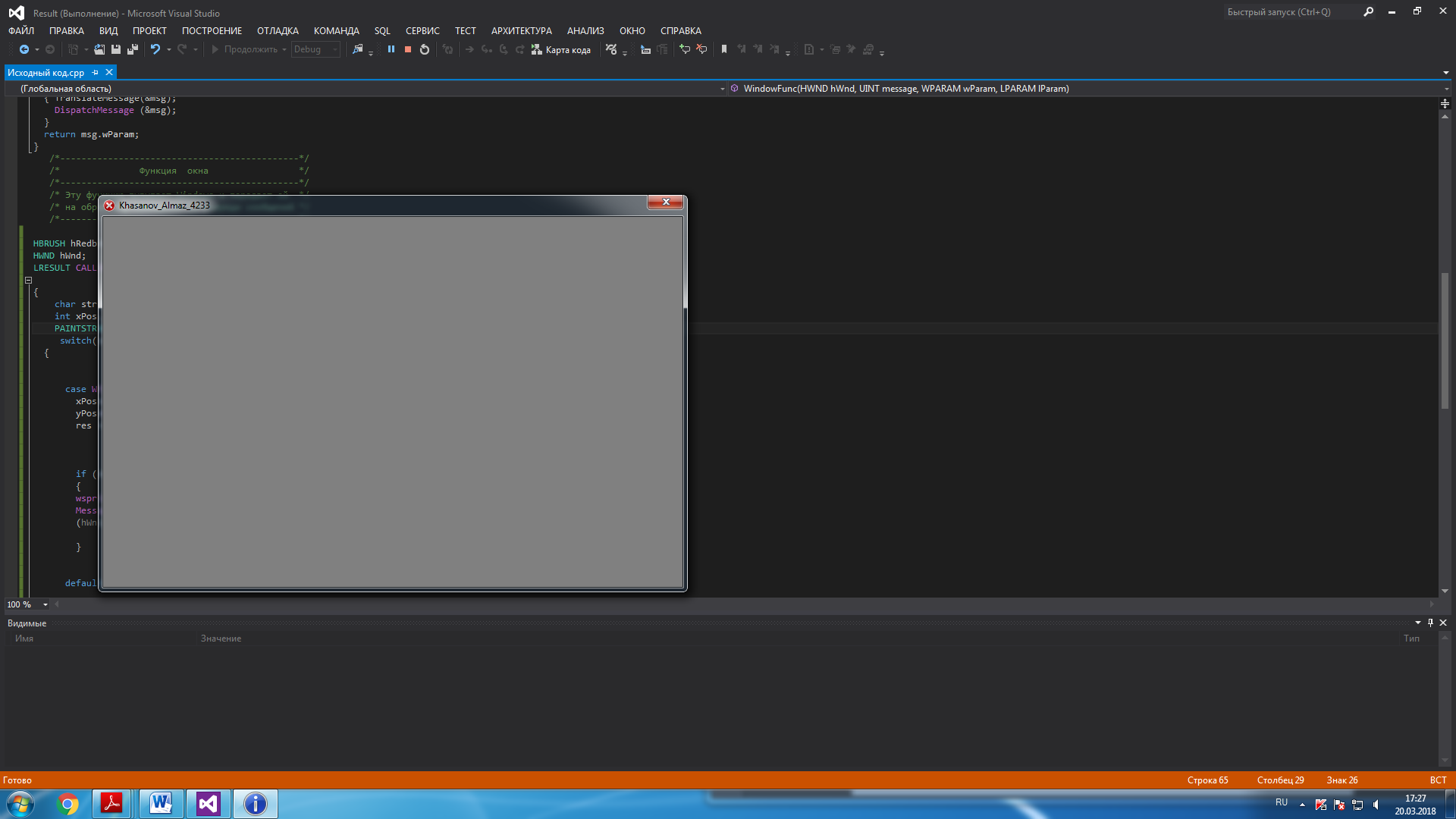
Хасанов Алмаз 4233.

