|  |
| --- |
| **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** |

|  |  |
| --- | --- |
| **Институт** | ИВТИ |
| **Кафедра** | ПМИИ |

|  |  |
| --- | --- |
|  |  |
| **Дисциплина: «Системное программирование»**  **Отчет по лабораторной работе №7**  **«Битовые карты и метафайлы»**  **Выполнил: студент группы А-13а-19**  **Рамазанов Н. М.**  **Преподаватель: Меньшикова К. Г.**  **Москва, 2021** | |

**1. Краткое описание функций, параметров и используемых событий.**

1) Работа с битовыми картами в WinAPI.

* **Дескриптор HBITMAP**

Дескриптор растрового изображения.

* **CreateCompatibleBitmap**

**HBITMAP CreateCompatibleBitmap (HDC** *hdc***, int** *nWidth***, int** *nHeight***);**

Функция **CreateCompatibleBitmap** создает точечный рисунок, совместимый с устройством, которое связано с заданным контекстом устройства.

Параметры: *hdc* – дескриптор контекста устройства; *nWidth* – определяет ширину точечного рисунка, в пикселях; *nHeight* – определяет высоту точечного рисунка, в пикселях.

Если функция завершается успешно, возвращаемое значение - дескриптор совместимого точечного рисунка (аппаратно-зависимая точечная картинка (DDB)). Если функция завершается ошибкой, возвращаемое значение - ПУСТО (NULL).

* **CreateCompatibleDC**

**HDC CreateCompatibleDC(HDC***hdc***);**

Функция CreateCompatibleDC создает контекст устройства в памяти (DC), совместимый с заданным устройством.

Параметр: *hdc* – дескриптор существующего контекста устройства (DC). Если этот дескриптор равен ПУСТО (NULL), функция создает контекст устройства в памяти, совместимый с текущим экраном приложения.

Если функция завершается успешно, возвращаемое значение - дескриптор контекста устройства в памяти. Если функция завершается ошибкой, возвращаемое значение - ПУСТО (NULL).

* **BitBlt**

**BOOL BitBlt (HDC***hdcDest***, int***nXDest***, int***nYDest***, int***nWidth***, int***nHeight***, HDC***hdcSrc***, int***nXSrc***, int***nYSrc***, DWORD***dwRop***);**

Функция BitBlt выполняет передачу битовых блоков данных о цвете, соответствующих прямоугольнику пикселей из заданного исходного контекста устройства в целевой контекст устройства.

Параметры: *hdcDest* – дескриптор целевого контекста устройства; *nXDest*, *nYDest* – устанавливают x- и y-координату, в логических единицах измерения, левого верхнего угла целевого прямоугольника; *nWidth*, *nHeight* – устанавливают ширину и высоту, в логических единицах измерения, исходного и целевого прямоугольников; *hdcSrc* – дескриптор источникового контекста устройства; *nXSrc*, *nYSrc* – устанавливают x- и y-координату, в логических единицах измерения, левого верхнего угла исходного прямоугольника; *dwRop* – устанавливает код растровой операции. Эти коды определяют, как данные о цвете для исходного прямоугольника должны быть объединены с данными о цвете для целевого прямоугольника, чтобы достигнуть конечного цвета. Список ниже показывает некоторые общие коды растровых операций.

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| **BLACKNESS** | Заполняет целевой прямоугольник, используя цвет, связанный с индексом 0 (черный) в физической палитре. |
| **DSTINVERT** | Инвертирует целевой прямоугольник. |
| **MERGECOPY** | Объединяет цвета исходного прямоугольника с кистью в текущий момент выбранной в hdcDest, при помощи использования булева оператора AND. |
| **MERGEPAINT** | Объединяет цвета инвертированного исходного прямоугольника с цветами целевого прямоугольника при помощи использования булева оператора OR. |
| **NOTSRCCOPY** | Копирует инвертированный исходный прямоугольник в целевой. |
| **NOTSRCERASE** | Комбинирует цвета исходных и целевых прямоугольников при помощи использования булева оператора OR и затем инвертирует получающийся в результате цвет. |
| **PATCOPY** | Копирует кисть, в текущий момент выбранную в hdcDest, в целевой точечный рисунок. |
| **PATINVERT** | Комбинирует цвета кисти, в текущий момент выбранной в hdcDest, с цветами целевого прямоугольника при помощи использования булева оператора XOR. |
| **PATPAINT** | Комбинирует цвета кисти, в текущий момент выбранной в hdcDest, с цветами инвертированного исходного прямоугольника при помощи использования булева оператора OR. Результаты этой операции объединяются с цветами целевого прямоугольника при помощи использования булева оператора OR. |
| **SRCAND** | Комбинирует цвета исходных и целевых прямоугольников при помощи использования булева оператора AND. |
| **SRCCOPY** | Копирует исходный прямоугольник непосредственно в целевой прямоугольник. |
| **SRCERASE** | Комбинирует инвертированные цвета целевого прямоугольника с цветами исходного прямоугольника при помощи использования булева оператора AND. |
| **SRCINVERT** | Комбинирует цвета источникового и целевого прямоугольников при помощи использования булева оператора XOR. |
| **SRCPAINT** | Комбинирует цвета источникового и целевого прямоугольников при помощи использования булева оператора OR. |
| **WHITENESS** | Заполняет целевой прямоугольник, используя цвет, связанный с индексом 1 (белый) в физической палитре. |

Если функция завершается успешно, величина возвращаемого значения - не нуль. Если функция завершается с ошибкой, величина возвращаемого значения - нуль.

* **DeleteDC**

**BOOL DeleteDC(HDC***hdc***);**

Функция DeleteDC удаляет заданный контекст устройства (DC).

Параметр: *hdc* – дескриптор контекста устройства (DC).

Если функция завершается успешно, величина возвращаемого значения - не нуль. Если функция завершается с ошибкой, величина возвращаемого значения - нуль.

* **LoadBitmap**

**HBITMAP LoadBitmap(HINSTANCE***hInstance***, LPCTSTR***lpBitmapName***);**

Функция LoadBitmap загружает заданный ресурс растрового изображения из модуля исполняемого файла.

Параметры: *hInstance* – дескриптор экземпляра модуля, исполняемый файл которого содержит загоужаемый точечный рисунок; *lpBitmapName* – указатель на символьную строку с нулем в конце, которая содержит название ресурса загружаемого растрового изображения.

Если функция завершается успешно, возвращаемое значение - дескриптор устанавливаемого точечного рисунка. Если функция завершается с ошибкой, возвращаемое значение - ПУСТО (NULL).

2) Работа с битовыми картами на C#.

