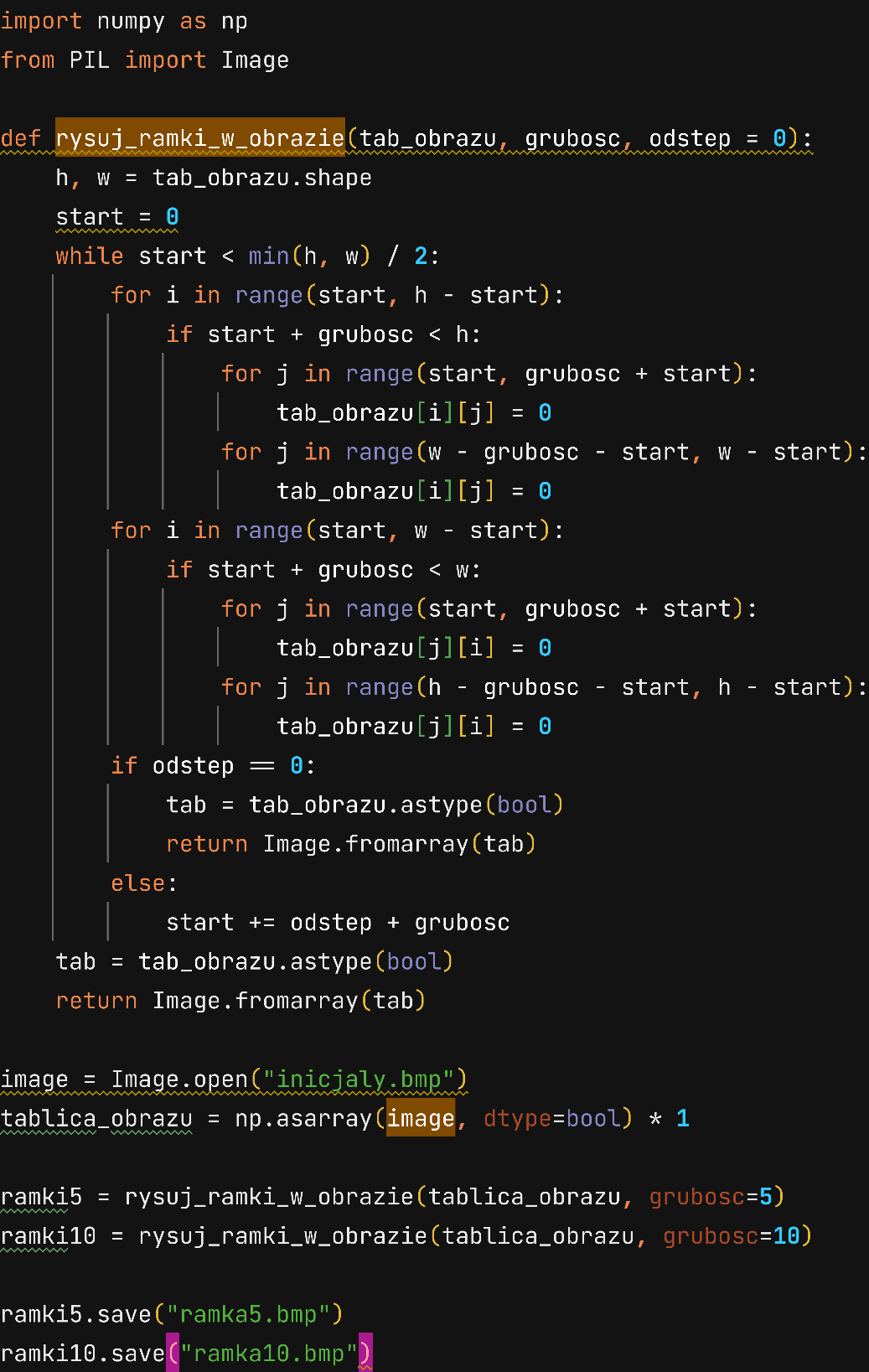
**Obrazy w trybie 1**



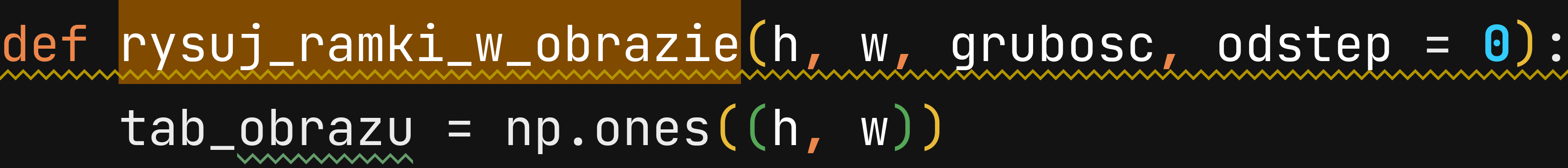
Wczytałem obraz inicjały 0 rozmiarach 150x50, następnie po zamianie go na tablicę podałem jako argument do funkcji. Funkcja ta działa dwojako, nie podając parametru odstęp automatycznie przypisuje do nie wartość 0. W taki sposób funkcja zrobi po prostu ramkę w obrazie. Na tym przykładzie rysowana jest ramka o grubości 5 i 10 widoczne poniżej

.

