PWSZ Tarnów
Informatyka

Architektury Systemów Rozproszonych

Dokumentacja

Wykonawcy:
Kamil Kapałka
Marcin Wijas

Prowadzący:
dr inż. Łukasz Mik

Wstęp

Program działa w architekturze klient-serwer. Wykorzystuje mechanizm RMI(Remote Method Invocation), który pozwala aplikacji klienta wywoływać metody zdalnych obiektów języka Java, czyli obiektów z innej przestrzeni adresowej, mogącej znajdować się na tej samej lub innej maszynie.

RMI obejmuje trzy procesy:

- proces klienta, wywołującego metody zdalnych obiektów
- proces serwera, dostarczającego zdalnych obiektów
- proces rejestru, wiążącego obiekty z ich nazwami

Obiekty są zgłaszane do rejestru przez serwer. Aplikacja klienta może uzyskać dostęp do zarejestrowanego obiektu pod warunkiem, że najpierw używając wcześniej poznanej nazwy obiektu, skorzysta z rejestru, by otrzymać referencję do obiektu odpowiadającą tej nazwie.

Binaryzacja jest operacją punktową, której wynikiem są obrazy binarne, czyli takie, w których próbki obrazu mają tylko jedną z dwóch wartości. Operacja ta jest często określana mianem progowania.

Binaryzacja z dolnym progiem:

$$g(x, y) = \begin{matrix} 0 \text{ jeżeli } f(x,y) \leq t \\ 1 \text{ jeżeli } f(x,y) > t \end{matrix}$$

Binaryzacja z górnym progiem:

$$g(x, y) = \begin{matrix} 0 \text{ jeżeli } f(x,y) \geq t \\ 1 \text{ jeżeli } f(x,y) < t \end{matrix}$$

Binaryzacja dwuprogowa:

$$g(x, y) = \begin{matrix} 0 \text{ jeżeli } f(x,y) < u \vee f(x,y) > v \\ 1 \text{ jeżeli } u \leq f(x,y) \leq v \end{matrix}$$

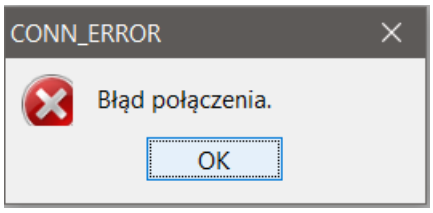
Binaryzacja warunkowa:

$$g(x, y) = \begin{matrix} 0 \text{ jeżeli } u \geq f(x,y) \\ s \text{ jeżeli } u < f(x,y) \leq v \\ 1 \text{ jeżeli } f(x,y) > v \end{matrix}$$

