Urządzenia Peryferyjne

Laboratorium nr 3 - Skaner

Maciej Bronikowski 248838
Michał Droń 248832

1 Wstęp

1.1 Wstęp teoretyczny

Skaner to urządzenie służące do przebiegowego odczytywania: obrazu, kodu paskowego lub magnetycznego, fal radiowych itp. do formy elektronicznej (najczęściej cyfrowej). Skaner przeszukuje kolejne pasma informacji odczytując je lub rejestrując. Nie jest to więc zwykły czytnik, a czytnik krokowy (np. skaner obrazu nie rejestruje całego obrazu w jednej chwili jak aparat fotograficzny, a zamiast tego rejestruje kolejne linie obrazu - dlatego głowica czytająca skanera przesuwa się lub skanowane medium pod nią). Nazwa skanera jako czytnika przebiegowego często przenoszona jest na czytniki nieprzebiegowe (np. elektroniczne).

Skaner optyczny w komputerach to peryferyjne urządzenie wejściowe umożliwiające przetworzenie statycznego obrazu rzeczywistego obiektu (np. kartka, powierzchnia ziemi, siatkówka ludzkiego oka) do postaci cyfrowej, w celu dalszej obróbki komputerowej. Plik komputerowy powstający w wyniku skanowania jakiegoś obrazu nosi nazwę skan. Skanery optyczne stosuje się w celu przygotowania do obróbki graficznej obrazu (DTP), rozpoznawania pisma, w systemach zabezpieczeń i kontroli dostępu, archiwizacji dokumentów i zbiorów starodruków, badaniach naukowych, medycznych itd.

Skanery optyczne kodów kreskowych są powszechnie stosowane w szeroko pojętej logistyce do szybkiego i automatycznego odczytywania informacji o produktach. Najczęściej spotykane są w kasach sklepowych, magazynach części zamiennych a także w kasach pocztowych (odczytywanie informacji z rachunków) czy w bibliotekach (identyfikacja książek i klientów).

1.2 Cel ćwiczenia

Celem ćwiczenia jest nauka obsługi skanera w aplikacji desktopowej.

2 Implementacja

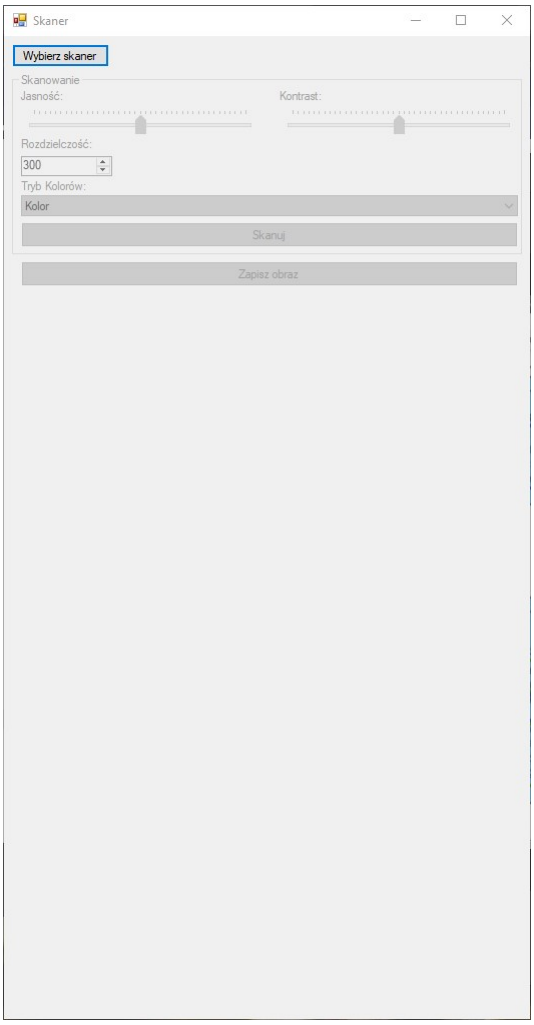
2.1 Działanie aplikacji

W ramach ćwiczeniach wykonano implementację aplikacji desktopowej, która służy do obsługi skanera. Aplikacja posiada następujące funkcje:

- Wybór urządzenia
- Skanowanie z wykorzystaniem UI
- Skanowanie bez wykorzystania UI
- Wyświetlanie uzyskanego obrazu
- Zmiana rozdzielczości skanera

- Zmian jasności i kontrastu skanu
- Zmiana trybu skanowania (1-bitowy, skala szarości, RGB)
- Zapis skanowanych obrazów

Aplikacja została napisana w języku C#. Do komunikacji ze skanerem wykorzystana została biblioteka Interop.WIA. Graficzny interfejs został przygotowany z użyciem interfejsu Windows Forms platformy .NET Framework.



