

Urządzenia Peryferyjne

Obsługa skanera płaskiego

Vladyslav Lutsenko, 245817

Termin zajęć Środa 11:00 TP

1.Cel ćwiczenia

Celem wykonywanego ćwiczenia było stworzenie programu, który będzie umożliwiać skanowanie przy pomocy skanera płaskiego. Program musi mieć następujące funkcje:

1. Skanowanie z wykorzystaniem UI.
2. Skanowanie bez wykorzystania UI.
3. Wyświetlenie uzyskanego obrazu.
4. Zmiana rozdzielności skanera.
5. Zmiana trybu skanowania.

Następnym celem było rozszerzyć ten program, żeby umożliwiał następujące funkcje:

1. Obsługa różnych trybów przesyłania danych.
2. Zapis skanowanych obrazów do plików graficznych.

2.Wstęp teoretyczny

Skaner jest wejściowym, zewnętrznym urządzeniem peryferyjnym, który przetwarza obraz rzeczywistego obiektu do postaci cyfrowej, zrozumiałej dla komputera. Skaner nie rejestruje całego obrazu naraz, w przeciwieństwie do aparatu. Skaner odczytuje kolejne pasma informacji, wykorzystując do tego ruchomą głowicę, więc jest on czytnikiem przebiegowym.

3.Realizacja ćwiczenia

W trakcie zajęć udało się wykonać następujące zadania:

1. Napisać program, który umożliwia skanowanie przy pomocy skanera płaskiego. Program ma następujące funkcje:

- a) Skanowanie z wykorzystaniem UI.
- b) Wyświetlenie uzyskanego obrazu.
- c) Zmiana rozdzielności skanera.
- d) Zmiana trybu skanowania.

2. Rozszerzyć ten program o możliwość zapisywania do plików graficznych.

Opis programu

Program został napisany z wykorzystaniem języka C#. Przy napisaniu korzystaliśmy z biblioteki WIA, która jest bardzo prosta do implementacji. Po uruchomieniu programu mamy możliwość zeskanować jakiś dokument za pomocą wybranego przez nas. Oprócz tego jest możliwość konfiguracji uzyskiwanego obrazu ze skanowania, parametrami obrazu, które można zmieniać jest DPI, kontrast, jasność. Możemy też wybrać, czy obraz ma być kolorowy czy w skali szarości. Program umożliwia zapisanie uzyskanego obrazu.

Menu programu:

The screenshot shows a Windows application window titled 'Form1'. The interface includes a 'LoadScanner' button, a dropdown menu, a 'DPI' text box containing '150', radio buttons for 'Color' and 'BlackWhite', sliders for 'Brightness' and 'Contrast', a file path 'F:\Programing\C#\Scanner\bin\Debug\scan', a file type dropdown menu showing '.png', and two buttons labeled 'UI' and 'Scan'.

Otrzymanie spisu dostępnych skanerów oraz ustawianie początkowych parametrów skanowania:

```
private void buttonLoad_Click(object sender, EventArgs e)
{
    labelPath.Text = System.IO.Path.GetDirectoryName(Application.ExecutablePath) + "\\scan";
    try
    {
        for (int i = 1; i <= deviceManager.DeviceInfos.Count; i++)
        {
            if (deviceManager.DeviceInfos[i].Type == WiaDeviceType.ScannerDeviceType)
            {
                comboBoxListOfScanner.Items.Add(deviceManager.DeviceInfos[i].Properties["Name"].get_Value());
            }
        }
        textBox1.Text = "150";
        trackBar1.Value = 5;
        trackBar2.Value = 5;
        comboBoxFile.Items.Add(".jpg");
        comboBoxFile.Items.Add(".png");
        comboBoxFile.Items.Add(".bmp");
    }
    catch (COMException exception)
    {
        MessageBox.Show(exception.Message);
    }
}
```

Skanowanie obrazu: ustalenie parametrów skanowania oraz zachowanie pliku.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        // ustalenie parametrów skanowania
        loadSettings();
        WIA.DeviceInfo deviceInfo = null;
        deviceInfo = deviceManager.DeviceInfos[comboBoxListOfScanner.SelectedIndex + 1];

        var device = deviceInfo.Connect();
        Item scannerItem = device.Items[1];

        AdjustScannerSettings(scannerItem, dpi, 0, 0, 1250, 1700, brightness, contrast, color);

        // zachowanie pliku
        var imageFile = (ImageFile)scannerItem.Transfer(FormatID.wiaFormatJPEG);

        var path = System.IO.Path.GetDirectoryName(Application.ExecutablePath) + "\\scan" + comboBoxFile.SelectedIndex;