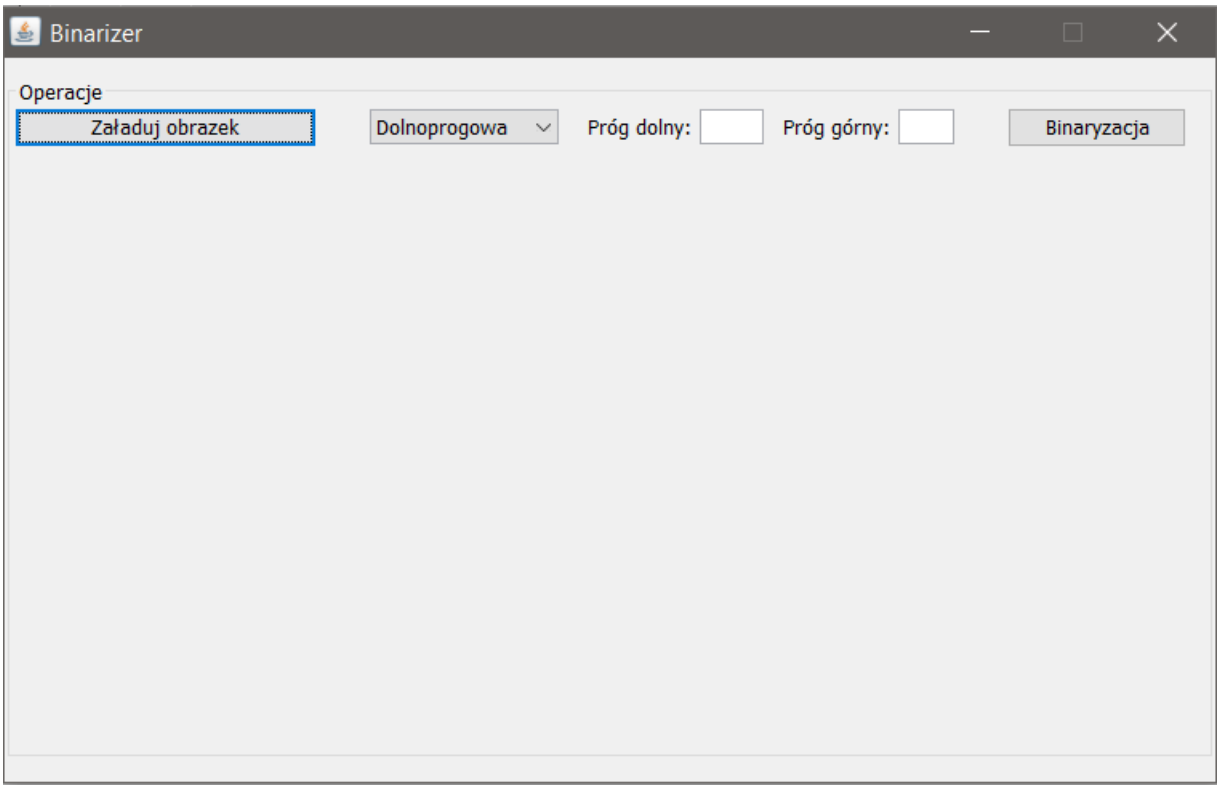
Gdzie T jest predefiniowanym progiem. Wartość 0 może być interpretowana jako czerń, 1 - jako biel.

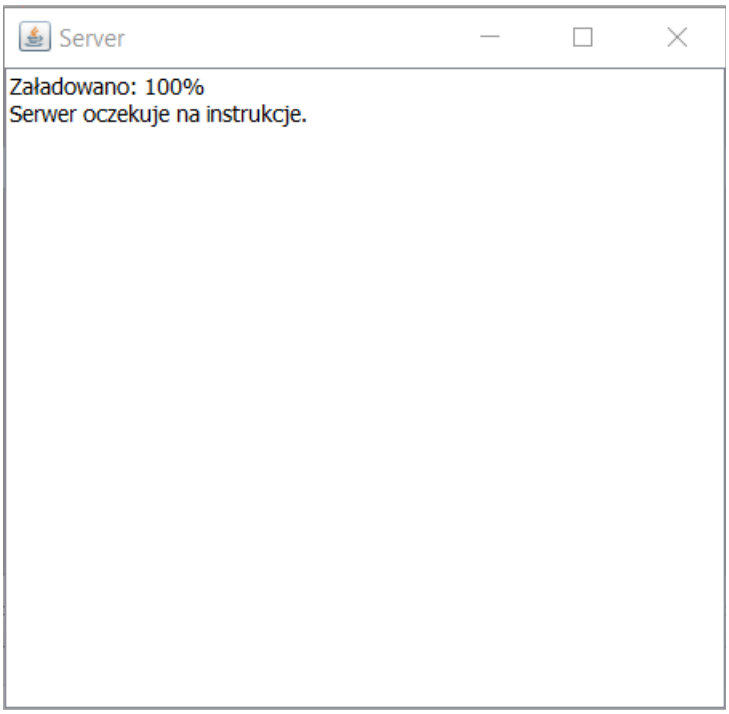
Funkcjonalność programu

Ten komunikat uruchamia się gdy chcesz zbinaryzować plik, do którego nie ma ścieżki.