Rysunek 1: GUI aplikacji po włączeniu

2.2 Klasa Program.cs

Jest to klasa główna programu. W metodzie statycznej Main ustawiane są podstawowe parametry aplikacji, oraz uruchamiana jest standardowa pętla aplikacji z wskazaniem na główny formularz klasy Form1.cs, który będzie wyświetlony.

2.3 Klasa Form1.cs

Klasa Windows Forms. Odpowiedzialna jest za GUI aplikacji. Znajdują się w niej wszystkie elementy potrzebne do obsługi formularza aplikacji, takie jak metody zdarzeń wywoływanych podczas manipulacji elementami formularza.

2.3.1 Pola klasy

- Scanner scanner - instancja klasy obsługi skanera
- int selectedColor - wybrany tryb skanowania
- MemoryStream ms - przechowuje zeskanowany obraz
- Decimal dpiInputValue - przechowuje wartość DPI skanowania
- System.Windows.Form.Timer - timer do przechwytywania skanu z eventu skanu z urządzenia
- MemoryStream msFromEvent - przechowuje skan z eventu skanu z urządzenia

2.3.2 Metody klasy

- scannerForm (konstruktor) - tworzy okno aplikacji, inicjalizuje timer
- tick_function - metoda wywoływana raz na 500 ms w czasie działania aplikacji. Sprawdza, czy został przechwycony plik skanu z eventu skanera. Jeżeli tak, pozwala na załadowanie pliku do programu.
- setmsFromEvent - publiczna metoda umożliwiająca klasie skanera ustawienie zeskanowanego pliku z eventu.
- scannerForm_Load - event wywoływany podczas ładowania formularza. Zawiera podstawowe ustawienia elementów GUI.
- connectButton_Click - Obsługa zdarzenia kliknięcia przycisku "Wybierz skaner"
- scanButton_Click - Obsługa zdarzenia kliknięcia przycisku "Skanuj"
- colorComboBox_SelectedIndexChanged - Obsługa zdarzenia zmiany trybu skanowania (1-bitowy, skala szarości, RGB)
- saveButton_Click - Obsługa zdarzenia kliknięcia przycisku "Zapisz obraz"
- dpiInput_ValueChanged - Obsługa zdarzenia zmiany rozdzielczości DPI skanera

2.4 Klasa Scaner.cs

Klasa odpowiedzialna za obsługę skanera za pomocą biblioteki Interop.WIA

2.4.1 Stałe

Zadeklarowane zostały stałe, które przechowują kody wiadomości do komunikowania się ze skanerem.

const string wiaFormatBMP = "{B96B3CAB-0728-11D3-9D7B-0000F81EF32E}" — GUID wskazujące format pliku BMP, który zostanie wysłany przez skaner
wiaEventScanImage = "{A6C5A715-8C6E-11D2-977A-0000F87A926F}" — GUID wskazujące zdarzenie skanu z urządzenia (skanowanie bez użycia UI)
const int WIA_IPS_CUR_INTENT = 6146 — wskazuje na wartość trybu skanowania (np. 1-bitowy, skala szarości, RGB)
const int WIA_IPS_XRES = 6147 — wskazuje na wartość poziomej rozdzielczości (DPI)
const int WIA_IPS_YRES = 6148 — wskazuje na wartość pionowej rozdzielczości (DPI)
const int WIA_IPS_BRIGHTNESS = 6154 — wskazuje na wartość jasności skanu (wartości od -1000 do 1000)
const int WIA_IPS_CONTRAST = 6155 — wskazuje na wartość kontrastu skanu (wartości od -1000 do 1000)
const int WIA_INTENT_IMAGE_TYPE_COLOR = 0x00000001 — Wskazuje na jeden z trybów skanowania (kolor)

const int WIA_INTENT_IMAGE_TYPE_GRAYSCALE = 0x00000002 — Wskazuje na jeden z trybów skanowania (Odcienie szarości)
const int WIA_INTENT_IMAGE_TYPE_TEXT = 0x00000004 — Wskazuje na jeden z trybów skanowania (1 - bitowy)