1. Создал графическое приложение в среде Microsoft Visual Studio.
2. Проверил работу приложения.
3. Изменил класс окна на Lab4Group4233Var12Tools.
4. Изменил атрибуты окна: заголовок, координаты и размеры окна, цвет фона, тип курсора, пиктограммы, способ отображения окна, стиль окна, чтобы нельзя было изменять мышкой размеры окна и окно не имело кнопок минимизации и максимизации окна.
5. Запустил программу, сделал скриншот:

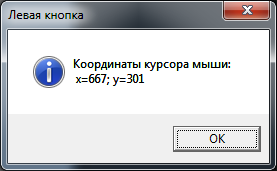
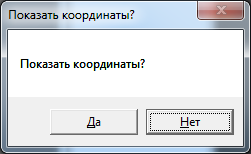


1. Изменил программу, чтобы при нажатии левой кнопки мыши

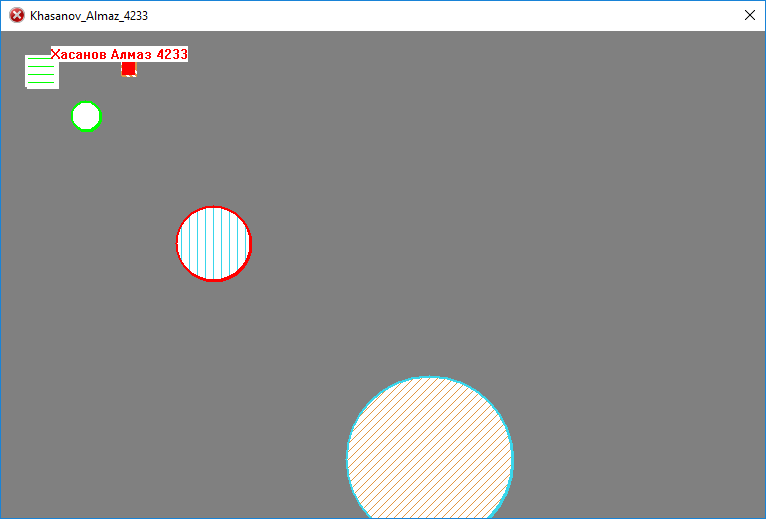
выводилась панель сообщения, а при нажатии кнопки “Да” на этой панели выводилась еще одна панель с

координатами курсора мыши.

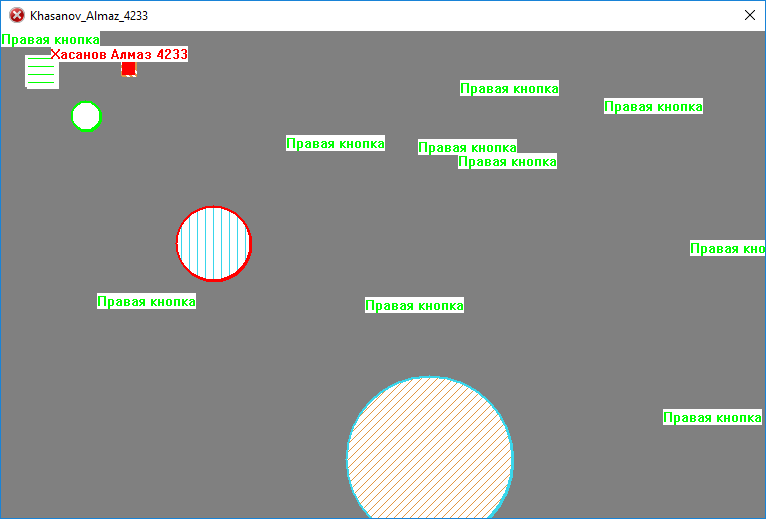
1. Запустил программу и сделал скриншоты:



1. Организовал вывод графических объектов при обработке сообщения WM\_PAINT. Нарисовал 2 квадрата и 3 круга разными цветами и вывел текстовое сообщение в точке x=50, y=15.
2. Проверил рвботу программы.
3. Сделал скриншот:



1. Добавил обработку сообщения о нажатии правой кнопкой мыши. В координатах нажатия выводится текст “Правая кнопка”.
2. Проверил работу программы.
3. Сделал скриншот:



Код программы

#include "windows.h"

/\* объявление функции окна \*/

LRESULT CALLBACK WindowFunc(HWND, UINT, WPARAM, LPARAM);

char szWinName[] = "Lab4Group4233Var12Tools"; // Имя "класса" окна

/\* Главная функция \*/

int WINAPI WinMain(HINSTANCE hThisInst, HINSTANCE hPrevInst,

LPSTR lpszArgs, int nWinMode)

{

HWND hWnd; // дескриптор окна

MSG msg; // сообщение

WNDCLASSEX wcl; // "класс" окна /\* Определение "класса" (стиля) окна \*/

wcl.hInstance = hThisInst; /\*дескриптор данного экземпляра\*/

wcl.lpszClassName = (LPCSTR)szWinName; // имя "класса" окна

wcl.lpfnWndProc = WindowFunc; // функция окна

wcl.style = 0; // стиль по умолчанию

wcl.cbSize = sizeof(WNDCLASSEX); // размер структуры

wcl.hIcon = LoadIcon(NULL, IDI\_ASTERISK); //больш.иконка

wcl.hIconSm = LoadIcon(NULL, IDI\_HAND); // малая иконка

wcl.hCursor = LoadCursor(NULL, IDC\_WAIT); //форма курсора

wcl.lpszMenuName = NULL; // меню не используется

wcl.cbClsExtra = 0; // дополнит. информации нет

wcl.cbWndExtra = 0;

/\* Фон окна задается белым \*/

wcl.hbrBackground = (HBRUSH)GetStockObject(GRAY\_BRUSH);

/\* Регистрация "класса" окна \*/

if (!RegisterClassEx(&wcl)) return 0;

/\* Создание окна \*/

hWnd = CreateWindowEx

(0, (LPCSTR)szWinName, // имя "класса" окна

(LPCSTR)"Khasanov\_Almaz\_4233", // заголовок

WS\_OVERLAPPED | WS\_SYSMENU, // стандартное окно

128, // координата Х - по умолчанию

256, // координата Y - по умолчанию

780, // ширина - по умолчанию

526, // высота - по умолчанию

HWND\_DESKTOP, // родительского окна нет

NULL, // меню нет

hThisInst, // дескриптор данного экземпляра приложения

NULL // дополнительных аргументов нет

);

/\* Отображение окна \*/

ShowWindow(hWnd, nWinMode);

/\* Цикл сообщений \*/

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

/\*---------------------------------------------\*/

/\* Функция окна \*/

/\*---------------------------------------------\*/

/\* Эту функцию вызывает Windows и передает ей \*/

/\* на обработку сообщения из очереди сообщений \*/

/\*---------------------------------------------\*/

HBRUSH hRedbrush, hWhitebrush, hLightbluebrush, hOrangebrush, hGreenbrush;

HPEN hRedpen, hWhitepen, hLightbluepen, hOrangepen, hGreenpen;

HWND hWnd;

int xPosR, yPosR;

LRESULT CALLBACK WindowFunc(HWND hWnd, UINT message,

WPARAM wParam, LPARAM lParam)

{

char str[40] = "Показать координаты?";

int xPos, yPos, res;

PAINTSTRUCT paintstruct;

switch (message)

{

case WM\_RBUTTONDOWN:

xPosR = LOWORD(lParam);

yPosR = HIWORD(lParam);