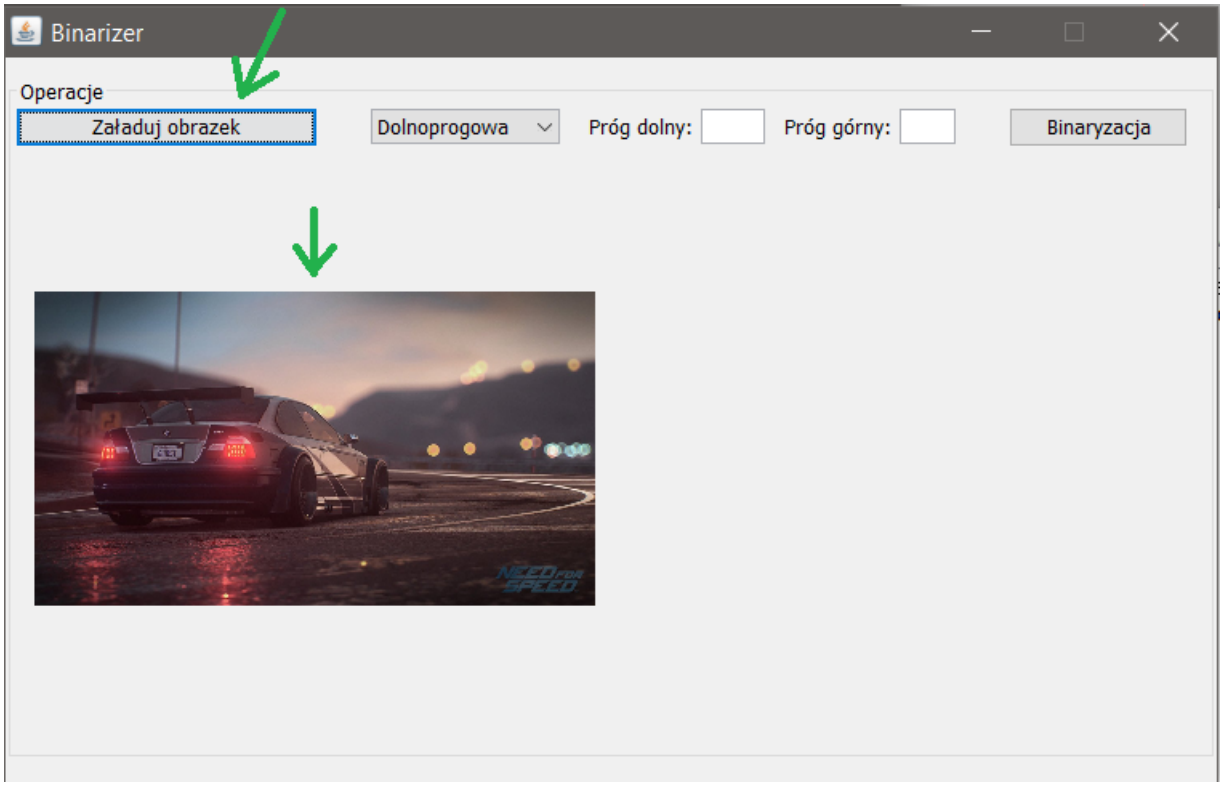


Po prawidłowym uruchomieniu programu użytkownikowi ukazuje się główne okno programu, a także okno servera. W przypadku błedu serwer zwróci komunikat: ">>>>> Wystąpił błąd: ".