2.4.2 Pola

WIA.DeviceManager manager — instancja menadżera urządzeń z biblioteki Interop.WIA
WIA.Device scanner — instancja skanera
scannerForm form — instancja formularza Windows Forms - używana podczas skanu z eventu

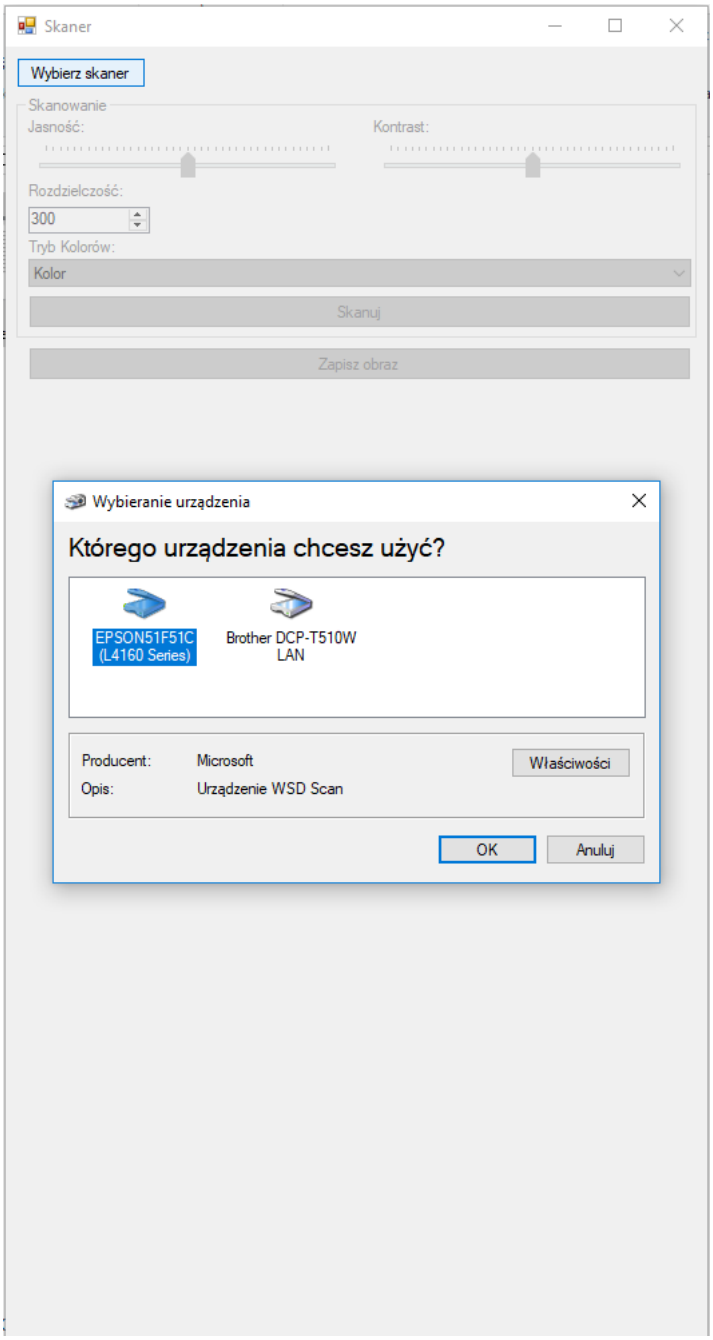
2.4.3 Metody

Scanner (konstruktor) — tworzy instancję menadżera urządzeń oraz eventu skanowania
dm_OnEvent — metoda zajmująca się obsługą zdarzenia skanu z urządzenia (z pominięciem UI). Przechwytuje i zapisuje plik skanu do odpowiedniej właściwości obiektu scannerForm.
isScannerSelected — sprawdza, czy zostało wybrane urządzenie
selectScanner — odpowiada za wyświetlanie okna dialogowego z wyborem urządzenia, lub automatycznie łączy z skanerem, jeżeli tylko jeden jest dostępny
scan — metoda odpowiada za sam proces skanowania. Parametry, które należy dostarczyć to tryb skanowania (kolor, skala szarości, 1-bitowy), rozdzielczość, jasność oraz kontrast. Zwracany jest zeskanowany obraz w formie obiektu MemoryStream.
checkDPI — metoda sprawdza, czy podłączony skaner obsługuje DPI, które wprowadzone zostało w aplikacji
setWIAProperty — metoda umożliwiająca zmienianie ustawień urządzenia (np. kontrastu skanowania). W parametrach należy podać obiekt właściwości urządzenia, id własności zmienianej (string), oraz wartość, na którą należy zmienić właściwość

