
	Politechnika Bydgoska im. J. J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki Zakład Systemów Teleinformatycznych		
Przedmiot	Przetwarzanie obrazów		
Prowadzący	mgr inż. Grzegorz Czczot		
Temat	Operacje geometryczne		
Student	Paweł Jońca		
Nr lab.	2	Data wykonania	19.10.2024r
Ocena		Data oddania spr.	19.10.2024r

Zad1.

```

import org.opencv.core.Core; // Importowanie klasy Core z biblioteki OpenCV, która zawiera podstawowe funkcje
import org.opencv.core.CvType; // Importowanie klasy CvType do określenia typu danych dla macierzy
import org.opencv.core.Mat; // Importowanie klasy Mat, która reprezentuje macierz obrazu
import org.opencv.core.Size; // Importowanie klasy Size do określania rozmiaru obrazu
import org.opencv.imgcodecs.Imgcodecs; // Importowanie klasy Imgcodecs do wczytywania i zapisywania obrazów
import org.opencv.highgui.HighGui; // Importowanie klasy HighGui do wyświetlania obrazów
import org.opencv.imgproc.Imgproc; // Importowanie klasy Imgproc do przetwarzania obrazów, w tym przekształceń

> public class Main {
>     public static void main(String[] args) {
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie natywnej biblioteki OpenCV
        Mat image = Imgcodecs.imread("dog.jpg"); // Wczytanie obrazu z pliku "dog.jpg" do obiektu typu Mat (macierzy)

        // Sprawdzenie, czy obraz został wczytany prawidłowo
        if (image.empty()) {
            System.out.println("Fail, try again");
            return; // Zakończenie programu, jeśli obraz nie został wczytany
        }

        // Utworzenie macierzy transformacji (przekształcenia affine) o rozmiarze 2x3
        // Mat.eye(2, 3, CvType.CV_32F) tworzy macierz jednostkową (identity matrix) o typie danych float
        Mat translationMatrix = Mat.eye(2, 3, CvType.CV_32F);

        // Ustawienie przesunięcia o 50 pikseli w prawo
        translationMatrix.put(0, 2, 150); // Pierwszy wiersz, trzecia kolumna

        // Ustawienie przesunięcia o 30 pikseli w dół
        // translationMatrix.put(1, 2, 30); // Drugi wiersz, trzecia kolumna

        // Utworzenie nowego obiektu Mat do przechowywania przesuniętego obrazu
        Mat shiftedImage = new Mat();

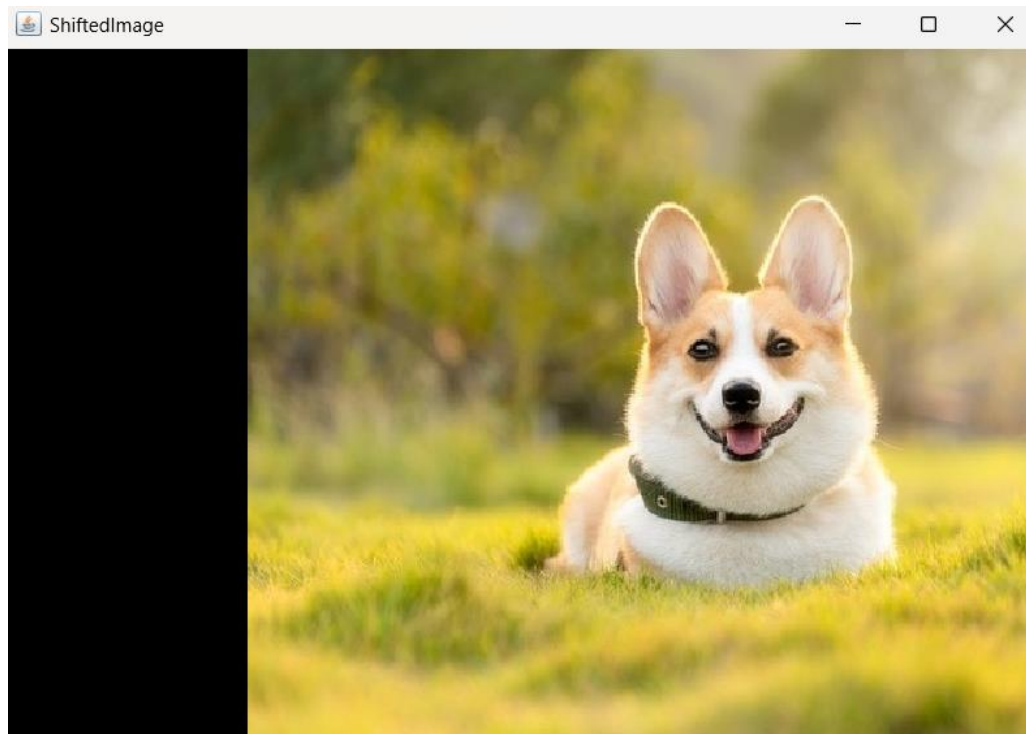
```

```
// // Ustawienie przesunięcia o 30 pikseli w dół
// translationMatrix.put(1, 2, 30); // Drugi wiersz, trzecia kolumna

// Utworzenie nowego obiektu Mat do przechowywania przesuniętego obrazu
Mat shiftedImage = new Mat();

// Zastosowanie przekształcenia affine (przesunięcia) na oryginalnym obrazie
// image: wejściowy obraz
// shiftedImage: wynikowy obraz po przekształceniu
// translationMatrix: macierz przekształcenia określająca przesunięcie
// image.size(): rozmiar obrazu wynikowego
Imgproc.warpAffine(image, shiftedImage, translationMatrix, image.size());

HighGui.imshow( winname: "ShiftedImage", shiftedImage); // Wyświetlenie
HighGui.waitKey();
HighGui.destroyAllWindows();
}
}
```



Tło pikseli jest czarne w miejscu gdzie nie ma obrazu.