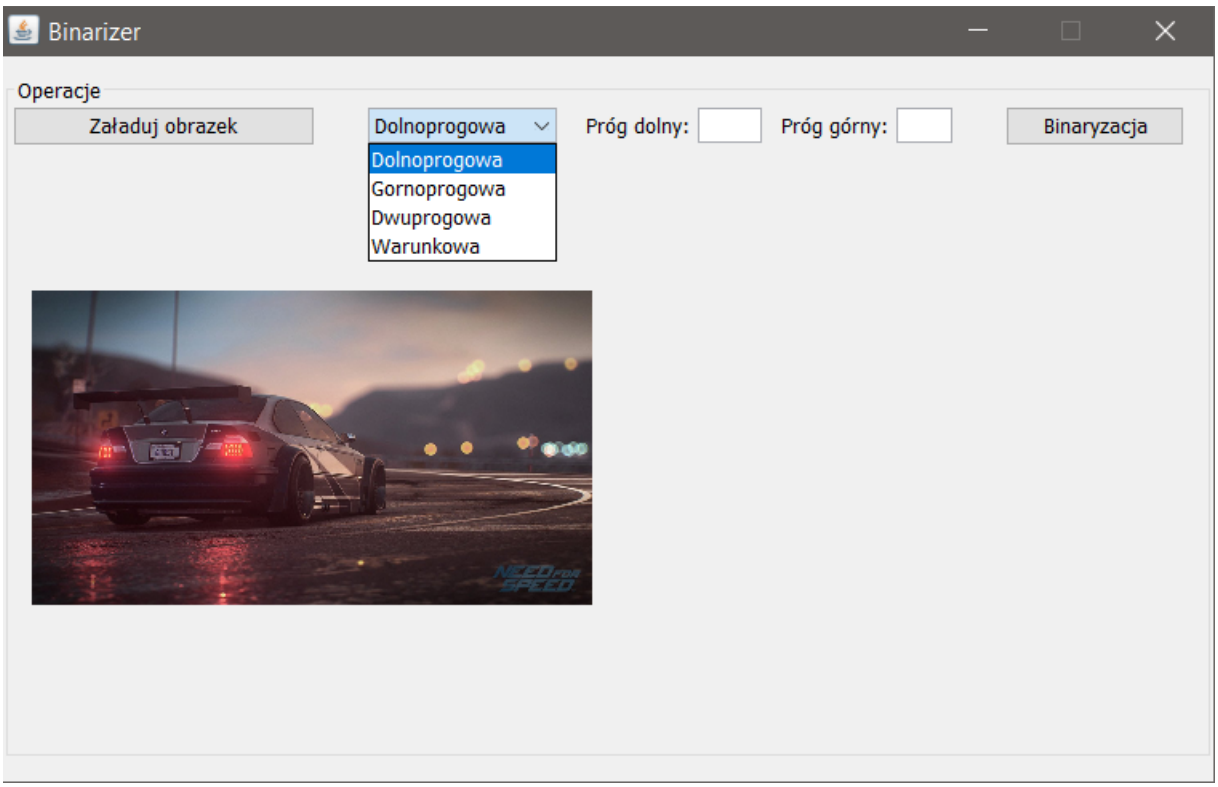
Przyciskiem "Załaduj obrazek" wybieramy obrazek, który chcemy binaryzować.
Zalecane są następujące formaty obrazków: .jpg, .jpeg, .png, .bmp,
Po wybraniu obrazka wyświetli się nam po lewej stronie.



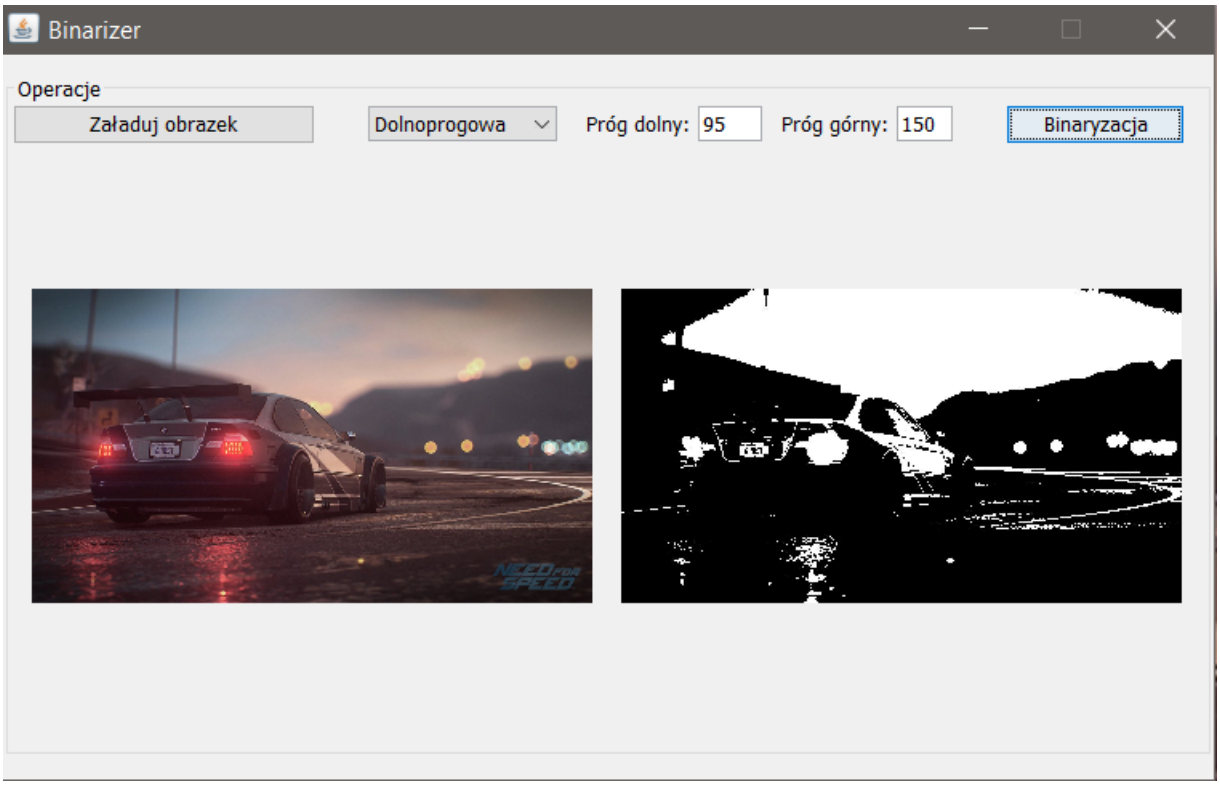
W przypadku prawidłowego wczytania obrazka serwer wyświetla komunikat: "Załadowano plik: sciezka_do_pliku". W przeciwnym razie Server wyświetli błąd: "Nie udało się załadować pliku."