Класс ***System.Drawing.Bitmap.***

|  |  |
| --- | --- |
| ***Конструкторы*** | |
| **Bitmap(Image)** | Инициализирует новый экземпляр класса Bitmap из указанного существующего изображения. |
| **Bitmap(Image, Int32, Int32)** | Инициализирует новый экземпляр класса Bitmap из указанного существующего изображения, масштабированного до заданного размера. |
| **Bitmap(Int32, Int32)** | Инициализирует новый экземпляр класса Bitmap с заданным размером. |
| **Bitmap(Int32, Int32, Graphics)** | Инициализирует новый экземпляр класса Bitmap с заданным размером и с разрешением указанного объекта Graphics. |
| ***Методы*** | |
| **DrawImage (<Bitmap>, x, y,Width, Height)** | Рисует заданный объект Image в заданном месте, используя указанный размер. |
| **DrawImage(<Bitmap>,<Rect>,<Rect>, GraphicsUnit.Pixel)** | Рисует заданную часть указанного объекта Image в заданном месте, используя заданный размер. |
| **GetPixel** | Возвращает цвет указанного пикселя в этом изображении Bitmap. |
| **SetPixel** | Задает цвет указанного пикселя в этом объекте Bitmap. |
| **MakeTransparent()** | Делает прозрачным прозрачный цвет по умолчанию для этого элемента Bitmap. |
| **MakeTransparent(Color)** | Делает заданный цвет прозрачным для данного изображения Bitmap. |
| **Dispose** | Выполняет определяемые приложением задачи, связанные с удалением, высвобождением или сбросом неуправляемых ресурсов |
| **Save(<имя файла>)** | Сохраняет объект Image в указанный файл или поток |

3) WinAPI. Работа с метафайлами.

* **Дескриптор HMETAFILE**

Microsoft Windows Metafile Format (WMF) используется для хранения векторных и растровых изображений и графических данных в памяти. Векторные данные, хранимые WMF-файлом, описывают команды Microsoft Windows Graphics Device Interface (GDI).

* **CreateMetaFile(NULL)**

**HDC CreateMetaFile(LPCSTR** *pszFile***);**

Функция CreateMetaFile создает контекст устройства для метафайла.

Параметр: *pszFile* – указывает на имя файла для создаваемого метафайла в формате Windows.

Если функция выполняется успешно, возвращаемое значение является дескриптором контекста устройства для метафайла в формате Windows. Если функция завершается неудачно, возвращаемое значение равно NULL.

* **CloseMetaFile**

**HDC CloseMetaFile(HDC** *hdc***);**

Функция CloseMetaFile закрывает контекст устройства метафайла.

Параметр: *hdc* – дескриптор контекста устройства для метафайла в формате Windows.

Если функция выполняется успешно, возвращаемое значение является дескриптором контекста устройства для метафайла в формате Windows. Если функция завершается неудачно, возвращаемое значение равно NULL.

* **PlayMetaFile**

**HDC PlayMetaFile(HDC** *hdc*, **HMETAFILE** *hmf***);**

Функция PlayMetaFile отображает изображение, сохраненное в данном метафайле формата Windows на указанном устройстве.

Параметры: *hdc* – дескриптор контекста устройства для метафайла в формате Windows; *hmf* – дескриптор метафайла в формате Windows.

Если функция выполняется успешно, возвращаемое значение – не нуль. Если функция завершается неудачно, возвращаемое значение равно нулю.

* **CopyMetaFile**

**HDC CopyMetaFile(HMETAFILE** *hmf*, **LPCSTR** *lpszFile***);**

Функция CopyMetaFile копирует содержимое метафайла формата Windows в указанный файл.

Параметры: *hmf* – дескриптор исходного метафайла в формате Windows; *lpszFile* – указатель на имя файла назначения. Если этот параметр NULL, исходный метафайл копируется в память.

Если функция выполняется успешно, возвращаемое значение является дескриптором контекста устройства для метафайла в формате Windows. Если функция завершается неудачно, возвращаемое значение равно нулю.

* **DeleteMetaFile**

**BOOL DeleteMetaFile(HMETAFILE** *hmf***);**

Функция CopyMetaFile копирует содержимое метафайла формата Windows в указанный файл.

Параметр: *hmf* – дескриптор исходного метафайла в формате Windows.

Если функция выполняется успешно, возвращаемое значение – не нуль. Если функция завершается неудачно, возвращаемое значение равно нулю.

* **GetMetaFile**

**HMETAFILE GetMetaFile(LPCSTR** *lpName***);**

Функция CopyMetaFile копирует содержимое метафайла формата Windows в указанный файл.

Параметр: *lpName* – указатель на строку, заканчивающуюся нулем, которая указывает имя метафайла.

Если функция выполняется успешно, возвращаемое значение является дескриптором метафайла в формате Windows. Если функция завершается неудачно, возвращаемое значение равно нулю.

**2. Выполнение работы.**

Для выполнения работы создал два приложения: программа WinAPI и программа на языке C#.

1. Перед началом работы приложения WinAPI создал две картинки (битовые карты) и включил их в ресурсы программы. Программный код обеспечивает вывод на рабочую поверхность окна большой картинки, а вторая (небольшая) картинка медленно перемещается по ее поверхности (анимация). Картинки составляют единый сюжет.

Необходимые переменные:

HBITMAP hBackImg; // картинка-фон

HBITMAP hPlaneImg; // картинка самолета

HWND hAnimateBtn; // кнопка запуска анимации

const static int idAnimateBtn = 0;

const static int idTimer = 2;

bool is\_animate = false;

RECT mainRect = { -10, 300, 360, 440 }; // начальное положение самолета

int x\_plane = 0; // х-координата его левой границы

int hWidth = 1200, hHeight = 800; // ширина и высота окна

Функция отрисовки битовой карты:

VOID DrawBitmap(HDC hDC, int x, int y, HBITMAP hBitmap) {

HBITMAP old\_bitmap;

HDC hImgDC;

BITMAP bitmap;

POINT ptSize, ptOrg;

hImgDC = CreateCompatibleDC(hDC);

old\_bitmap = (HBITMAP)SelectObject(hImgDC, hBitmap);

if (old\_bitmap)

{

SetMapMode(hImgDC, GetMapMode(hDC));

GetObject(hBitmap, sizeof(BITMAP), (LPSTR)&bitmap);

ptSize.x = bitmap.bmWidth;

ptSize.y = bitmap.bmHeight;

DPtoLP(hDC, &ptSize, 1);

ptOrg.x = 0;

ptOrg.y = 0;

DPtoLP(hImgDC, &ptOrg, 1);

BitBlt(hDC, x, y, ptSize.x, ptSize.y,

hImgDC, ptOrg.x, ptOrg.y, SRCCOPY);