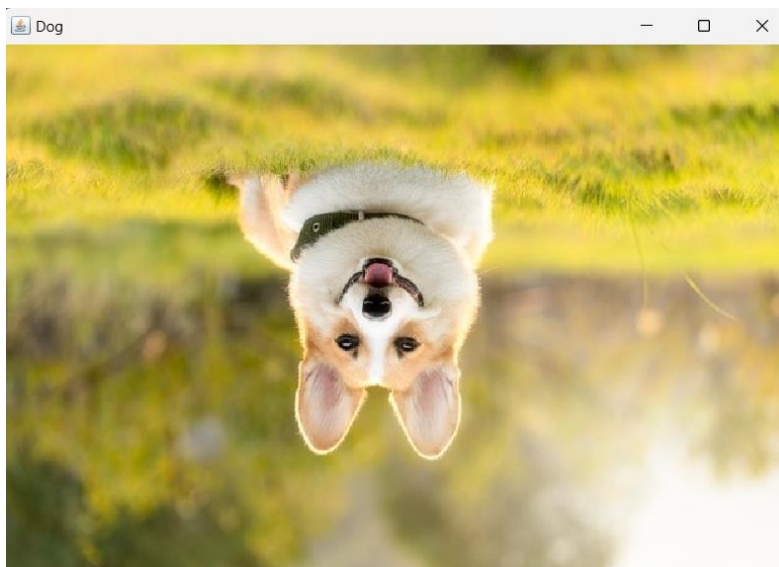
Zad2.

```

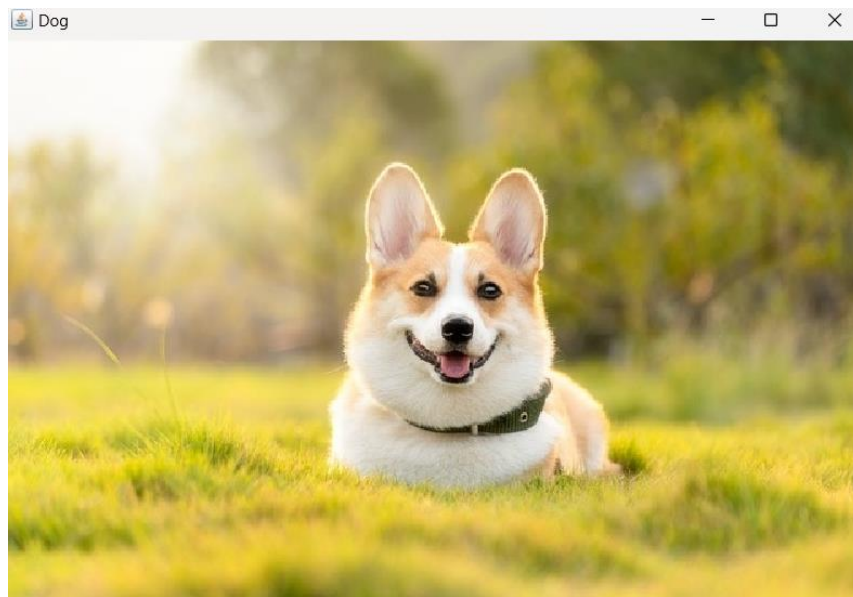
1  import org.opencv.core.Core;           // Importowanie klasy Core z biblioteki OpenCV, która zawiera podstawowe funkcje
2  import org.opencv.core.CvType;         // Importowanie klasy CvType do określenia typu danych dla macierzy
3  import org.opencv.core.Mat;             // Importowanie klasy Mat, która reprezentuje macierz obrazu
4  import org.opencv.core.Size;           // Importowanie klasy Size do określania rozmiaru obrazu
5  import org.opencv.imgcodecs.Imgcodecs; // Importowanie klasy Imgcodecs do wczytywania i zapisywania obrazów
6  import org.opencv.highgui.HighGui;     // Importowanie klasy HighGui do wyświetlania obrazów
7  import org.opencv.imgproc.Imgproc;     // Importowanie klasy Imgproc do przetwarzania obrazów, w tym przekształceń
8
9  public class Main {
10     public static void main(String[] args) {
11         System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie natywnej biblioteki OpenCV
12         Mat image = Imgcodecs.imread("dog.jpg"); // Wczytanie obrazu z pliku "dog.jpg" do obiektu typu Mat (macierzy)
13
14         // Sprawdzenie, czy obraz został wczytany prawidłowo
15         if (image.empty()) {
16             System.out.println("Fail, try again");
17             return; // Zakończenie programu, jeśli obraz nie został wczytany
18         }
19
20         Mat FlippedImage = new Mat();
21         Core.flip(image, FlippedImage, flipCode: 0);
22
23         HighGui.imshow("Dog", FlippedImage); // Wyświetlenie
24         HighGui.waitKey();
25         HighGui.destroyAllWindows();
26     }
27 }