Fragment kodu odpowiedzialny za wczytanie obrazka:

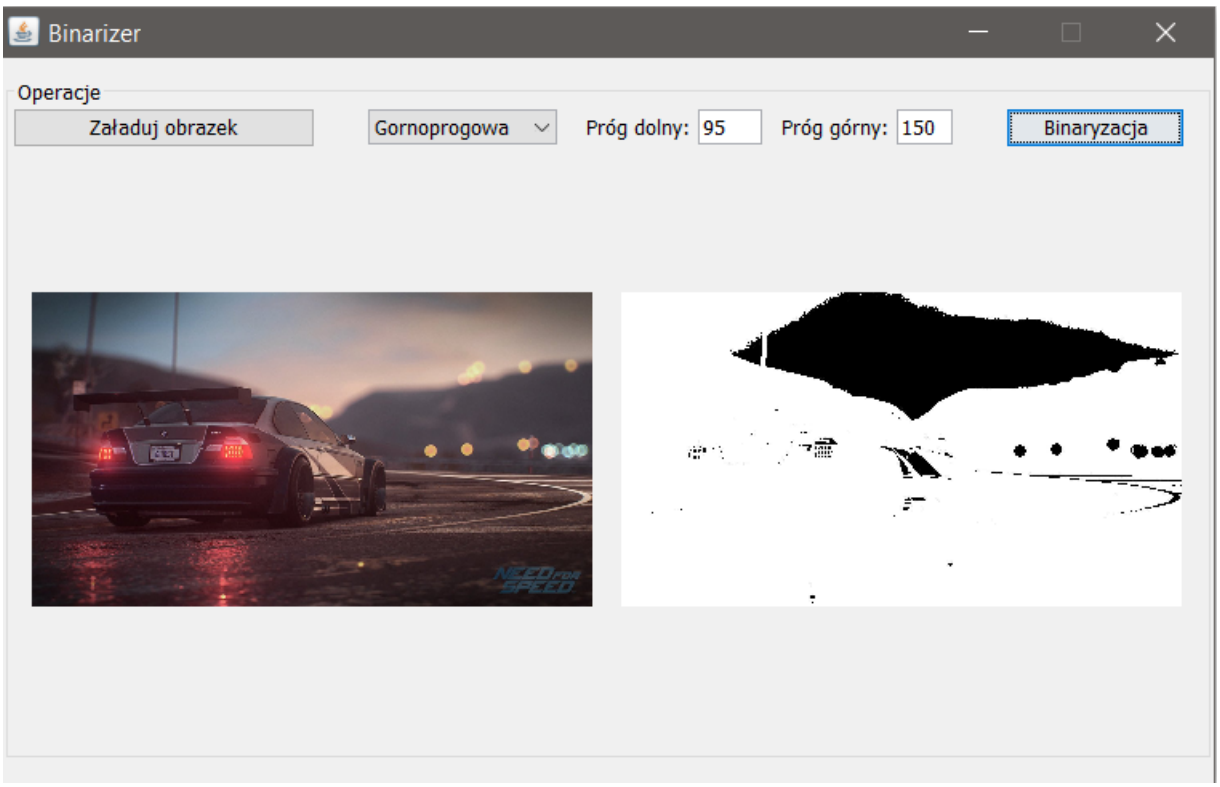
```
@Override
public void loadFile(File file) throws RemoteException{
    try{
        loadedFile=file;
        Extension=file.getName().substring(file.getName().lastIndexOf(".")+1);
    gui.getTextArea().setText(gui.getTextArea().getText()+
            "\nZaładowano plik: "+file.getCanonicalPath()+" .");
    }catch (IOException ioex){
        gui.getTextArea().setText(gui.getTextArea().getText()+
            "\nNie udało się załadować pliku.");
    }
```



