

<p align="center"><b>Programowanie w języku Java - Projekt</b></p> <p align="center">Wydział Elektrotechniki Automatyki i Informatyki</p> <p align="center">Politechnika Świętokrzyska</p>	
Studia: <b>Stacjonarne I stopnia</b>	Kierunek: <b>Informatyka</b>
Grupa: <b>2ID14A</b>	1. Daniel Legawiec 2. Piotr Stępień
<p align="center">Temat projektu:</p> <p align="center"><b>Przeglądarka zdjęć</b></p>	

## 1.Opis projektu

Celem naszego projektu było stworzenie przeglądarki zdjęć. Program ma za zadanie wyświetlenie obrazu w formacie .png .jpg .jpeg , który użytkownik może znaleźć po naciśnięciu na przycisk search, a następnie go otworzyć. Sam kod został napisany z wykorzystaniem Java Swing. W przeglądarce można obracać, powiększać, zmniejszać rozmiar zdjęcia oraz przewijać między obrazami które się znajdują w tym samym pliku. Program można jedynie uruchomić poprzez skompilowanie kodu w środowisku programistycznym np. IntelliJ

## 2.Implementacja

Program został napisany w języku Java, który został stworzony jako projekt Maven z wykorzystaniem środowiska programistycznego IntelliJ. Projekt został podzielony na siedem klas znajdujących się w jednym pakiecie takich jak:

- **Main.java** - odpowiada za wywołanie metody, która rozpoczyna program.
- **Program.java** - zawiera klasy JFrame nazwaną ramka odpowiada za stworzenie głównego okna programu, Container nazwaną containerGlownt oraz dwie metody:

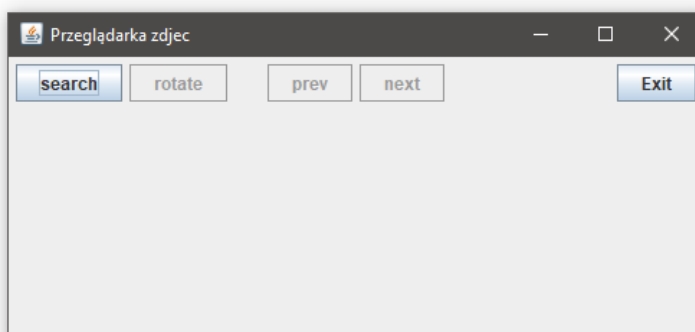
- **addFrame** jej zadaniem jest dodanie opisu do okna, ustawienie rozmiaru, dodanie przycisków do okna, dodanie ramki która wyświetla obraz oraz dodanie słuchaczy zdarzeń do przycisków

-**actionPerformed** - wykonuje odpowiednie działania po zarejestrowaniu zdarzenia od konkretnych przycisków czyli jeśli naciśniemy przycisk search pozwoli nam to do przejścia do nowego okna, aby wyszukać zdjęcie

- **Rotate.java** - klasa odpowiada za obracanie obrazu, działa na zasadzie przerysowania pikseli, obrócony obraz jest zwracany jako obiekt typu BufferedImage
- **Buttons.java** - jej zadaniem jest wyświetlenie przycisków, funkcja addButtons tworzy nowy JPanel, a w nim dodatkowe 3 JPanele aby przyciski były oddzielone od siebie ustawia odpowiedni layout
- **LoadImages.java** - zawiera takie metody jak:
  - **LoadImages** - tworzy okno w którym ma być wyświetlony obraz
  - chooseImage**- zwraca ścieżkę do obrazu wybranego w oknie JFileChooser (pokazuje tylko pliki z rozszerzeniem .png .jpeg .jpg)
  - loadImage** - uruchamia okno FileChooser, wczytuje obraz z podanej ścieżki do obiektu typu BufferedImage, za rysowanie obrazu jest odpowiedzialna klasa wewnętrzna obiektu ImagePanel jest to nadpisana metoda paintComponent, która dopasowuje wielkość obrazu do rozmiarów ramki
  - loadNextImage** - obsługa przycisku next, wczytuje kolejny obraz do imagePanel z tablicy obrazów
  - loadPrevImage** - obsługa przycisku prev, wczytuje poprzedni obraz do imagePanel z tablicy obrazów
  - searchForOtherImages** - szuka innych obrazów w katalogu z wybranym obrazem
- **Zoom.java** - klasa jest odpowiedzialna głównie za przybliżanie i oddalanie obrazu jest rozszerzeniem klasy JPanel, w której jest umieszczony powiększony obraz

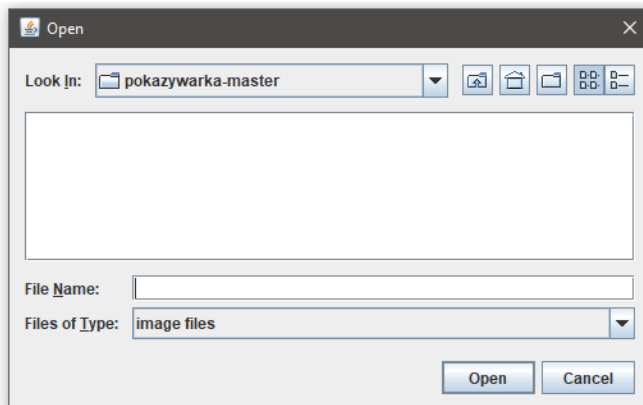
## 1)Menu główne

W menu głównym użytkownik może wyszukać zdjęcie po naciśnięciu przycisku search lub wyjść naciskając przycisk Exit:



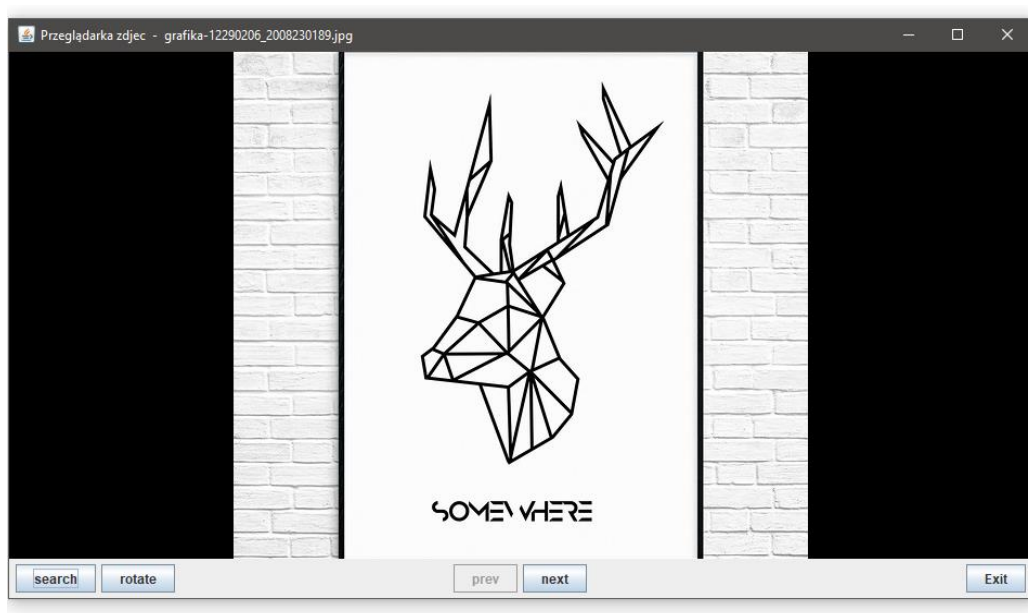
## 2) Okno wyszukiwania obrazu

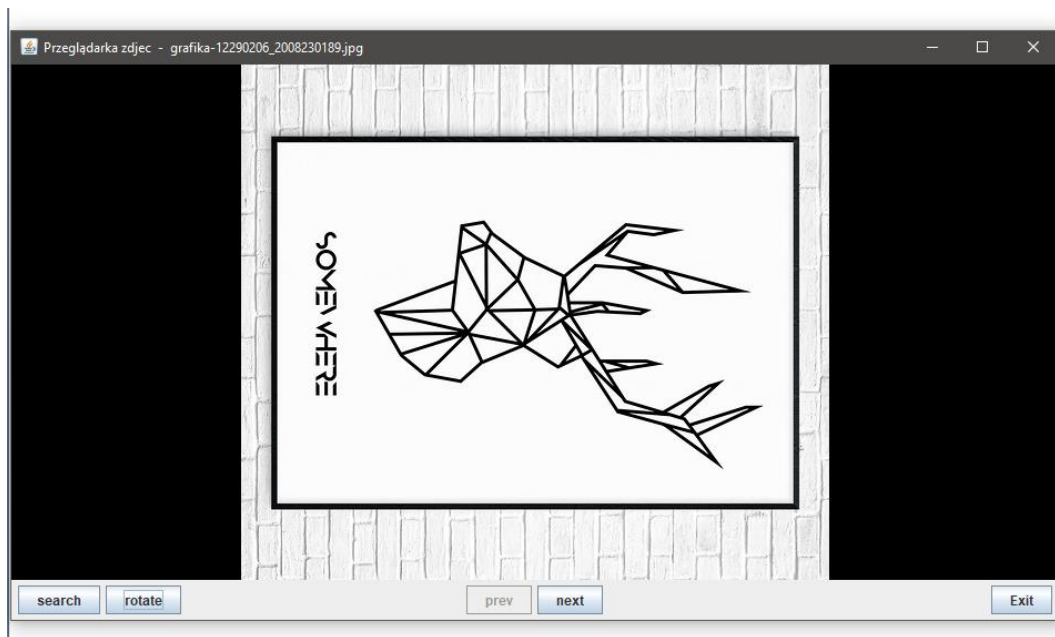
Tutaj użytkownik może wyszukać obraz poprzez wpisanie jego nazwy lub też przeglądanie katalogów w zakładce Look in. W zakładce Files of Type można wybrać czy użytkownik chce aby były wyświetlane pliki z rozszerzeniem .png .jpg .jpeg lub też wszystkie pliki wraz z katalogami.



## 3) Okno wyświetlenia obrazu

W tym oknie wyświetlany jest wybrany obraz, który może być obrócony lub też przybliżony lub oddalony





### 3.Podsumowanie

Program spełnia swoje podstawowe założenia, czyli wyświetla obraz, który można obrócić, przybliżyć jak i oddalić. Praca nad projektem była wspólna, czyli każdy z zespołu wykonał po połowie projekt.