

Projektowanie Oprogramowania Systemów

Opracowanie wymagań, modele UML

1. Zidentyfikowanie interesariuszy

Rola	Opis
Klient (Zlecający projekt)	dr hab. inż. Marek Wójcikowski
Użytkownik końcowy	Użytkownik korzystający z programu z poziomu terminala.
Zespół projektowy	Artur Radko 188805 Marek Omilian 188759 Jan Reza 188866 Łukasz Saldat 188714

2. Wymagania funkcjonalne i нефunkcjonalne

Funkcjonalne

- Program uruchamiany z linii poleceń.
- Program wczytuje plik konfiguracyjny .ini ze ścieżką źródłową i docelową.
- Program wczytuje wszystkie obrazy z folderu źródłowego.
- Przetwarzanie obrazów (detekcja konturów).
- Zapisywanie wynikowych obrazów w folderze docelowym.
- Stworzenie matrycy miniatur oryginalnych i przetworzonych obrazów.
- Obsługa formatów JPG/PNG.

Pozafunkcjonalne

- Wydajność: wielowątkowe przetwarzanie.
- Skalowalność: możliwość łatwego dodania kolejnych funkcji (np. rozmycie, detekcja kolorów).
- Niezawodność: program nie może przerwać działania przy błędnym pliku.
- Zgodność: działa na Windows
- Czytelność kodu i dokumentacji.

3. Scenariusze użycia

UC1: Przetwarzanie obrazów

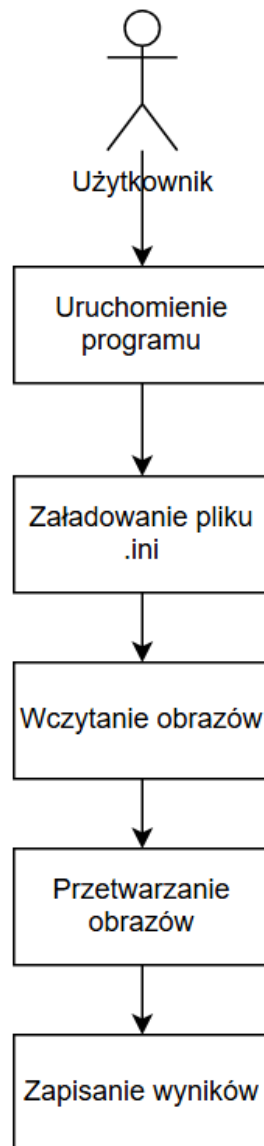
Aktor: Użytkownik

Opis: Użytkownik uruchamia program z linii poleceń. Program przetwarza obrazy i zapisuje wyniki.

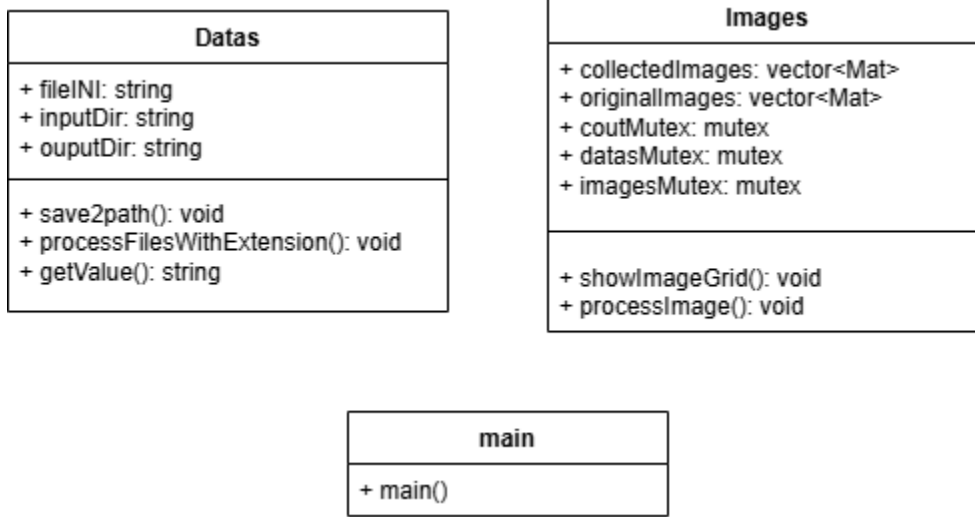
Scenariusz główny:

1. Użytkownik uruchamia program z parametrem config.ini.
2. Program odczytuje ścieżki wejściową i wyjściową.
3. Program łączy wszystkie obrazy z katalogu wejściowego.
4. Detekcja konturów wykonywana jest w wielu wątkach.
5. Obrazy wynikowe zapisywane są w katalogu docelowym.
6. Tworzone są dwie matryce obrazów – siatki miniatur obrazów oryginalnych i wynikowych.

4. UML: Diagram przypadków użycia



5. UML: Diagram klas (podstawowy model)



6. Technologie i biblioteki

Obszar	Technologia
Język	C++23
Przetwarzanie obrazów	OpenCV (cv2)
Przetwarzanie ini	fstream
Wielowątkowość	thread
Obsługa plików	png, jpg

7. Harmonogram pracy

Etap	Opis	Termin
1	Zebranie wymagań, projekt UML	15.05
2	Stworzenie prototypu (config.ini, wczytywanie obrazów)	22.05
3	Implementacja detekcji konturów + wielowątkowość	29.05
4	Zapis wyników i miniatur	01.06
5	Testowanie	07.06
6	Dokumentacja końcowa	08.06
7	Oddanie projektu	12.06

8. Kamienie milowe

1. Poprawny odczyt z pliku INI
2. Wczytywanie i przetwarzanie obrazów
3. Detekcja konturów wielowątkowo
4. Zapis do folderu wyjściowego
5. Generacja miniatur
6. Gotowa dokumentacja i diagramy UML

9. Narzędzia wspierające kontrolę wersji

GitHub - [GitHub](#)

Bugzilla - [Bugzilla](#)