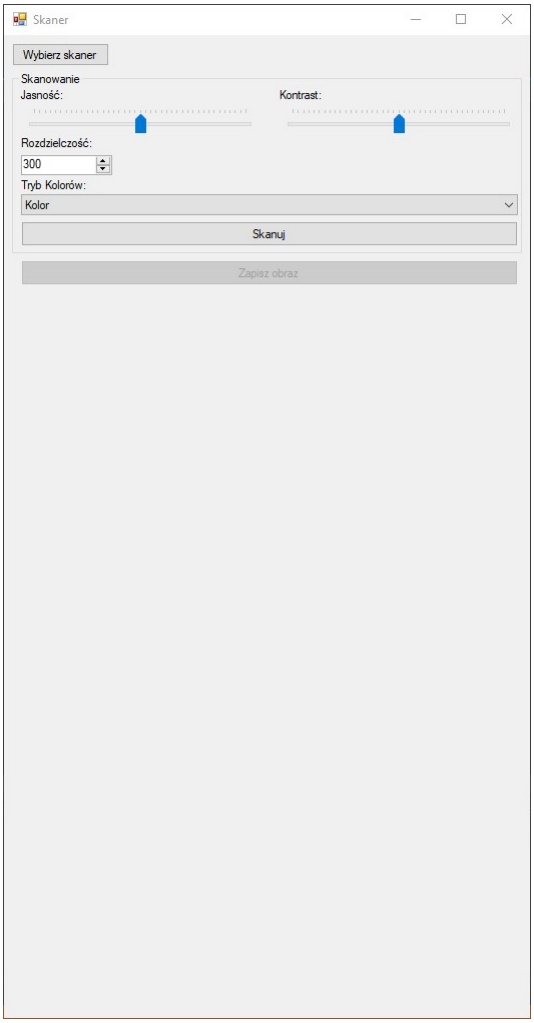
3 Realizacja zadań

Czarny pasek który pojawia się po prawej stronie skanowanych obrazów wynika z uszkodzonej głowicy skanera. Mimo wszystko, skanowanie przebiega poprawnie.

3.1 Wybór urządzenia

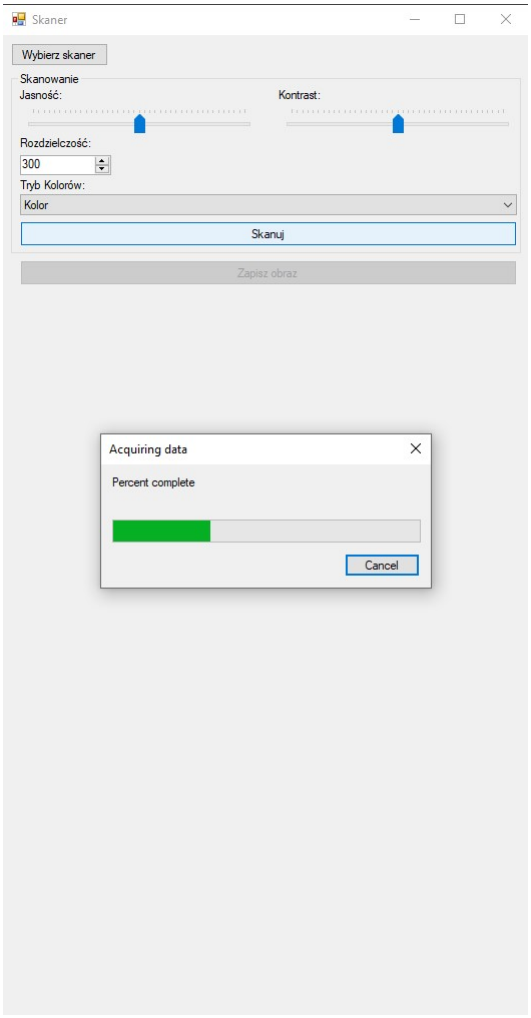


Rysunek 2: Jeśli podłączony jest więcej niż jeden skaner, wyświetla się komunikat o wyborze urządzenia.