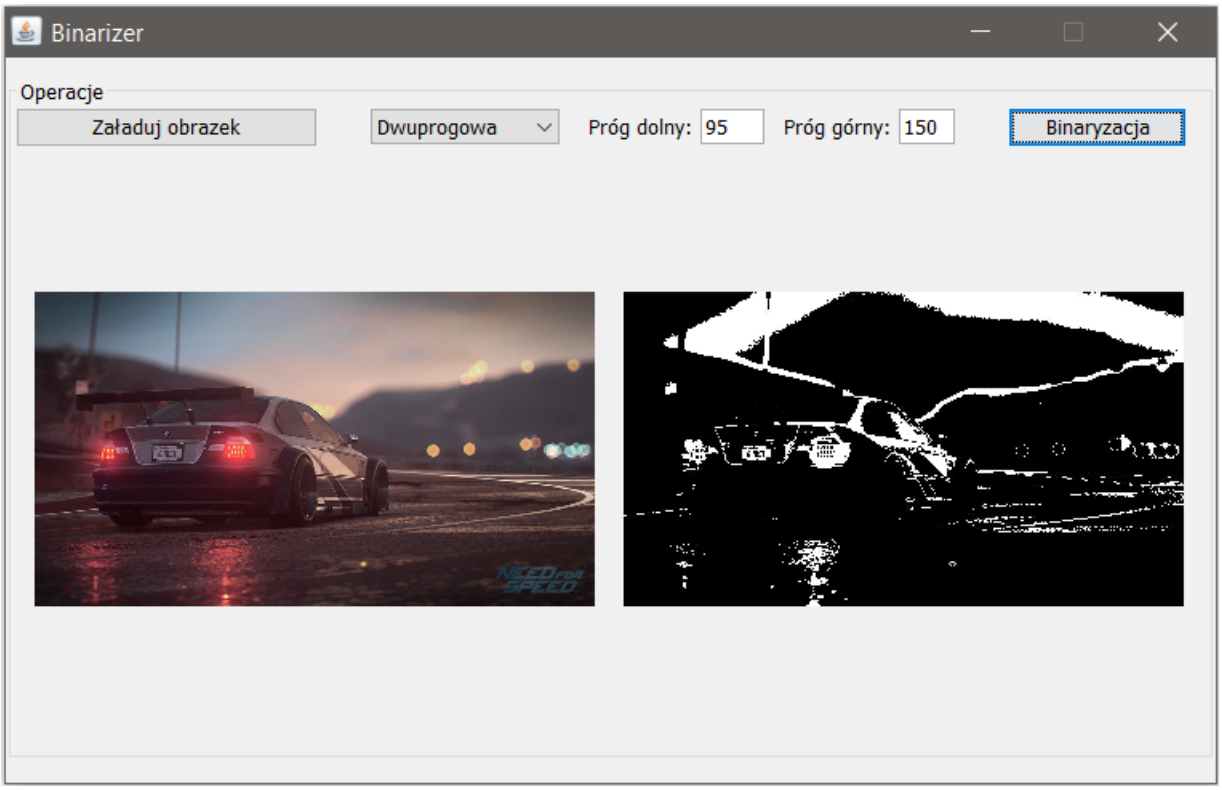
Binaryzacja dolnoprogowa:



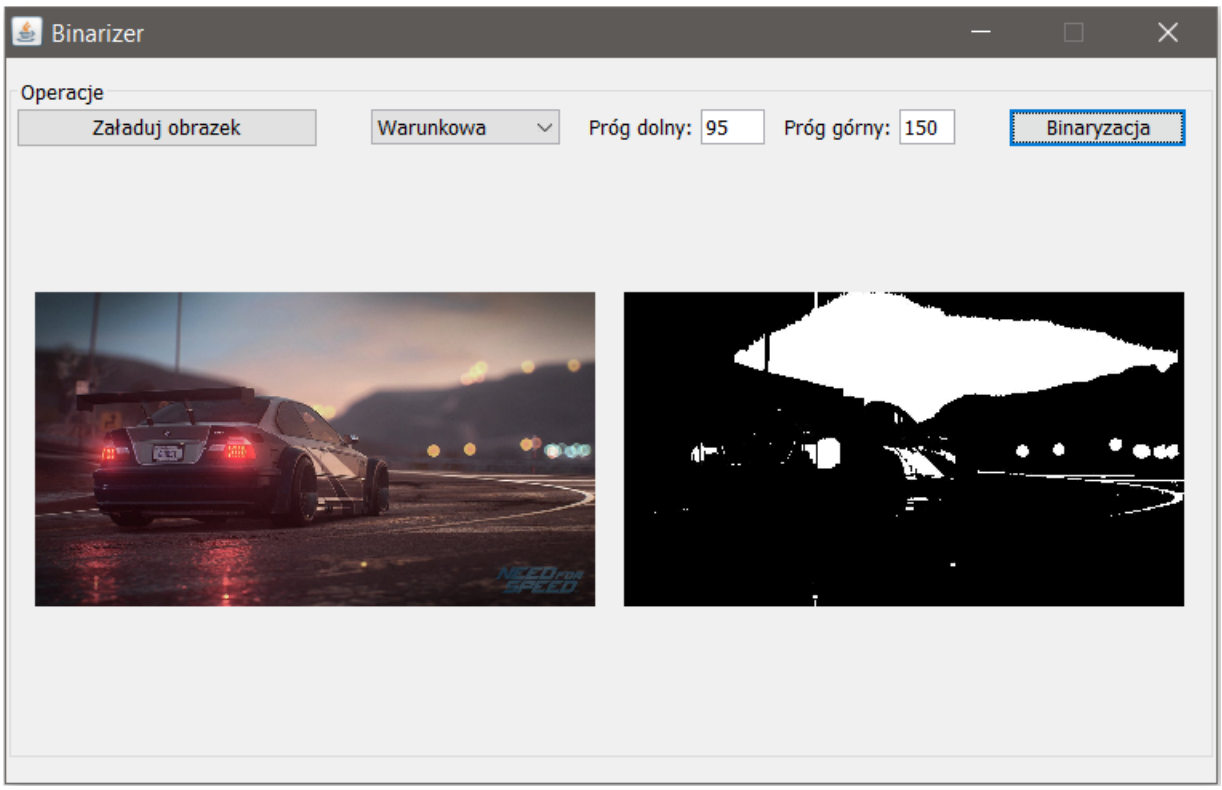
Binaryzacja górnoprogowa:



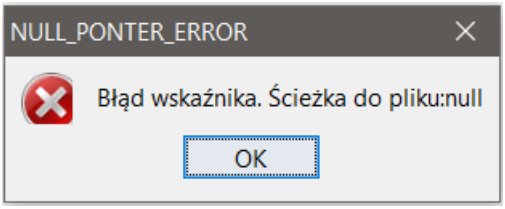
Binaryzacja dwuprogowa:



Binaryzacja warunkowa:



W zależności od wybranej binaryzacji, znajdują się odpowiednie zabezpieczenia. Jeśli przy binaryzacji dolnoprogowej dolny próg zostanie ustawiony na 0 to zostanie zwrócony błąd:



Podobnie jest w przypadku innych opcji:

binaryzacja górnoprogowa górny próg: 0 (zwraca błąd)
binaryzacja dwuprogowa dolny próg i górny próg: 0 (zwraca błąd)
binaryzacja warunkowa dolny próg lub górny próg: 0 (zwraca błąd)