SelectObject(hImgDC, old\_bitmap);

}

DeleteDC(hImgDC);

}

Кнопка:

hAnimateBtn = CreateWindow(L"BUTTON", L"Запустить анимацию", WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON, 0, 0, 200, 25, hWnd, (HMENU)idAnimateBtn, hInstance, NULL);

Добавим реакцию на события: в WM\_CREATE вызовем LoadBitmap для обеих картинок, WM\_TIMER для движения самолета.

case WM\_CREATE:

{

hBackImg = LoadBitmap(hInst, MAKEINTRESOURCE(IDB\_BACK));

hPlaneImg = LoadBitmap(hInst, MAKEINTRESOURCE(IDB\_PLANE));

}

break;

case WM\_TIMER:

{

if (x\_plane < hWidth) {

x\_plane += 15;

mainRect.left += 15;

mainRect.right += 15;

InvalidateRect(hWnd, &mainRect, FALSE);

UpdateWindow(hWnd);

}

else {

x\_plane = -360;

mainRect.left = -370;

mainRect.right = 0;

}

}

break;

В WM\_COMMAND добавим реакцию на нажатие кнопки:

case idAnimateBtn:

{

is\_animate = !is\_animate;

if (is\_animate)

SetTimer(hWnd, idTimer, 200, NULL);

else

KillTimer(hWnd, idTimer);

}

break;

В WM\_PAINT – отрисовку битовых карт, а в WM\_DESTROY – их удаление.

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

DrawBitmap(hdc, 0, 0, hBackImg);

DrawBitmap(hdc, x\_plane, mainRect.top, hPlaneImg);

EndPaint(hWnd, &ps);

}

break;

case WM\_DESTROY:

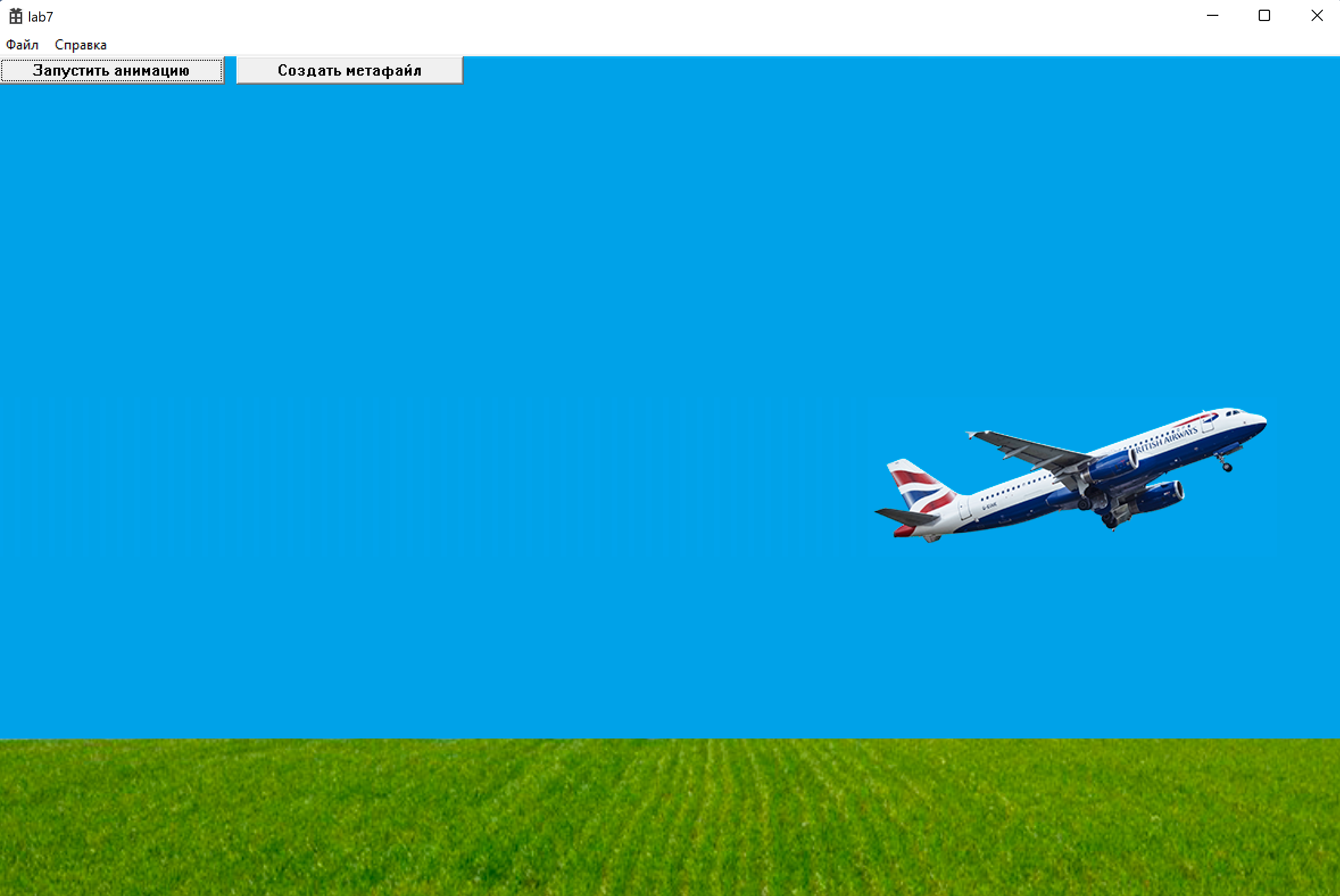
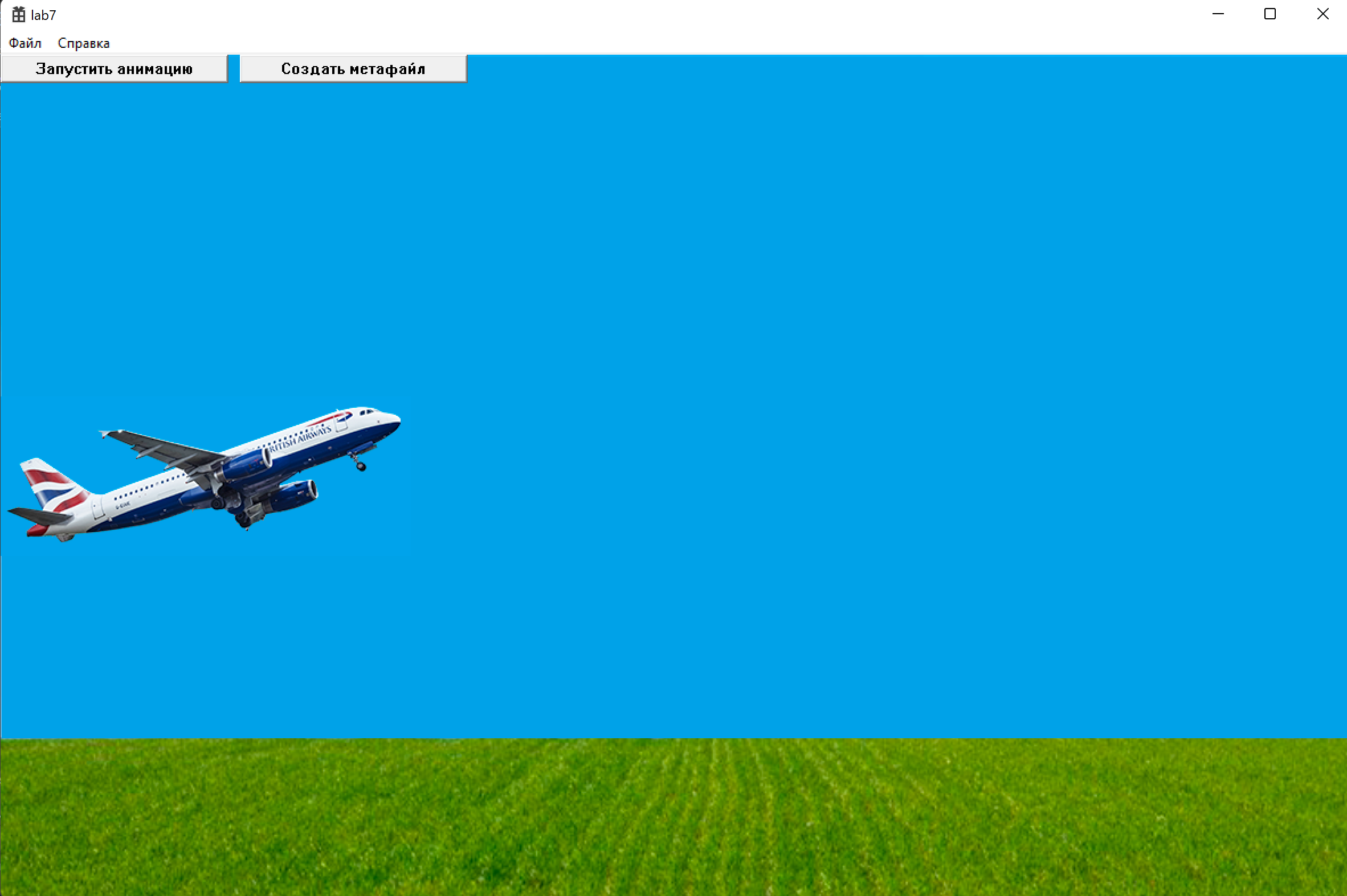
PostQuitMessage(0);