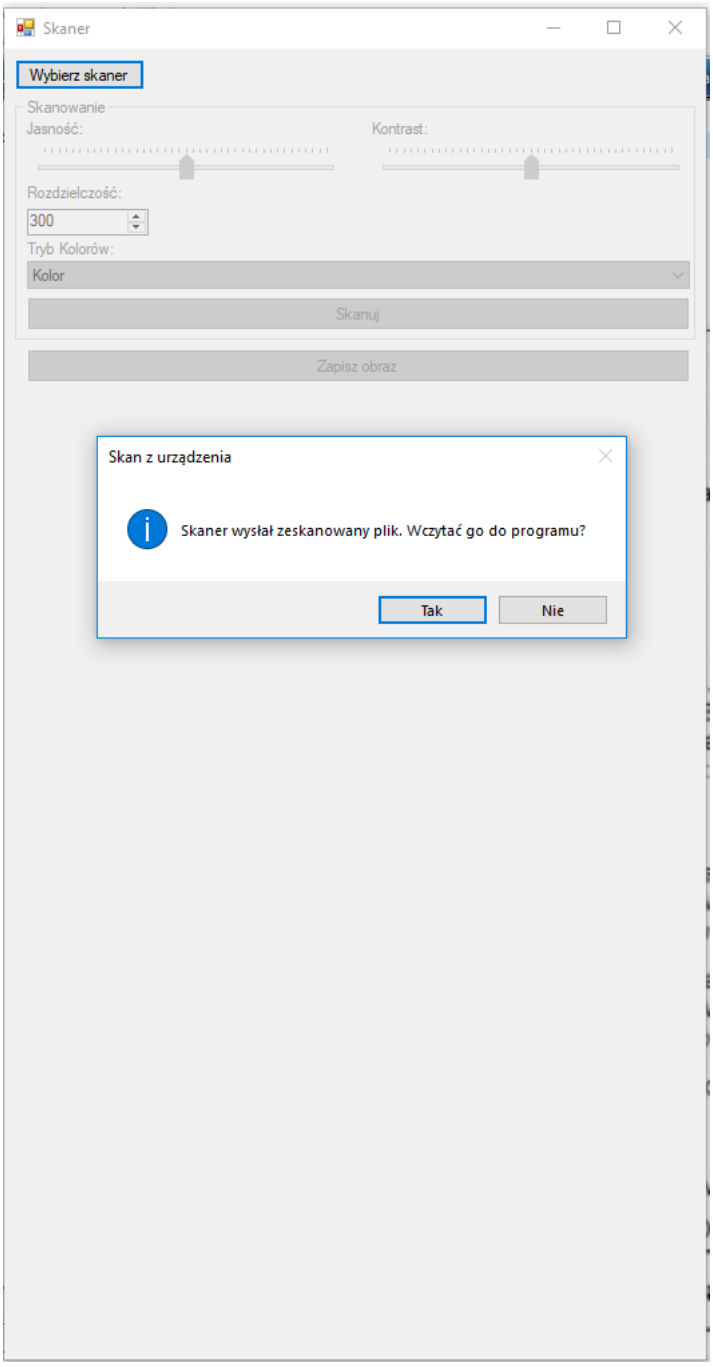
Rysunek 3: Jeśli podłączony jest tylko jeden skaner, automatycznie dochodzi do połączenia z nim i możliwe jest rozpoczęcie skanowania lub zmiana ustawień.