```

- w pionie



- w poziomie



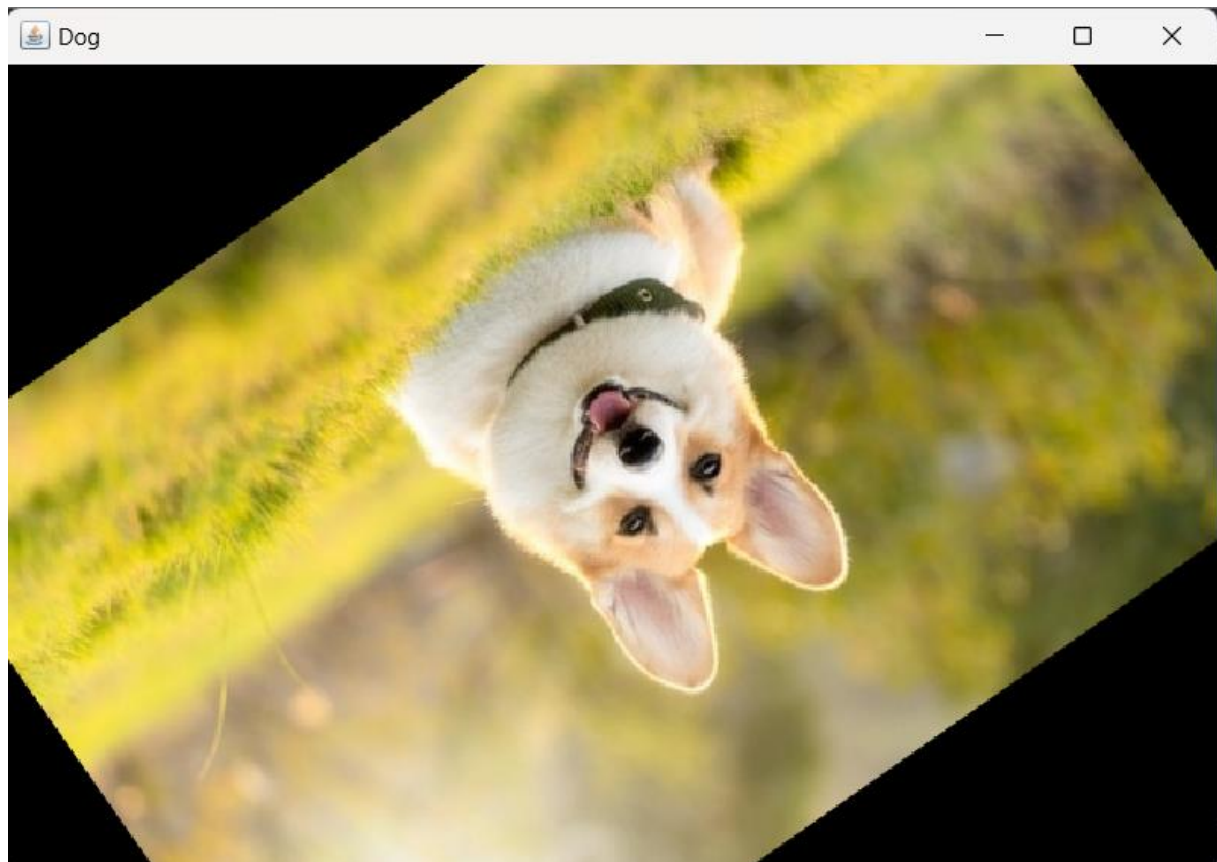
Flip:

1 – odbicie poziome

0 – odbicie pionowe

-1 – poziome i pionowe

Zad 3



```

1 > import ...
2
3
4
5
6 public class Main {
7     public static void main(String[] args) {
8         System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie biblioteki OpenCV
9         Mat image = Imgcodecs.imread( filename: "dog.jpg"); // Wczytanie "dog.jpg" do obiektu typu Mat (macierzy)
10
11         if (image.empty()) {
12             System.out.println("Fail, try again");
13             return;
14         }
15         //srodek obrazu
16         Point center = new Point( x: image.cols() / 2, y: image.rows() / 2);
17         double angle = -145; //kat obrotu pozytywny w lewo na minusie w prawo
18         double scale = 1.0; //brak zmian przy skalowaniu
19
20         Mat rotation = Imgproc.getRotationMatrix2D(center, angle, scale);
21         Size size = new Size(image.cols(), image.rows());
22
23         Mat rotatedImage = new Mat();
24         Imgproc.warpAffine(image, rotatedImage, rotation, size);
25         HighGui.imshow( winname: "Dog", rotatedImage); // Wyświetlenie
26         HighGui.waitKey();
27         HighGui.destroyAllWindows();
28     }
29 }

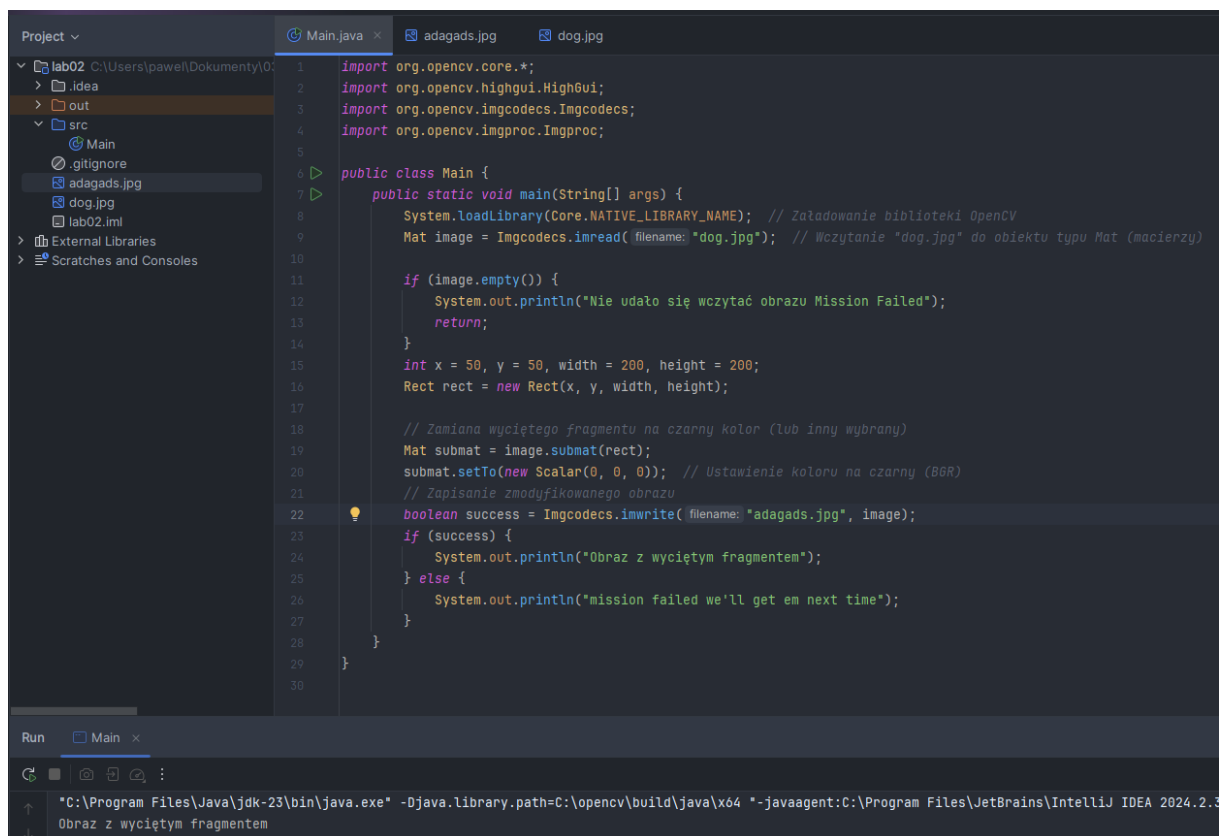
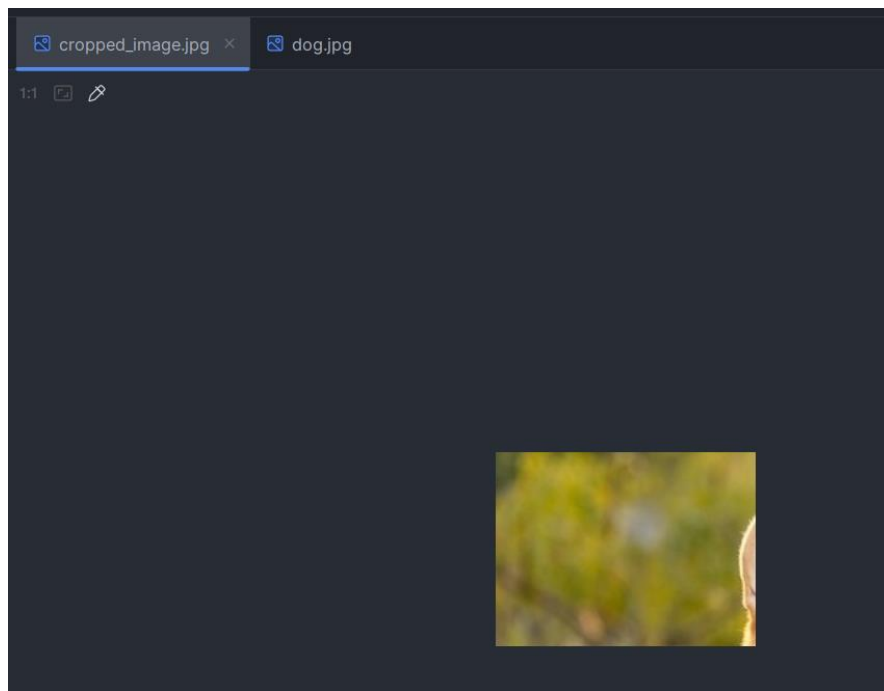
```

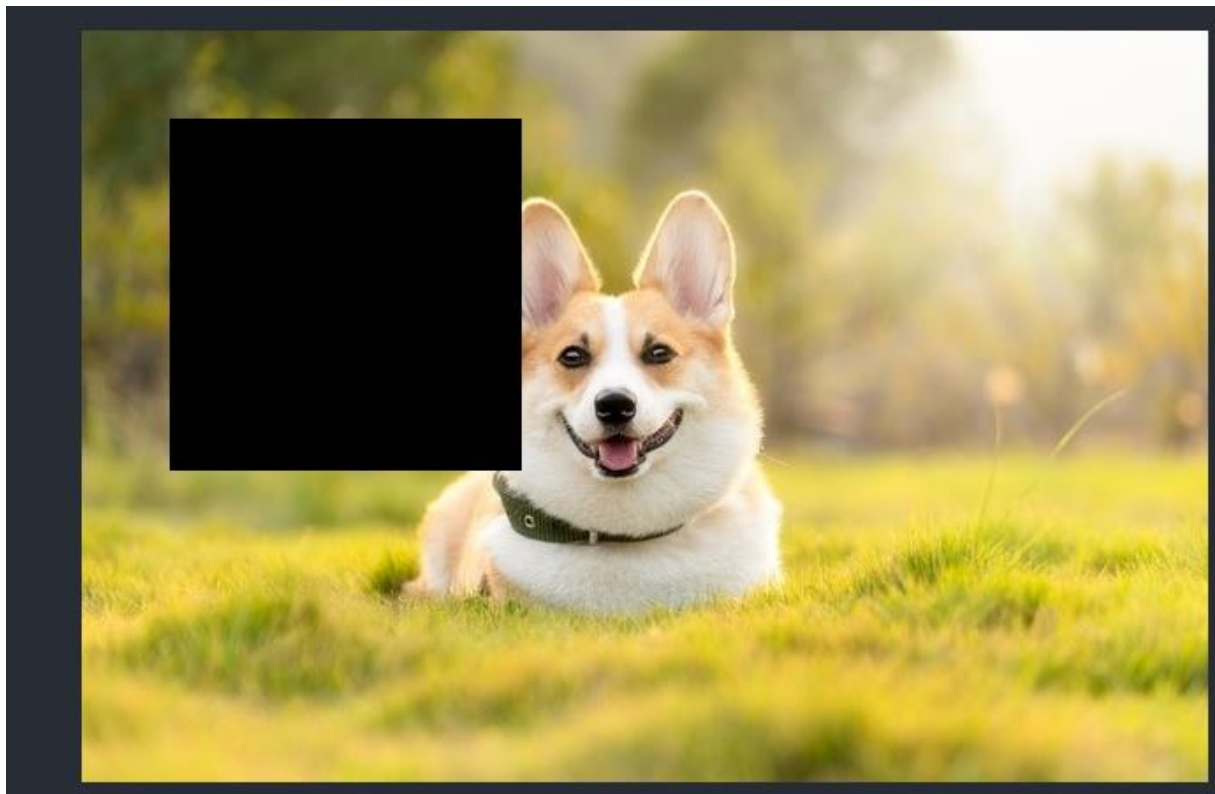
Zad. 4

```

1 > import ...
2
3
4
5
6 public class Main {
7     public static void main(String[] args) {
8         System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie biblioteki OpenCV
9         Mat image = Imgcodecs.imread( filename: "dog.jpg"); // Wczytanie "dog.jpg" do obiektu typu Mat (macierzy)
10
11         if (image.empty()) {
12             System.out.println("Fail, try again");
13             return;
14         }
15
16         int y = 50, x = 50;
17         int width = 200;
18         int height = 150;
19         Rect rect = new Rect(x, y, width, height);
20         Mat roi = new Mat(image, rect);
21         Imgcodecs.imwrite( filename: "abc.jpg", roi);
22         Imgcodecs.imwrite( filename: "cropped_image.jpg", roi);
23
24         System.out.println("Wycięty fragment zapisano jako 'cropped_image.jpg'");
25         // HighGui.imshow("Original", image);
26         // HighGui.imshow("Cropped", roi);
27         // HighGui.waitKey();
28         // HighGui.destroyAllWindows();
29     }
30 }