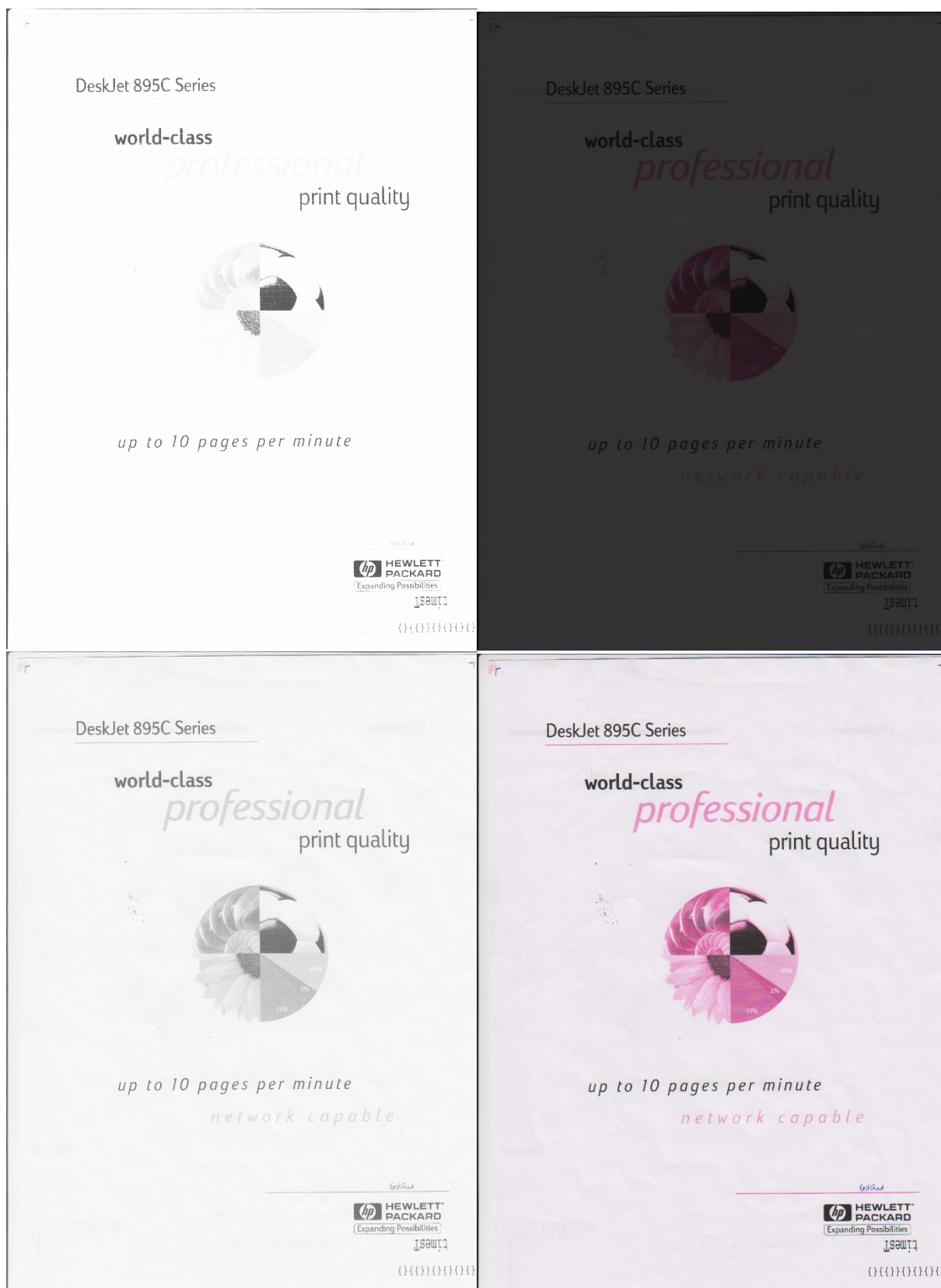
        if (File.Exists(path))
        {
            File.Delete(path);
        }
        imageFile.SaveFile(path);

        pictureBox1.ImageLocation = path;
    }
    catch (COMException exception)
    {
        MessageBox.Show(exception.Message);
    }
}
```

Ustalenie parametrów skanowania

```
private static void AdjustScannerSettings(Item scannerItem, int scanResolutionDPI,
    int scanStartLeftPixel, int scanStartTopPixel, int scanWidthPixels, int scanHeightPixels,
    int brightnessPercents, int contrastPercents, int colorMode)
{
    const string WIA_SCAN_COLOR_MODE = "6146";
    const string WIA_HORIZONTAL_SCAN_RESOLUTION_DPI = "6147";
    const string WIA_VERTICAL_SCAN_RESOLUTION_DPI = "6148";
    const string WIA_HORIZONTAL_SCAN_START_PIXEL = "6149";
    const string WIA_VERTICAL_SCAN_START_PIXEL = "6150";
    const string WIA_HORIZONTAL_SCAN_SIZE_PIXELS = "6151";
    const string WIA_VERTICAL_SCAN_SIZE_PIXELS = "6152";
    const string WIA_SCAN_BRIGHTNESS_PERCENTS = "6154";
    const string WIA_SCAN_CONTRAST_PERCENTS = "6155";
    SetItemIntProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
    SetItemIntProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
    SetItemIntProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_START_PIXEL, scanStartLeftPixel);
    SetItemIntProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_START_PIXEL, scanStartTopPixel);
    SetItemIntProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_SIZE_PIXELS, scanWidthPixels);
    SetItemIntProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_SIZE_PIXELS, scanHeightPixels);
    SetItemIntProperty(scannerItem.Properties, WIA_SCAN_BRIGHTNESS_PERCENTS, brightnessPercents);
    SetItemIntProperty(scannerItem.Properties, WIA_SCAN_CONTRAST_PERCENTS, contrastPercents);
    SetItemIntProperty(scannerItem.Properties, WIA_SCAN_COLOR_MODE, colorMode);
}
```

Efekty skanowania z różnymi konfiguracjami:



3. Wnioski

W trakcie wykonywania ćwiczenia żadnych trudnych problemów nie było. Trochę czasu zajęło napisanie metod, które odpowiadają za konfigurację skanowania. Dużo czasu poszło na poszukiwanie informacji o tym, jak wykonać skanowanie bez wykorzystania UI, ale tak i nie udało się tego wykonać.

Program działa poprawnie i za jego pomocą można dowolnie manipulować obrazem.