3.2 Skanowanie z wykorzystaniem UI



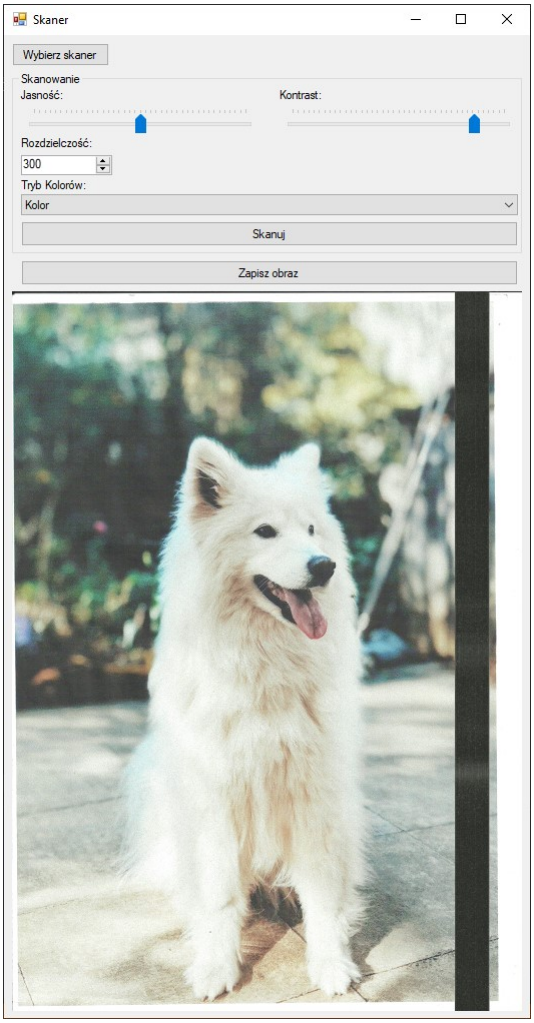
Rysunek 4: Po naciśnięciu "Skanuj" dochodzi do skanowania

3.3 Skanowanie bez wykorzystania UI



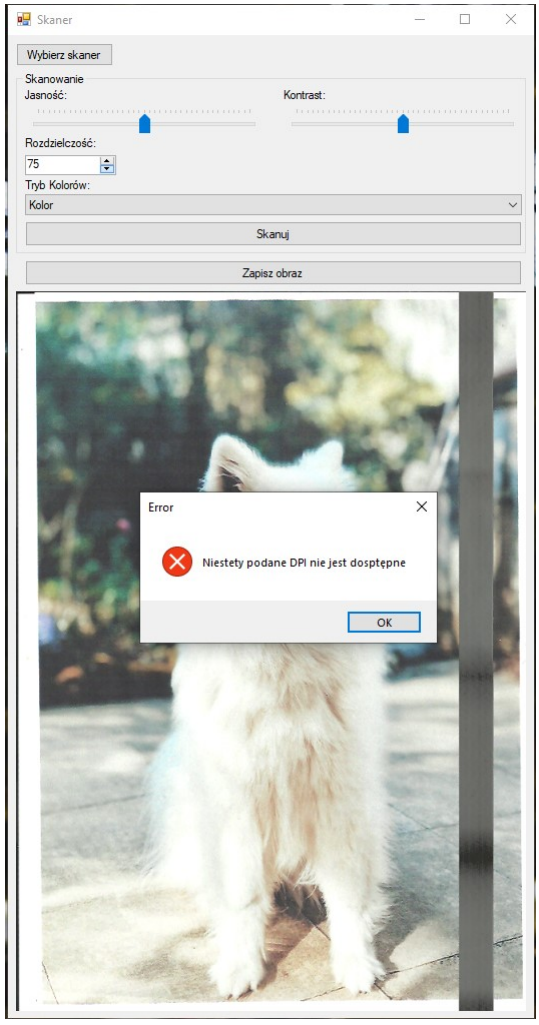
Rysunek 5: Po naciśnięciu przycisku skanowania na skanerze i zakończeniu tego procesu, w aplikacji wyświetla się komunikat czy wczytać zdjęcie.

