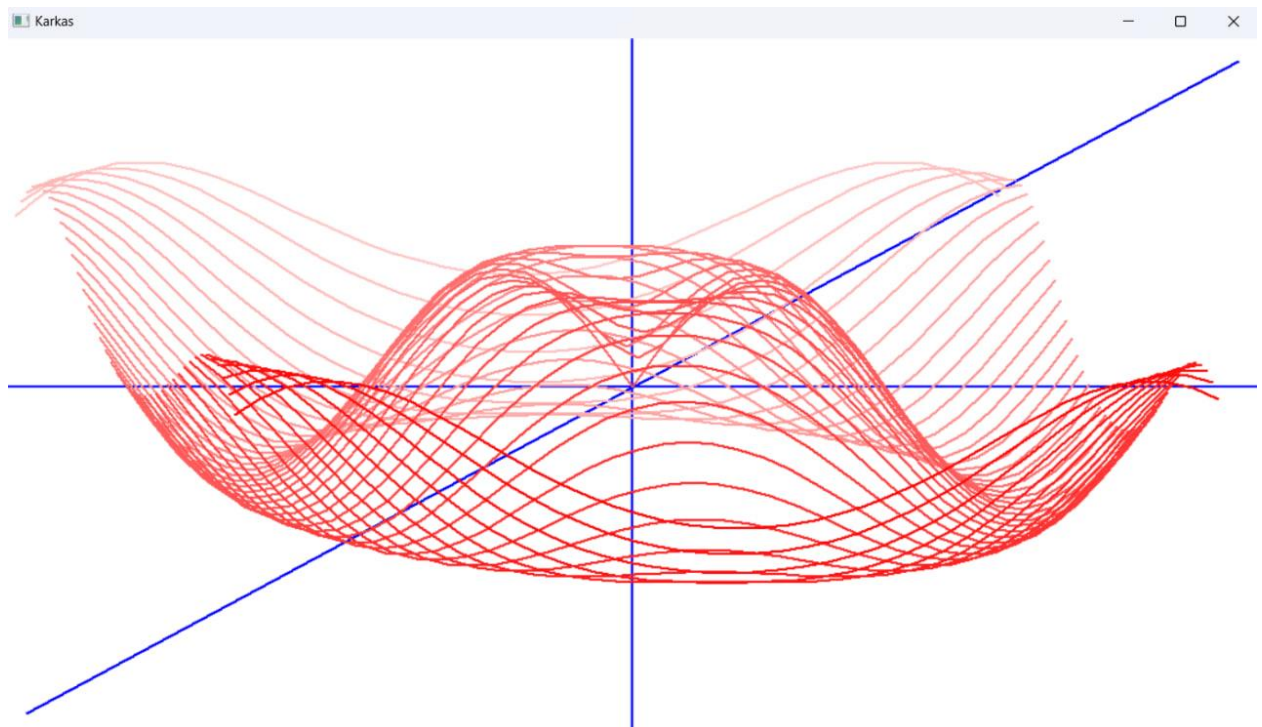


Рисунок 3. Поверхность.

ЗАДАНИЕ 4

На основе полученного кода самостоятельно построить функцию $y = 2^x$ (вариант №7).

Ниже представлена программа:

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam) {
    PAINTSTRUCT ps;
    HDC hdc;
    static int sx, sy;
    static HPEN hpen1, hpen2;
    int a, b; //экранные координаты
    double x, y, Ky, Kx, x_min, x_max, y_min, y_max, x1, y1; //физические
координаты
    switch (message) {
    case WM_CREATE:
        hpen1 = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
        hpen2 = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));
        break;
    case WM_SIZE:
        sx = LOWORD(lParam);
        sy = HIWORD(lParam);
        break;
    case WM_PAINT: {
        hdc = BeginPaint(hWnd, &ps);
        a = sx / 2; //координаты центра окна
        b = sy / 2;
        SelectObject(hdc, hpen1); //синее перо
        MoveToEx(hdc, 0, b, NULL);
        LineTo(hdc, sx, b);
        MoveToEx(hdc, a, 0, NULL);
        LineTo(hdc, a, sy);
        MoveToEx(hdc, 0, b, NULL);
        SelectObject(hdc, hpen2);
        x_min = -20;
        x_max = 20;
        y_min = -20;
        y_max = 20;
        Kx = sx / (x_max - x_min);
        Ky = sy / (y_max - y_min);
        double xphy = x_min * Kx + a;
        double yphy = b - pow(2, x_min) * Ky;
```