DeleteObject(hBackImg);

DeleteObject(hPlaneImg);

break;

Ссылка на видео – демонстрацию работоспособности: <https://disk.yandex.ru/i/OcEfmkIG5boYtA>



1. В приложении C# создал слайд-презентацию. На слайде постепенно проявляются (двигаются, приближаются, удаляются и т.д.) различные картинки – битовые карты (графики, диаграммы, объекты и т.д.), а также выводится соответствующий текст.

По нажатию кнопки вызываем функцию вывода изображений:

private void button1\_Click(object sender, EventArgs e)

{

ShowPresentation();

}

Функция вывода изображений:

private void ShowPresentation()

{

Graphics GB = PB.CreateGraphics();

Font font = new Font("Comic Sans MS", 36, FontStyle.Bold);

System.Drawing.Bitmap bitmap;

GB.Clear(Color.White);

bitmap = new Bitmap(@"D:\учеба\5th\_semester\СистПрог\lab7\lab7\_CSharp\Screenshot\_1.png");

for (float i = 0; i <= 10; i++)

{

GB.DrawImage(bitmap, bitmap.Width - i \* 76, bitmap.Height - i \* 48, bitmap.Width, bitmap.Height);

System.Threading.Thread.Sleep(50);

}

GB.DrawString("Ирис щетинистый", font, Brushes.White, 0, 0);

bitmap.Dispose();

System.Threading.Thread.Sleep(2000);

GB.Clear(Color.White);

bitmap = new Bitmap(@"D:\учеба\5th\_semester\СистПрог\lab7\lab7\_CSharp\Screenshot\_2.png");

for (int i = 10; i >= 0; --i)

{

GB.DrawImage(bitmap, 20 \* i, 16 \* i, bitmap.Width - 40 \* i, bitmap.Height - 32 \* i);

System.Threading.Thread.Sleep(50);

}

GB.DrawString("Ирис виргинский", font, Brushes.Lime, 140, 200);

bitmap.Dispose();

System.Threading.Thread.Sleep(2000);

GB.Clear(Color.White);

bitmap = new Bitmap(@"D:\учеба\5th\_semester\СистПрог\lab7\lab7\_CSharp\Screenshot\_3.png");

for (int i = 12; i > 0; --i)

{

GB.DrawImage(bitmap, 0, 0, bitmap.Width / i, bitmap.Height / i);

System.Threading.Thread.Sleep(50);