3.4 Wyświetlanie uzyskanego obrazu



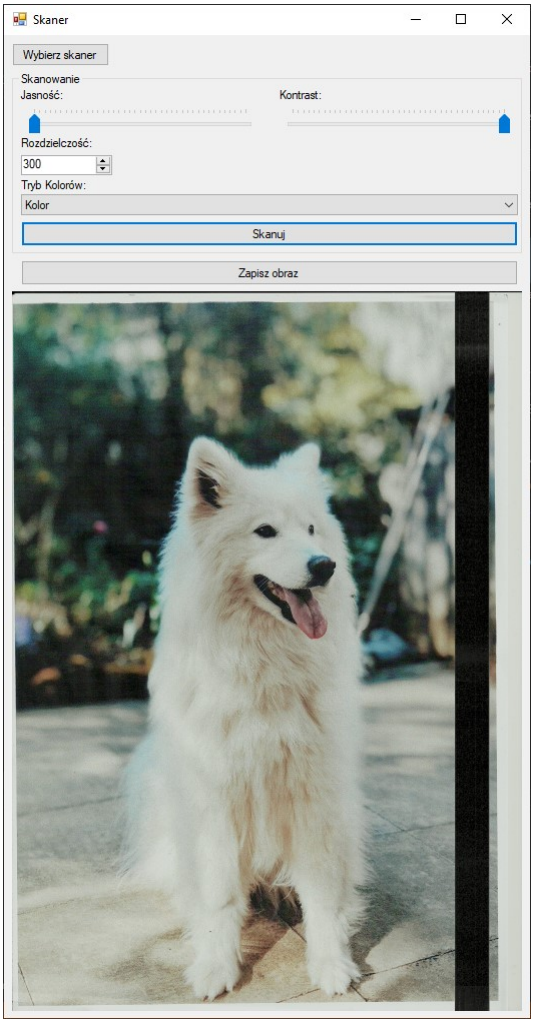
Rysunek 6: Po zakończeniu skanowania obraz wyświetlany jest w oknie programu.

3.5 Zmiana rozdzielczości skanowanego obrazu



Rysunek 7: Możliwa jest zmiana rozdzielczości (DPI) skanowanego obrazu. Aplikacja dodatkowo sprawdza, czy podane DPI jest obsługiwane przez skaner.