```





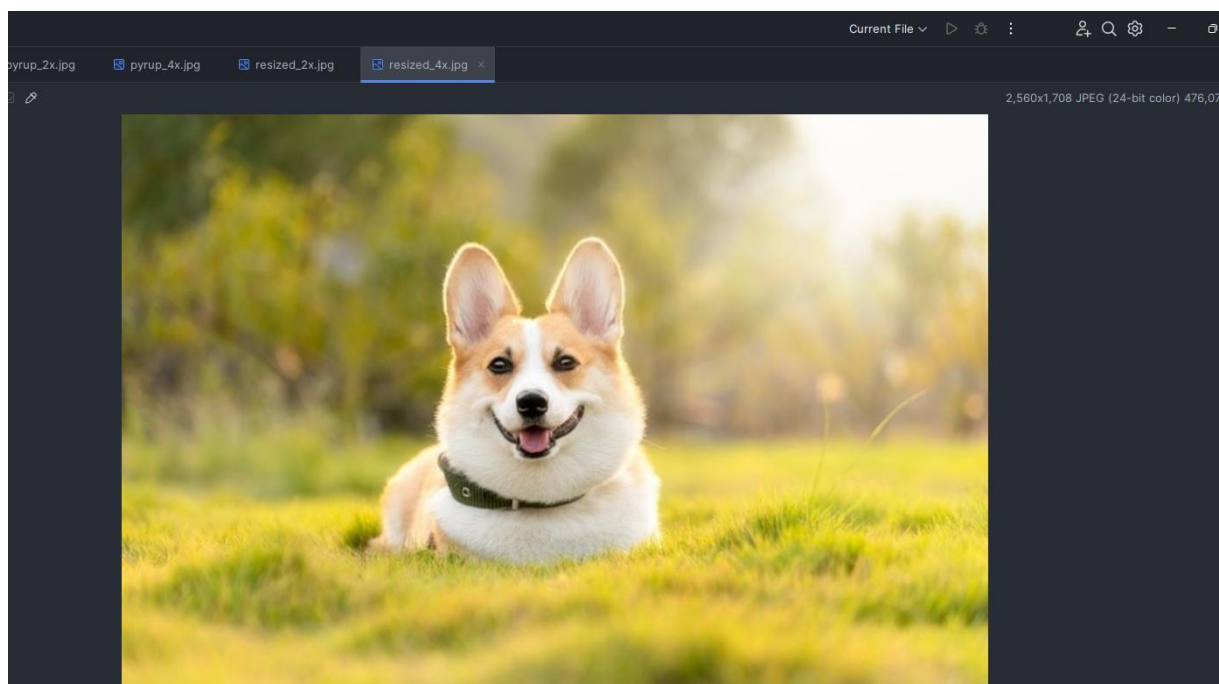
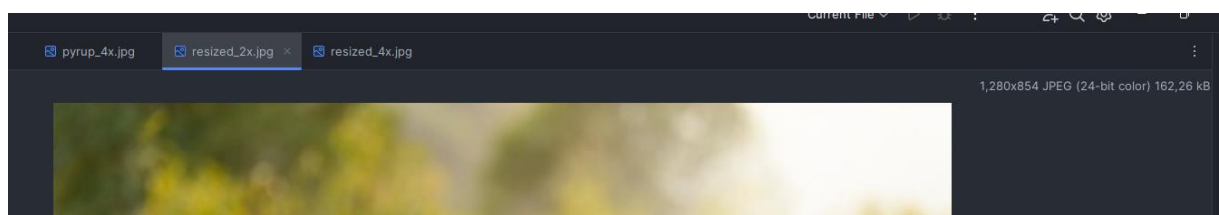
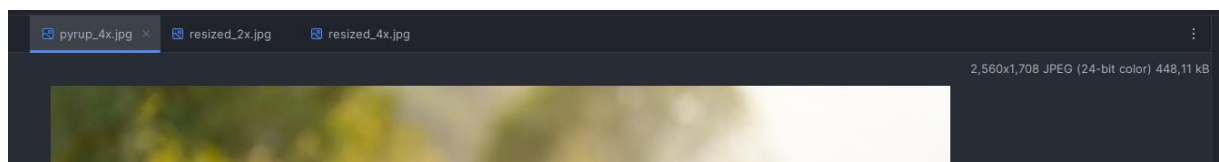
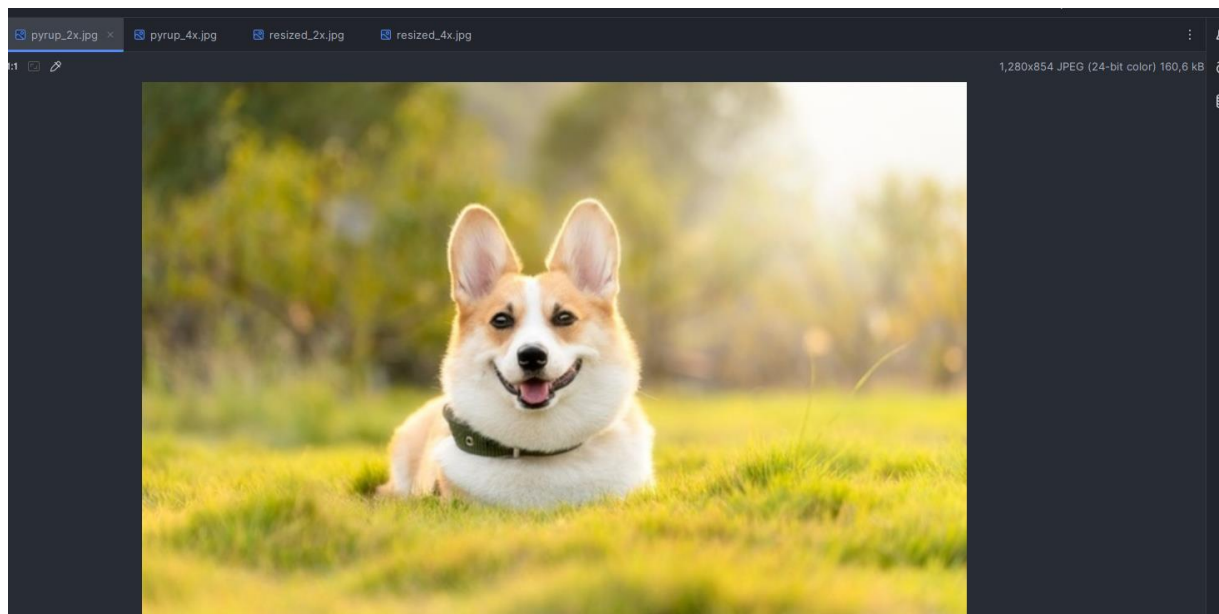
Zadanie 5

```
Project: lab02
C:\Users\pawel.Dokumenty\lab02
> idea
> out
> src
  @ Main
  .gitignore
  adagads.jpg
  dog.jpg
  lab02.iml
  pyrup_2x.jpg
  pyrup_4x.jpg
  resized_2x.jpg
  resized_4x.jpg
> External Libraries
> Scratches and Consoles

1  import java
2  public class Main {
3      public static void main(String[] args) {
4          System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie biblioteki OpenCV
5          Mat image = Imgcodecs.imread("dog.jpg"); // Wczytanie "dog.jpg" do obiektu typu Mat (macierzy)
6
7          if (image.empty()) {
8              System.out.println("Nie udało się wczytać obrazu, spróbuj ponownie");
9              return;
10         }
11
12         // Powiększenie obrazu za pomocą resize
13         Mat powiekszonyresized2x = new Mat();
14         Mat powiekszonyresized4x = new Mat();
15         Imgproc.resize(image, powiekszonyresized2x, new Size(image.cols() * 2, image.rows() * 2), 0, 0, Imgproc.INTER_LINEAR);
16         Imgproc.resize(image, powiekszonyresized4x, new Size(image.cols() * 4, image.rows() * 4), 0, 0, Imgproc.INTER_LINEAR);
17
18         // Zapisywanie powiększonych obrazów za pomocą resize
19         Imgcodecs.imwrite("resized_2x.jpg", powiekszonyresized2x);
20         Imgcodecs.imwrite("resized_4x.jpg", powiekszonyresized4x);
21
22         // Powiększenie obrazu za pomocą pyrUp
23         Mat pyrup2x = new Mat();
24         Mat pyrup4x = new Mat();
25         Imgproc.pyrUp(image, pyrup2x, new Size(image.cols() * 2, image.rows() * 2));
26         Imgproc.pyrUp(pyrup2x, pyrup4x, new Size(image.cols() * 4, image.rows() * 4));
27
28         // Zapisywanie powiększonych obrazów za pomocą pyrUp
29         Imgcodecs.imwrite("pyrup_2x.jpg", pyrup2x);
30         Imgcodecs.imwrite("pyrup_4x.jpg", pyrup4x);
31     }
32 }
33 }
```