}

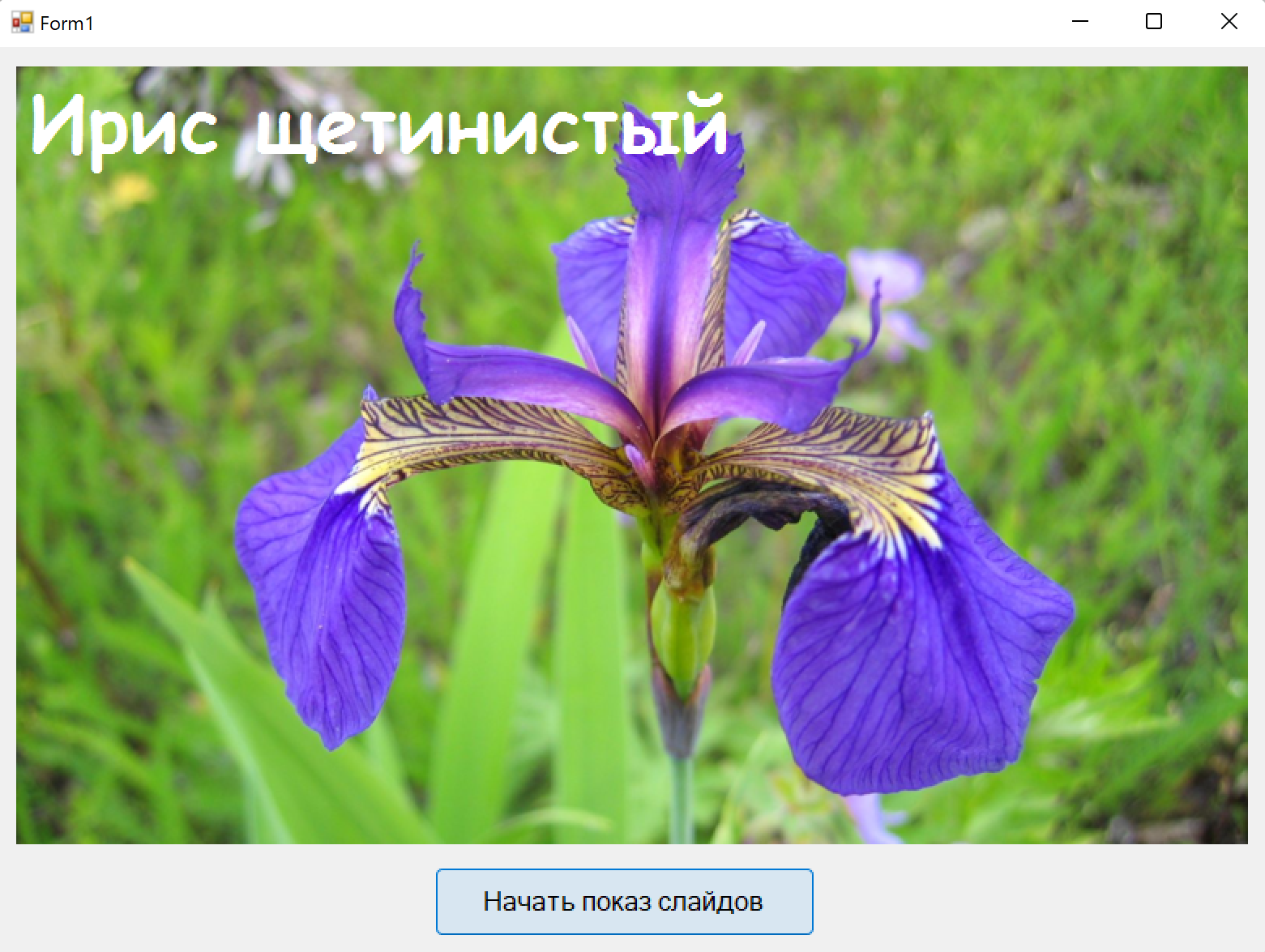
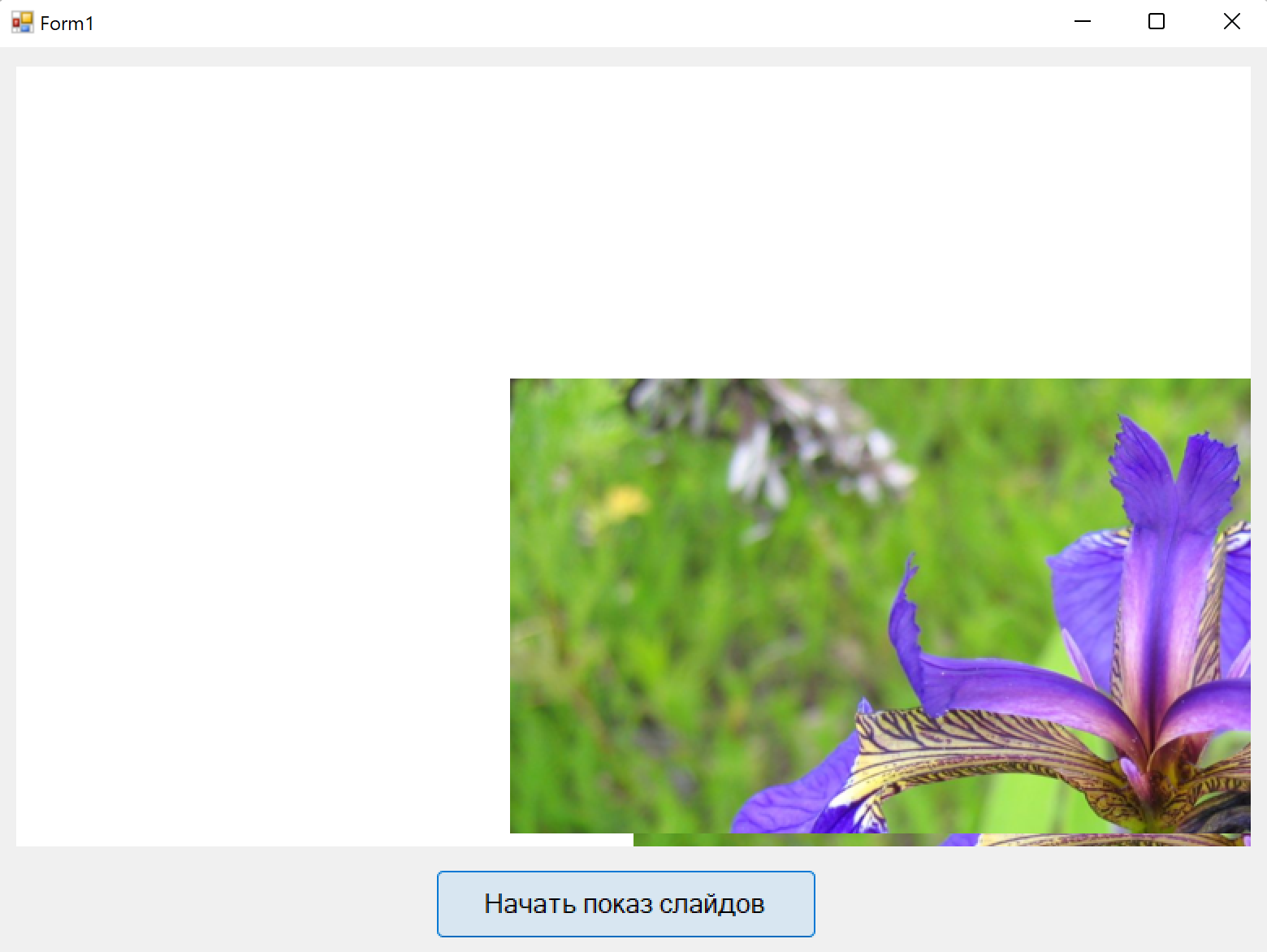
GB.DrawString("Ирис разноцветный", font, Brushes.Orange, 280, 400);