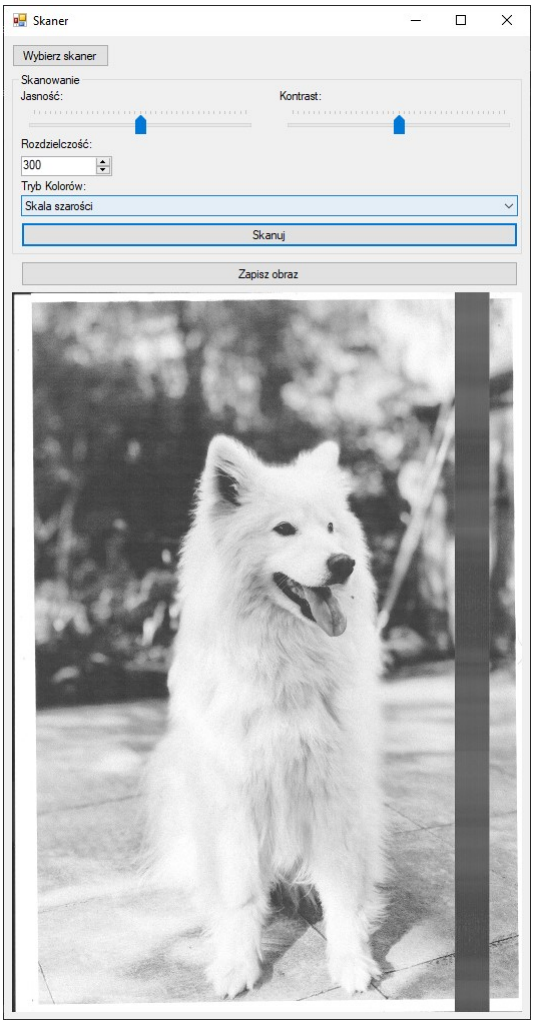
3.6 Zmian jasności i kontrastu skanu



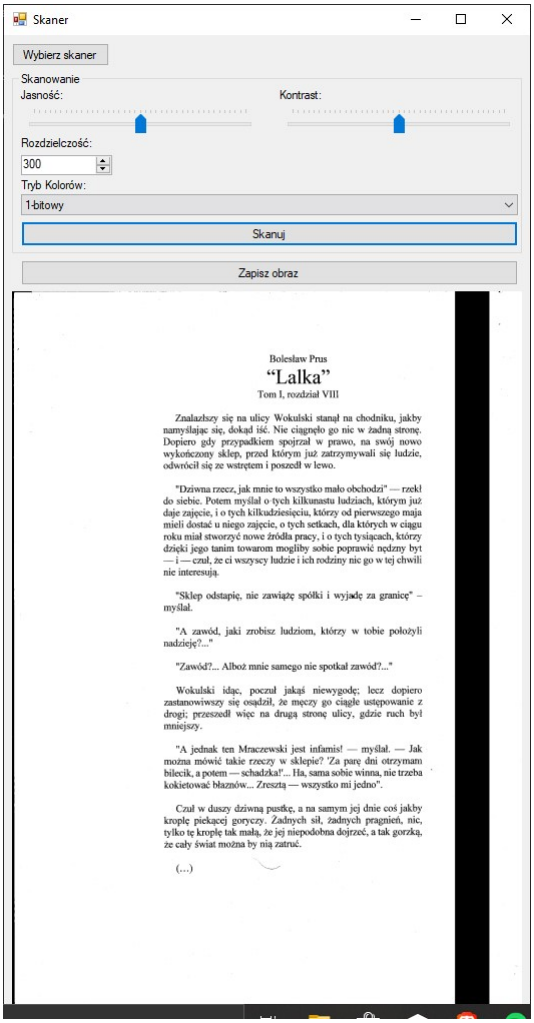
Rysunek 8: Możliwa jest zmiana jasności i kontrastu skanowanego obrazu. (dla porównania patrz rysunek 6.)

3.7 Zmian trybu skanowania

Oprócz trybu skanowania kolorowego, możliwe jest skanowanie w skali szarości oraz w trybie tekstowym (1-bitowym).

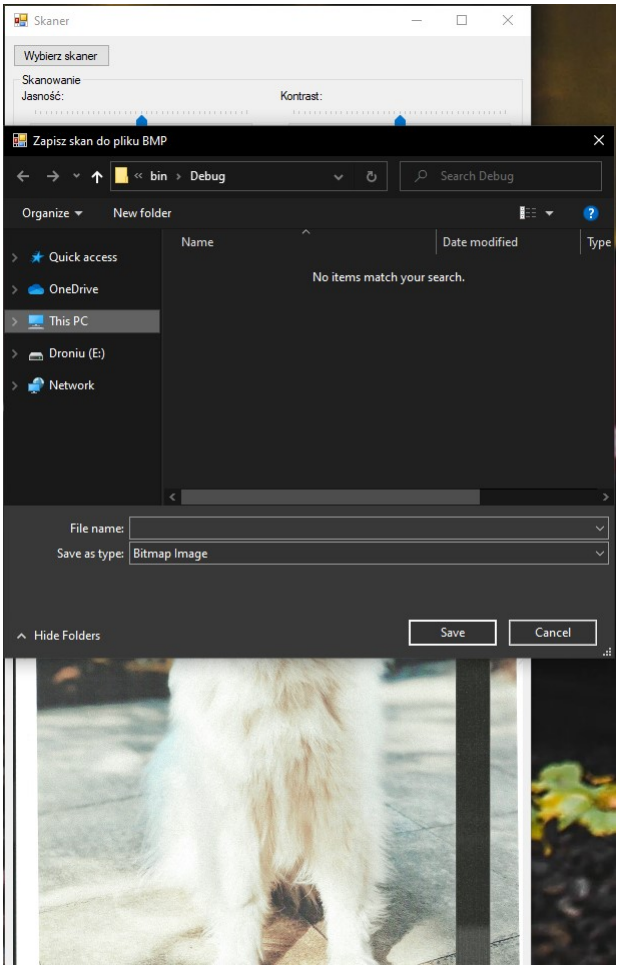


Rysunek 9: Zeskanowany obraz w skali szarości



Rysunek 10: Zeskanowany tekst w trybie 1-bitowym

3.8 Zapisywanie zeskanowanego obrazu



Rysunek 11: Po kliknięciu "Zapisz obraz" mamy możliwość zapisania skanu do pliku BMP.

4 Podsumowanie

W wyniku realizacji ćwiczeń laboratoryjnych stworzona została aplikacja desktopowa do obsługi skanera. Posiada ona stosunkowo szeroką funkcjonalność. Podczas tworzenia aplikacji nie zostały napotkane żadne problemy, które uniemożliwiłyby realizację ćwiczeń.

Pełny kod aplikacji znajduje się w serwisie GitHub:
`https://github.com/JeLLek1/UrzadzeniaPeryferyjne` w folderze Lab_9