Run Main

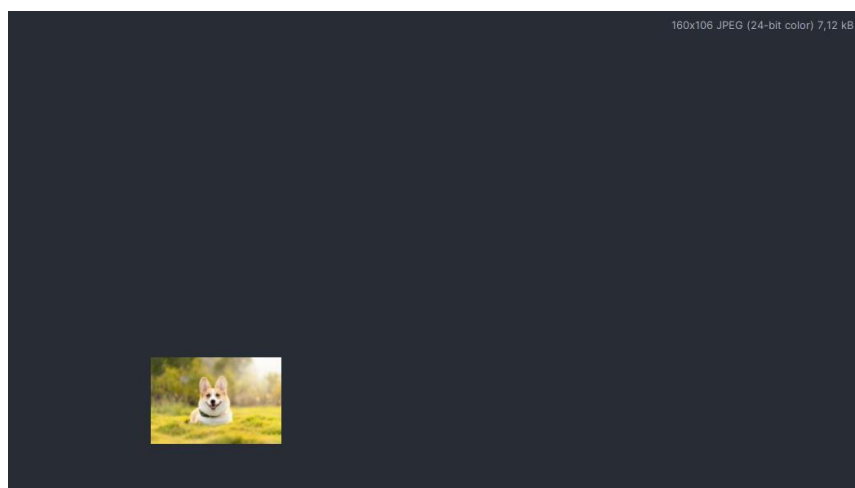
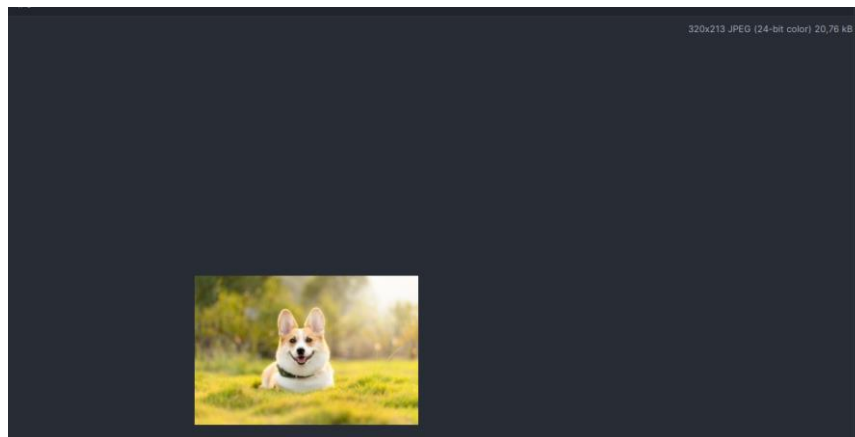
"C:\Program Files\Java\jdk-23\bin\java.exe" -Djava.library.path=C:\opencv\build\java\x64 -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\lib\idea_rt.jar=51232:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\bin Process finished with exit code 0

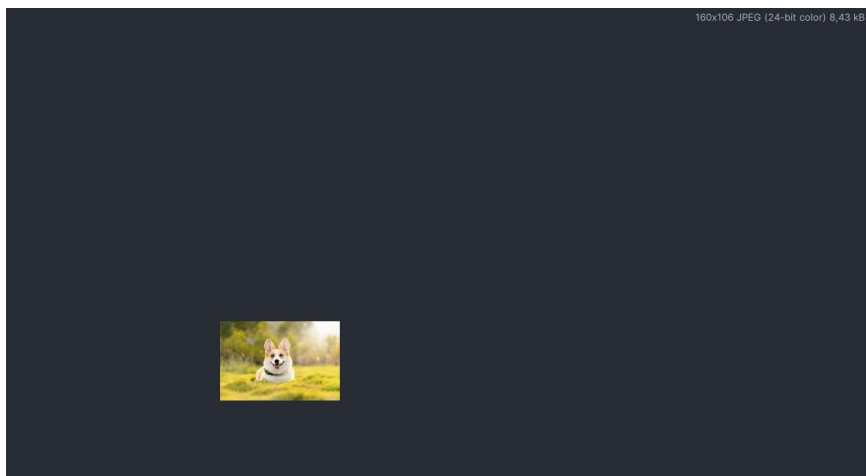
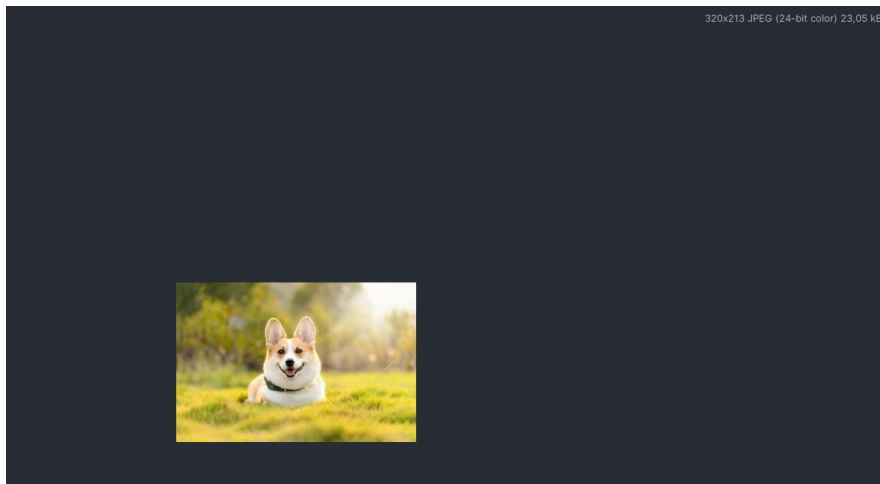


Metoda pyrUp może zapewnić lepsze efekty przy wielokrotnym powiększaniu, ponieważ bazuje na wyższych poziomach obrazów gaussowskich

Zad 6.