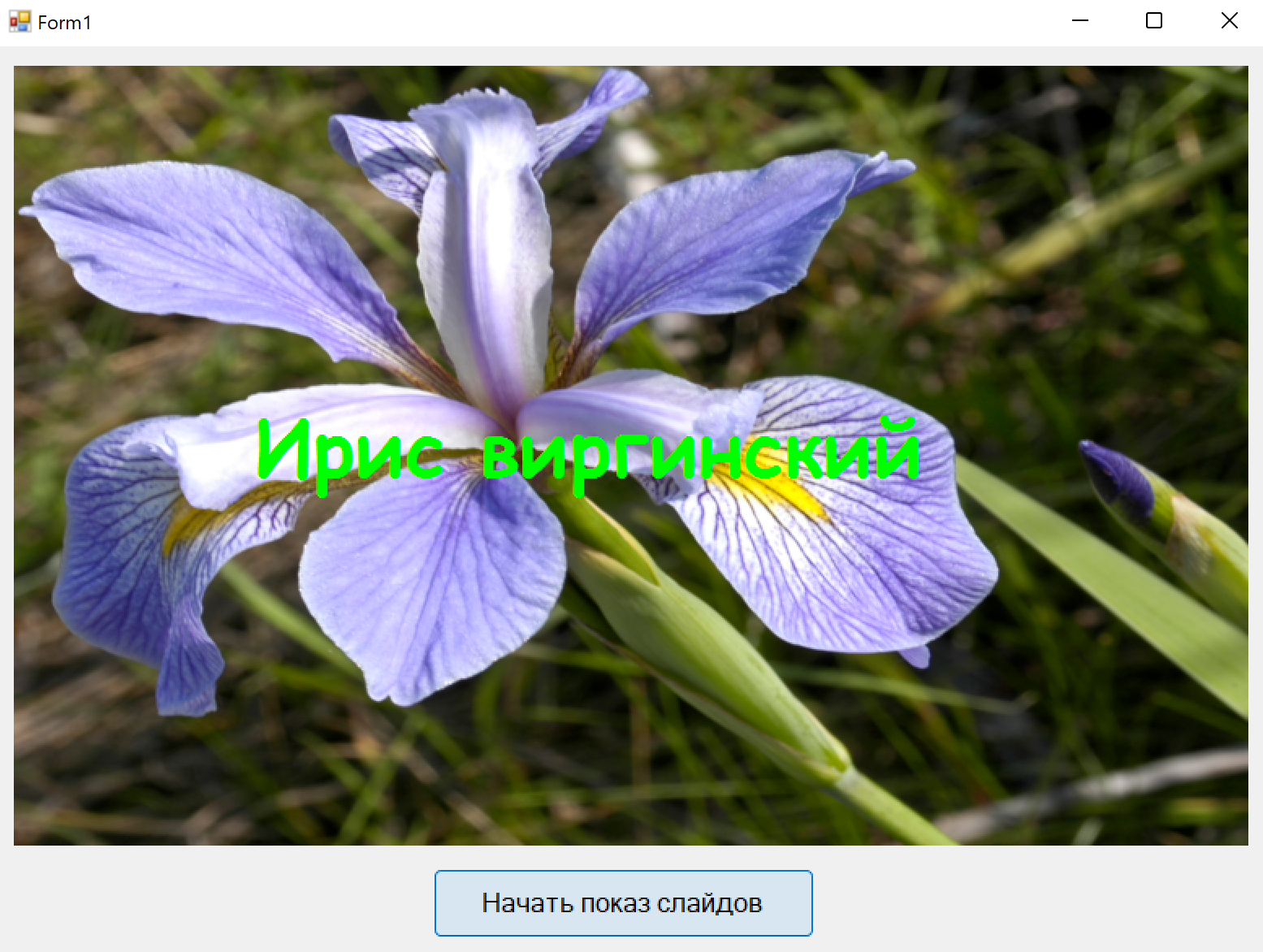
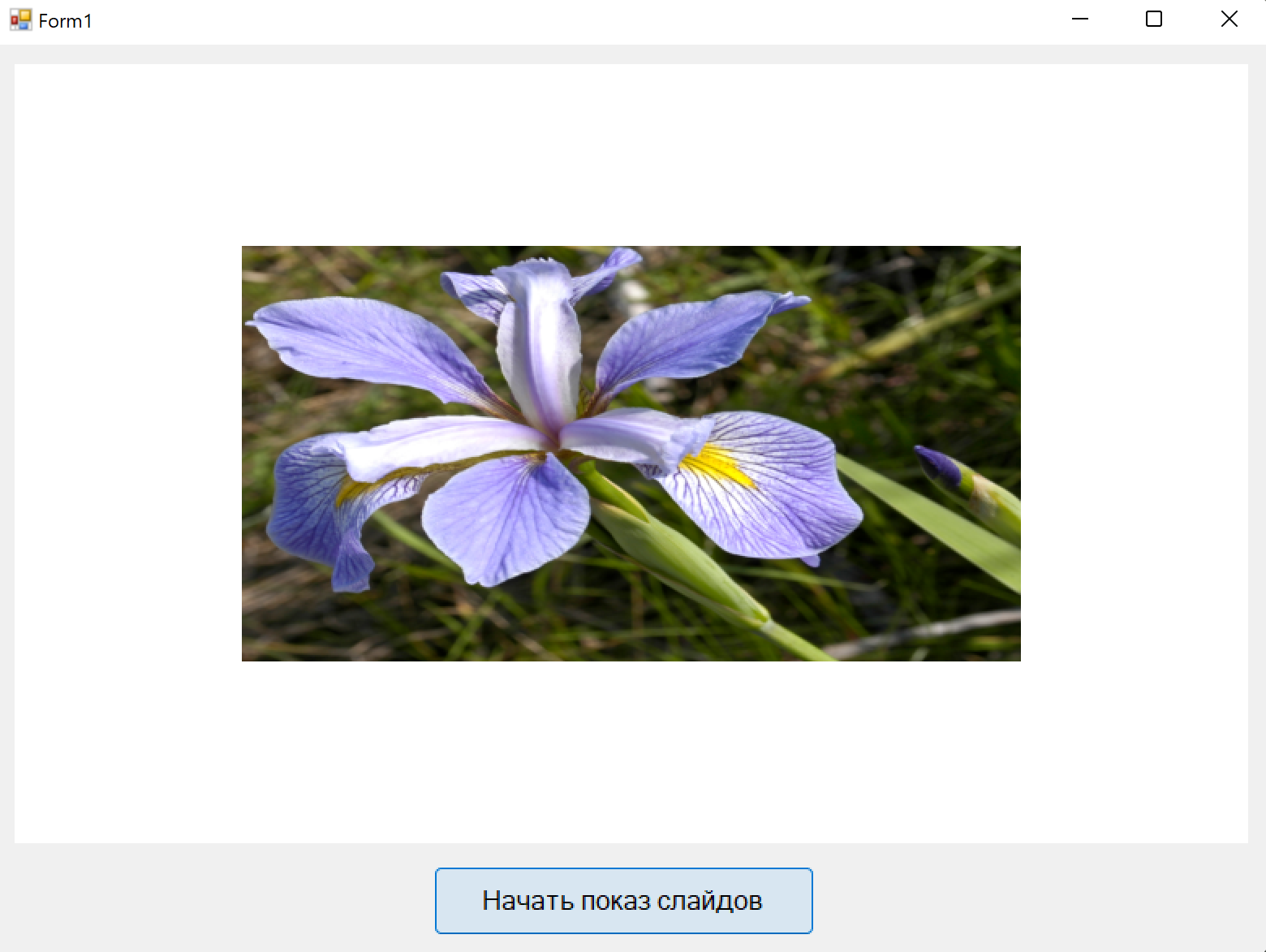
bitmap.Dispose();