InvalidateRect(hWnd, NULL, 0);

break;

case WM\_PAINT: {

hGreenpen = CreatePen(PS\_SOLID, 2, RGB(0, 255, 0));

hGreenbrush = CreateHatchBrush(HS\_HORIZONTAL, RGB(0, 255, 0));

hRedpen = CreatePen(PS\_DASH, 2, RGB(255, 0, 0));

hRedbrush = CreateSolidBrush(RGB(255, 0, 0));

hWhitepen = CreatePen(PS\_DOT, 3, RGB(255, 255, 255));

hWhitebrush = CreateSolidBrush(RGB(255, 255, 255));

hLightbluebrush = CreateHatchBrush(HS\_VERTICAL, RGB(57, 214, 235));

hLightbluepen = CreatePen(PS\_DASHDOT, 2, RGB(57, 214, 235));

hOrangebrush = CreateHatchBrush(HS\_BDIAGONAL, RGB(225, 150, 67));

hOrangepen = CreatePen(PS\_DASHDOTDOT, 1, RGB(225, 150, 67));

COLORREF c1, c2, c3, c4, c5;

c1 = RGB(255, 0, 0);

c2 = RGB(0, 255, 0);

c3 = RGB(0, 0, 0);

c4 = RGB(255, 255, 255);

c5 = RGB(0, 0, 355);

HDC hdc = BeginPaint(hWnd, &paintstruct);

SelectObject(hdc, hWhitepen);

Rectangle(hdc, 27, 27, 57, 57);

SelectObject(hdc, hGreenbrush);

Rectangle(hdc, 25, 25, 55, 55);

SelectObject(hdc, hOrangepen);

Rectangle(hdc, 121, 31, 136, 46);

SelectObject(hdc, hRedbrush);

Rectangle(hdc, 120, 30, 135, 45);

SelectObject(hdc, hGreenpen);

Ellipse(hdc, 71, 71, 101, 101);

SelectObject(hdc, hWhitebrush);

Ellipse(hdc, 70, 70, 100, 100);

SelectObject(hdc, hRedpen);

Ellipse(hdc, 176, 176, 251, 251);

SelectObject(hdc, hLightbluebrush);

Ellipse(hdc, 175, 175, 250, 250);

SelectObject(hdc, hLightbluepen);

Ellipse(hdc, 513, 513, 346, 346);

SelectObject(hdc, hOrangebrush);

Ellipse(hdc, 512, 512, 345, 345);

SetTextColor(hdc, c1);

SetBkColor(hdc, c4);

TextOut(hdc, 50, 15, "Хасанов Алмаз 4233", 18);

SetTextColor(hdc, c2);

SetBkColor(hdc, c4);

TextOut(hdc, xPosR, yPosR, "Правая кнопка", 13);

EndPaint(hWnd, &paintstruct);

DeleteObject(hGreenbrush);

DeleteObject(hRedbrush);

DeleteObject(hWhitebrush);

DeleteObject(hLightbluebrush);

DeleteObject(hOrangebrush);

DeleteObject(hGreenpen);

DeleteObject(hRedpen);

DeleteObject(hWhitepen);

DeleteObject(hLightbluepen);

DeleteObject(hWhitepen);

break;

}

case WM\_DESTROY: /\* "завершить программу" \*/

PostQuitMessage(0);

break;

case WM\_LBUTTONDOWN:

xPos = LOWORD(lParam);

yPos = HIWORD(lParam);

res = MessageBox(hWnd, str, "Показать координаты?", MB\_YESNO);

if (res == IDYES)

{

wsprintf(str, "Координаты курсора мыши:\n x=%d; y=%d", xPos, yPos);

MessageBox

(hWnd, str, "Левая кнопка", MB\_OK | MB\_ICONINFORMATION);

}

break;

default:

/\* Остальные сообщения обрабатывать \*/

/\* операционной системе \*/

return DefWindowProc(hWnd, message, wParam, lParam);

}

return 0;

}