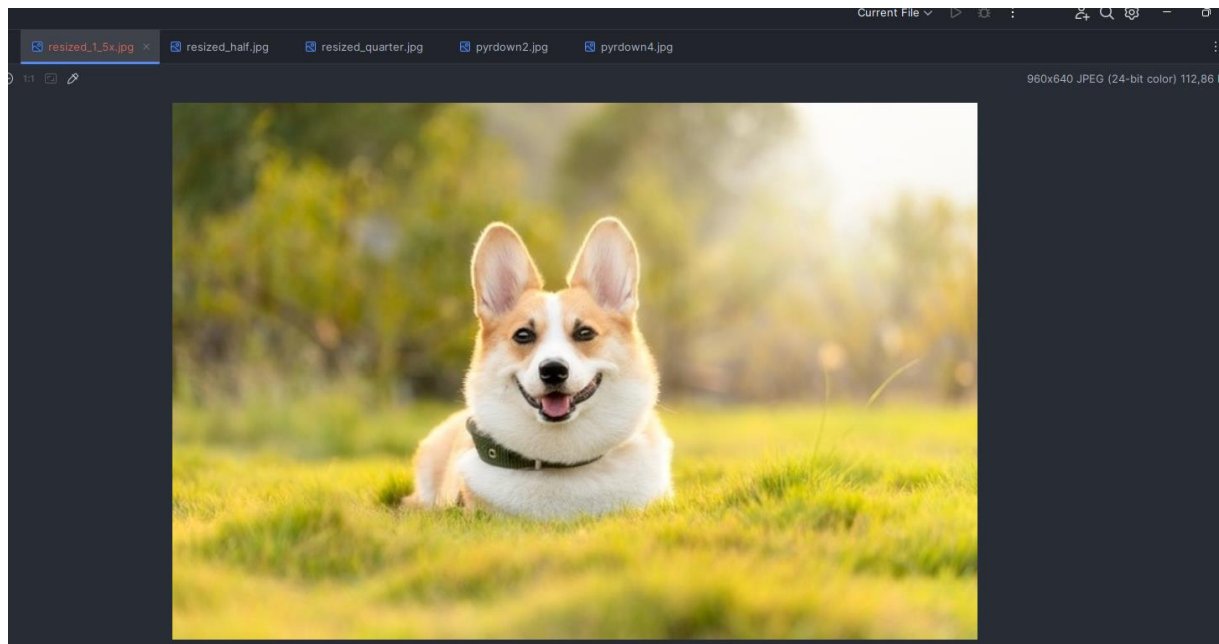
```
1 > import ...
4
5 public class Main {
6     public static void main(String[] args) {
7         System.loadLibrary(Core.NATIVE_LIBRARY_NAME); // Załadowanie biblioteki OpenCV
8         Mat image = Imgcodecs.imread(filename: "dog.jpg"); // Wczytanie "dog.jpg" do obiektu typu Mat (macierzy)
9
10        if (image.empty()) {
11            System.out.println("Mission failed");
12            return;
13        }
14
15        // Pomniejszanie obrazu za pomocą resize
16        Mat resizedHalf = new Mat();
17        Mat resizedQuarter = new Mat();
18        Imgproc.resize(image, resizedHalf, new Size( width: image.cols() / 2, height: image.rows() / 2), fx: 0, fy: 0, Imgproc.INTER_LINEAR);
19        Imgproc.resize(image, resizedQuarter, new Size( width: image.cols() / 4, height: image.rows() / 4), fx: 0, fy: 0, Imgproc.INTER_LINEAR);
20
21        // Zapisywanie pomniejszonych obrazów za pomocą resize
22        Imgcodecs.imwrite( filename: "resized_half.jpg", resizedHalf);
23        Imgcodecs.imwrite( filename: "resized_quarter.jpg", resizedQuarter);
24
25        // Pomniejszanie obrazu za pomocą pyrDown
26        Mat pyrDownHalf = new Mat();
27        Mat pyrDownQuarter = new Mat();
28        Imgproc.pyrDown(image, pyrDownHalf, new Size( width: image.cols() / 2, height: image.rows() / 2));
29        Imgproc.pyrDown(pyrDownHalf, pyrDownQuarter, new Size( width: image.cols() / 4, height: image.rows() / 4));
30
31        // Zapisywanie pomniejszonych obrazów za pomocą pyrDown
32        Imgcodecs.imwrite( filename: "pyrdown2.jpg", pyrDownHalf);
33        Imgcodecs.imwrite( filename: "pyrdown4.jpg", pyrDownQuarter);
34
35    }
36 }
```





Zadanie 7

```
1  > import ...
2
3
4
5  public class Main {  Pawelgabrieljonca *
6  public static void main(String[] args) {  Pawelgabrieljonca *
7      System.loadLibrary(Core.NATIVE_LIBRARY_NAME);  // Załadowanie biblioteki OpenCV
8      Mat image = Imgcodecs.imread( filename: "dog.jpg");  // Wczytanie "dog.jpg" do obiektu typu Mat (macierzy)
9
10
11      if (image.empty()) {
12          System.out.println("Nie udało się wczytać obrazu, spróbuj ponownie");
13          return;
14      }
15      // Powiększanie obrazu o współczynnik 1,5
16      Mat resizedImage = new Mat();
17      Imgproc.resize(image, resizedImage, new Size( width: image.cols() * 1.5, height: image.rows() * 1.5), fx: 0, fy: 0, Imgproc.INTER_LINEAR);
18
19      // Zapisywanie powiększonego obrazu
20      Imgcodecs.imwrite( filename: "resized_1_5x.jpg", resizedImage);
21  }
22  }
```



Należy wybrać metodę `resize` ponieważ ona umożliwia precyzyjne określenie współczynnika skalowania, który nie musi być całkowity. Metoda `pyrUp` nie działa dla skalowania 1.5 nie można jej do tego użyć jak już to `x2`

Wnioski:

Wszystkie zadania zostały wykonane zgodnie z poleceniami z pliku. Zadania pomogły w dalszym zapoznaniu się w bibliotece OpenCV, mogłem poznać i zobaczyć jak manipulować obrazami.