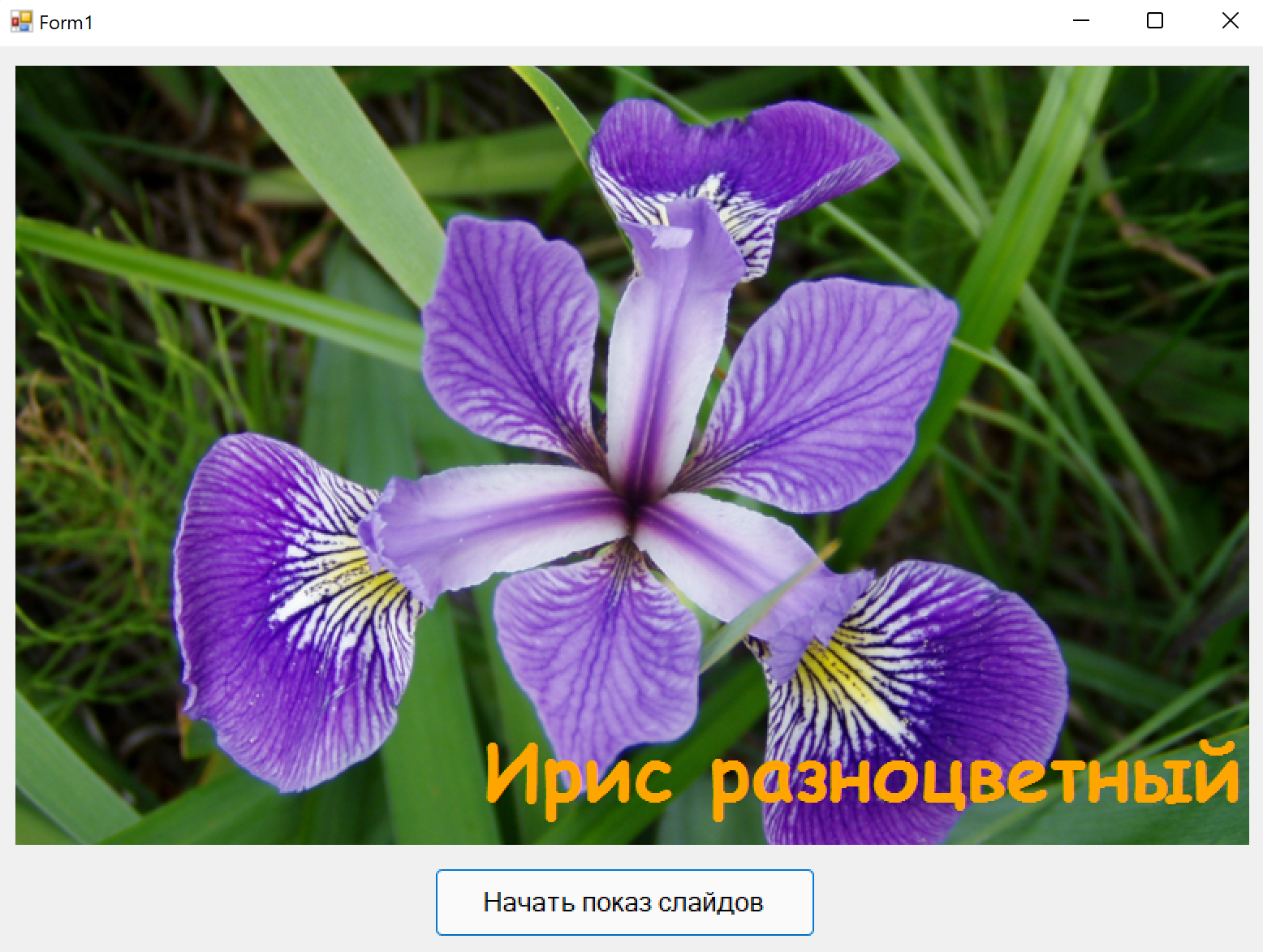
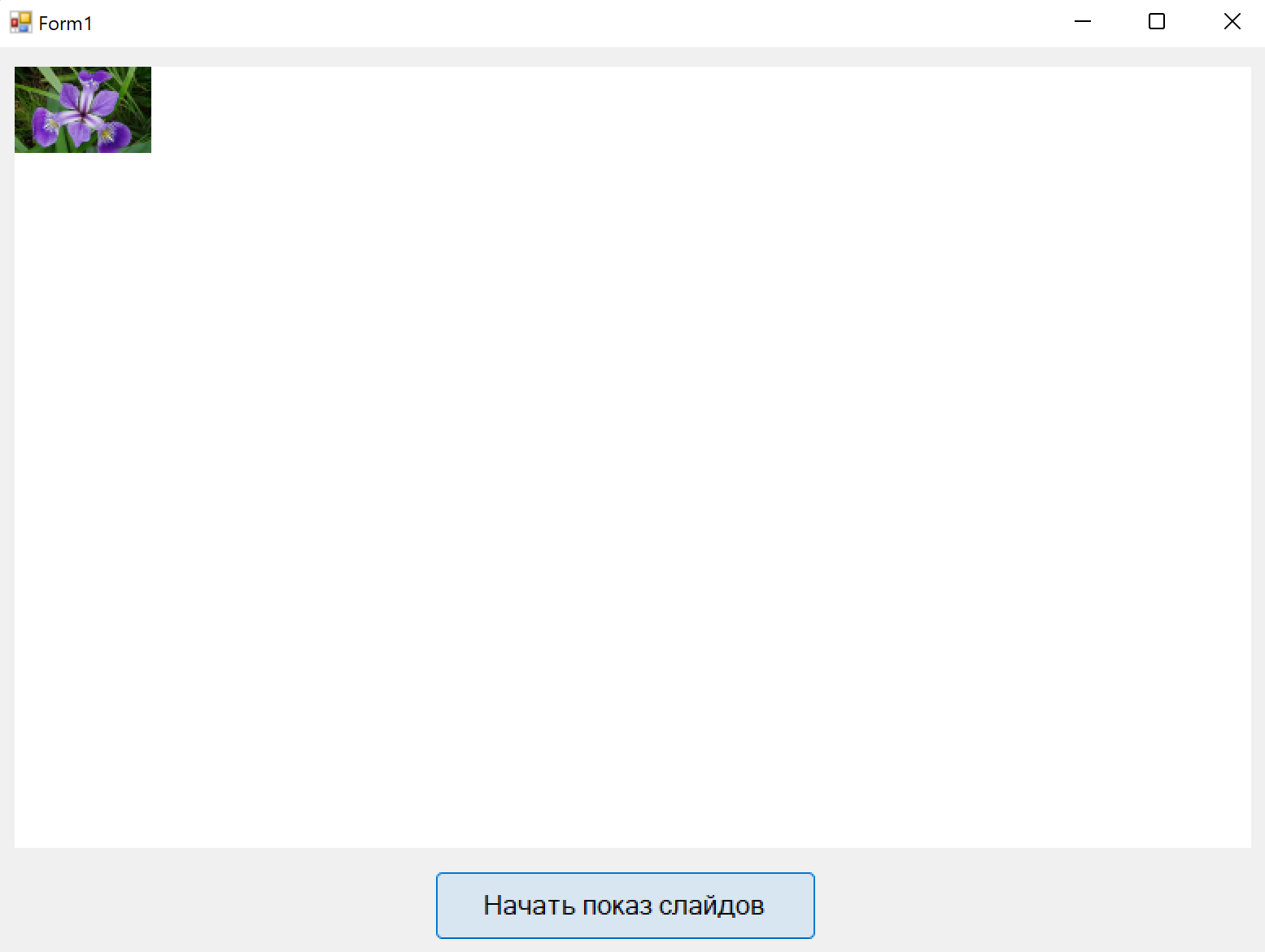
}