```

MoveToEx(hdc, xphy, yphy, NULL);
for (x = x_min; x < x_max; x += 0.01) {
    y = pow(2, x);
    xphy = x * Kx + a;
    yphy = b - y * Ky;
    LineTo(hdc, xphy, yphy);
}

EndPaint(hWnd, &ps);
}

break;
case WM_DESTROY:
    DeleteObject(hpen1);
    DeleteObject(hpen2);
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат выполнения программы представлен ниже (рис. 4).

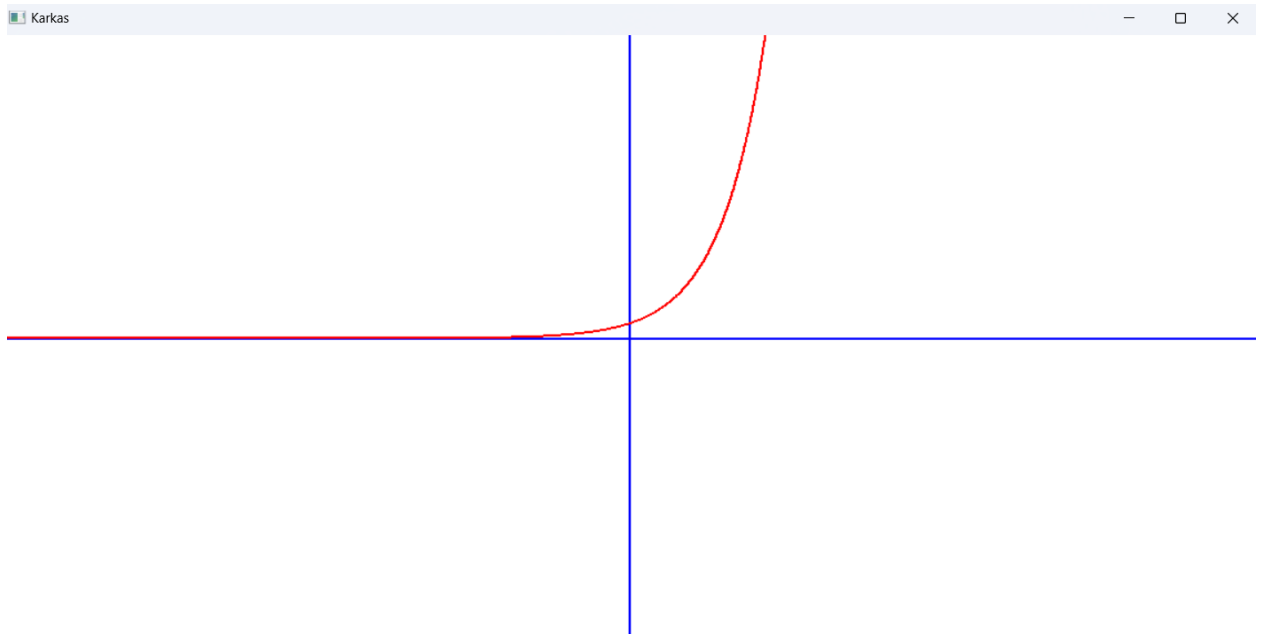


Рисунок 4. График функции $y = 2^x$.

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы были рассмотрены и изучены необходимые аспекты: функция, график, циклы. В результате работы получилось создать несколько графических объектов, представляющие собой графики.