Fragment kodu odpowiedzialny za binaryzację obrazu:

```
    @Override
   public String binarize(String mode, int value1, int value2) throws
RemoteException{
```

```
        if(mode==null) mode="Dolnoprogowa";
        if(value1==0 && "Dolnoprogowa".equals(mode)){
                        gui.getTextArea().setText(gui.getTextArea().getText()+
                "\nBlad: nie ustawiono dolnego progu.");
                        return null;
                    }
        if(value2==0 && "Gornoprogowa".equals(mode)){
                        gui.getTextArea().setText(gui.getTextArea().getText()+
                "\nBlad: nie ustawiono gornego progu.");
                        return null;
                    }
        if(value1==0 && value2==0 && ("Dwuprogowa".equals(mode) ||
"Warunkowa".equals(mode))){
            gui.getTextArea().setText(gui.getTextArea().getText()+
                "\nBlad: nie ustawiono progów.");
                        return null;
        }
        gui.getTextArea().setText(gui.getTextArea().getText()+
                "\nRozpoczęto binaryzację pliku. Tryb: "+mode+ ". Progi: "+value1+",
"+value2+".");
        switch(Extension){
            case "jpg": case "jpeg":
            case "bmp":
            case "png":
                try{
              newFile=loadedFile.getPath().substring(0,loadedFile.getPath().lastIndexOf("."))
+"-binarized."+Extension;
            gui.getTextArea().setText(gui.getTextArea().getText()+
                "\nUtworzono sciezke do nowego pliku.");
            BufferedImage loadedImg =ImageIO.read(loadedFile);
gui.getTextArea().setText(gui.getTextArea().getText()+
                "\ndebug");
            WritableRaster raster= (WritableRaster) loadedImg.getData();
             gui.getTextArea().setText(gui.getTextArea().getText()+
                "\ndebug");
            int pixel,blue,green,red,greyscale;
            int[] black={0,0,0};
            int[] white={255,255,255};
            int[] warunkowa={0,0,0};

            for(int i=0;i<raster.getWidth();i++){
                for(int j=0;j<raster.getHeight();j++){
                    pixel = loadedImg.getRGB(i, j);
                    blue = pixel & 0xff;
                    green = (pixel & 0xff00) >> 8;
```

```
            red = (pixel & 0xff0000) >> 16;
            greyscale=(blue+green+red)/3;
            if ("Dolnoprogowa".equals(mode)){

               if(greyscale<=value1){
                  raster.setPixel(i, j, black);
               }
               else {
                  raster.setPixel(i, j, white);
               }
            }
            else if("Gornoprogowa".equals(mode)){

               if(greyscale>=value2) raster.setPixel(i, j, black);
               else raster.setPixel(i, j, white);
            }
            else if("Dwuprogowa".equals(mode)){
               if(greyscale<=value1 || greyscale>value2) raster.setPixel(i,j,black);
               else raster.setPixel(i,j,white);
            }
            else if("Warunkowa".equals(mode)){
               if(greyscale<=value1){
                  raster.setPixel(i,j,black);
                  warunkowa=black;
               }
               else if(greyscale>value2){
                  raster.setPixel(i,j,white);
                  warunkowa=white;
               }
               else if(i==1){
                  if(greyscale<value1+Math.round((value2-value1)/2)){
                     raster.setPixel(i,j,black);
                     warunkowa=black;
                  }
                  else{ raster.setPixel(i,j,white);
                  warunkowa=white;
                  }
               }
               else raster.setPixel(i,j,warunkowa);
            }
         }
      }
      ColorModel colorModel = new ComponentColorModel(
ColorSpace.getInstance(ColorSpace.CS_sRGB),
new int[]{8, 8, 8}, // bits
```

```
    false, // hasAlpha
    false, // isPreMultiplied
    ComponentColorModel.OPAQUE,
    DataBuffer.TYPE_BYTE);
       newImage=new BufferedImage(colorModel,raster,false,null);
       gui.getTextArea().setText(gui.getTextArea().getText()+
          "\ndebug");
       ImageIO.write((RenderedImage)newImage,Extension,new File (newFile));

        gui.getTextArea().setText(gui.getTextArea().getText()+
          "\nUstawiono nowe informacje.");
        break;
          } catch (IOException ioex){
             gui.getTextArea().setText(gui.getTextArea().getText()+
          "\nNie odczytano pliku. Rozszerzenie: "+Extension+".");
             return null;
          }
       default:
          gui.getTextArea().setText(gui.getTextArea().getText()+
          "\nNie odczytano pliku. Rozszerzenie: "+Extension+".");
          return null;
    }
    gui.getTextArea().setText(gui.getTextArea().getText()+
          "\nOdsyłam ścieżkę do pliku do klienta: "+newFile+".");
    return newFile;
}
```