Ссылка на демонстрацию: <https://disk.yandex.ru/i/egfUQSmhOHRrNw>

Демонстрация работоспособности:







1. В приложении WinAPI создал метафайл и нарисовал в нем несколько геометрических фигур (сохранил файл). Создал такую же картинку и сохранил в формате .bmp. Сравнил файлы по размеру.

Необходимые переменные:

HWND hMetaBtn; // кнопка создания метафайла

const static int idMetaBtn = 1;

bool is\_meta = false;

Кнопка:

hMetaBtn = CreateWindow(L"BUTTON", L"Создать метафайл", WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON, 210, 0, 200, 25, hWnd, (HMENU)idMetaBtn, hInstance, NULL);

Реакция на нажатие кнопки в WM\_COMMAND:

case idMetaBtn:

{

HDC metaDC = CreateMetaFile(L"metafile.wmf");

if (metaDC != NULL) {

HBRUSH rb = CreateSolidBrush(RGB(255, 0, 0));

HBRUSH gb = CreateSolidBrush(RGB(0, 255, 0));

SelectObject(metaDC, rb);

Rectangle(metaDC, 0, 0, 300, 300);

SelectObject(metaDC, gb);

Ellipse(metaDC, 0, 0, 300, 300);

SelectObject(metaDC, rb);

RoundRect(metaDC, 100, 100, 200, 200, 10, 10);

CloseMetaFile(metaDC);

DeleteDC(metaDC);

}

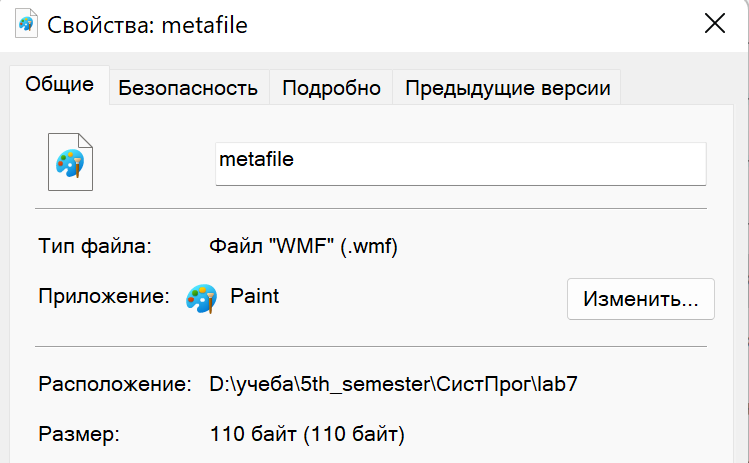
else

MessageBox(hWnd, L"Не удалось создать metafile.wmf ", L"Error", MB\_OK | MB\_ICONEXCLAMATION);

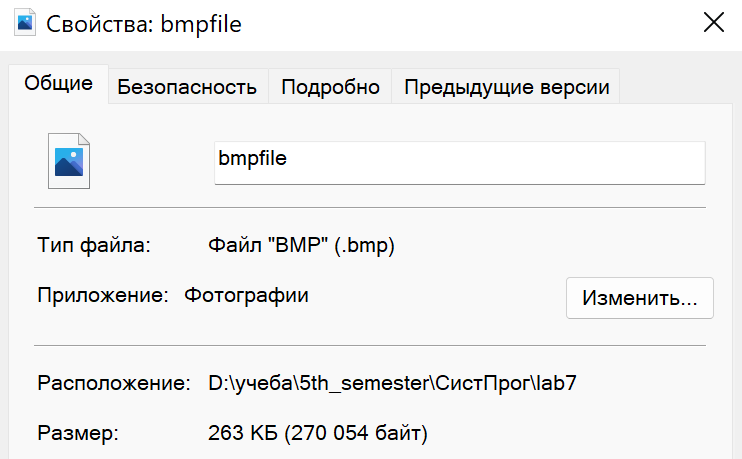
}

break;

После нажатия кнопки создается метафайл, посмотрим на его размер:



Всего 110 байт! А теперь создадим такой же рисунок в формате bmp.



Разница более чем в 2000 раз!

Полный код программы доступен по ссылкам: <https://github.com/programzan/SystemProgramming/blob/lab7/lab7.cpp>   
<https://github.com/programzan/SystemProgramming/blob/lab7/Form1.cs>

Здесь же размещены все сопутствующие файлы: <https://github.com/programzan/SystemProgramming/tree/lab7>