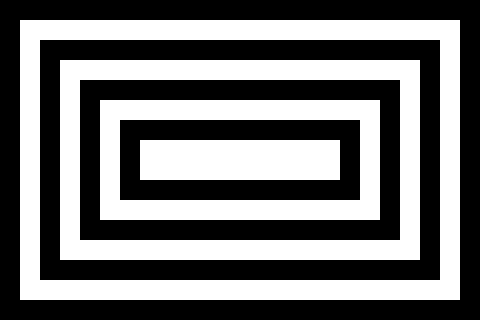
Ramka o grubości 5



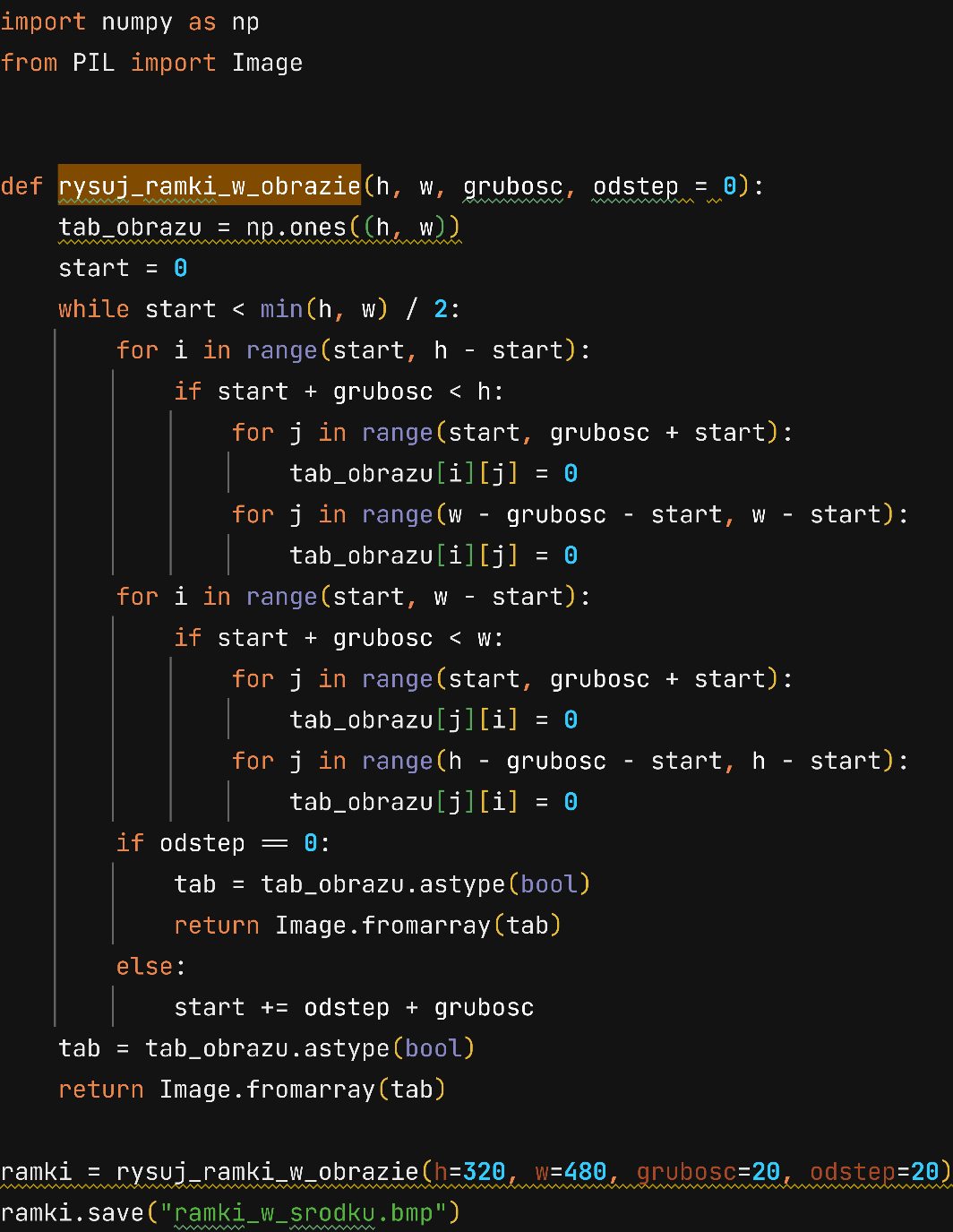
Ramka o grubości 10

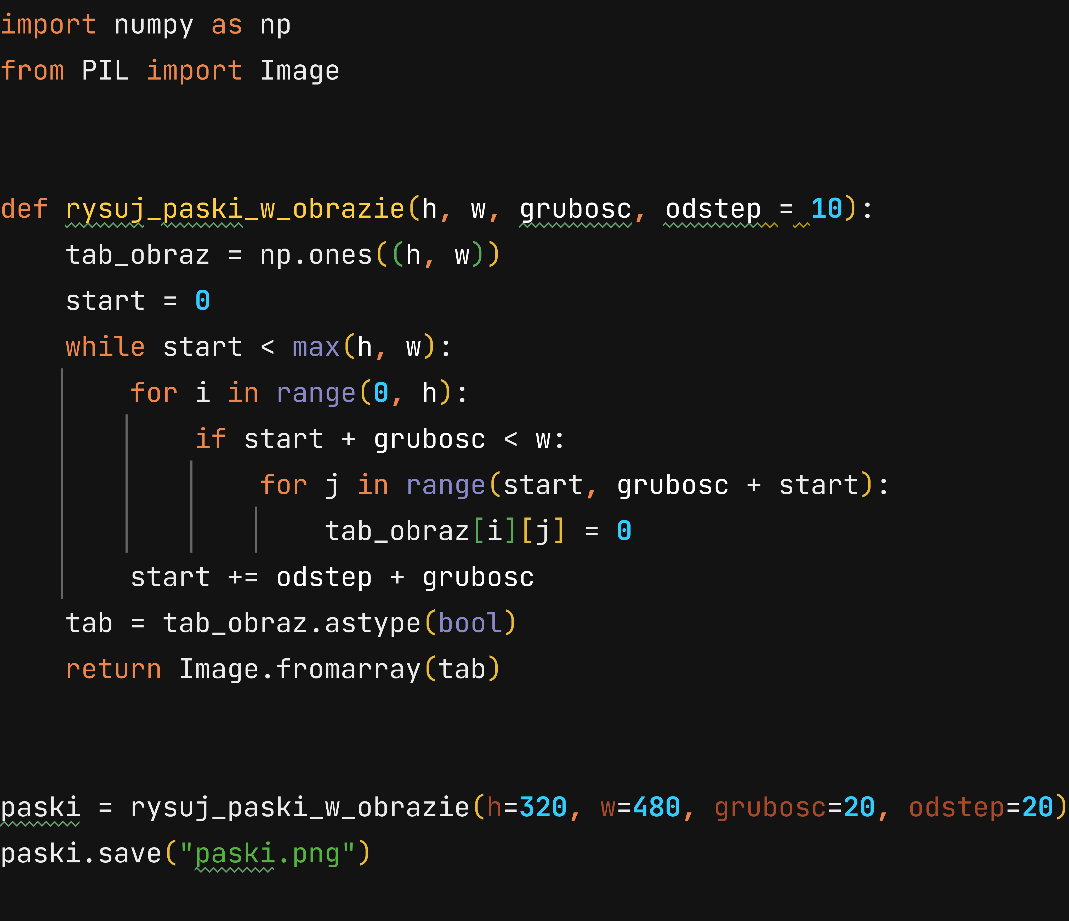
Ten sam kod spełnia funkcje tworzenia ramek na przemian, że raz jest biała raz czarna. Jedynie różni się, że zamiast wczytywać i przekształcać obrazek, od razu tworzę tablice: 

W tym przypadku obraz jest cały biały, a czarne ramki są po prostu dodawane poprzez zmianę punktu startowego. Parametry jakie przyjąłem są takie same jak w poprzednim przykładzie, ale tu zastosowałem dodatkowy parametr odstęp, który pozwala sterować odległością pomiędzy ramkami. Parametry: h=320. W=480, grubość=20, odstep=20

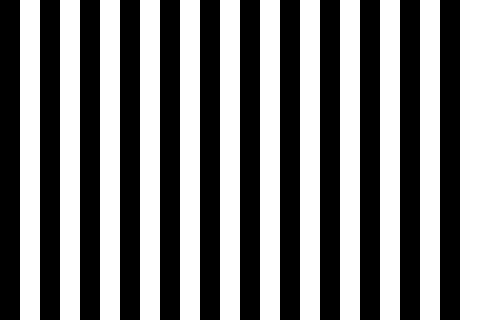
.

Ramki o grubości 20 i odstępie 20

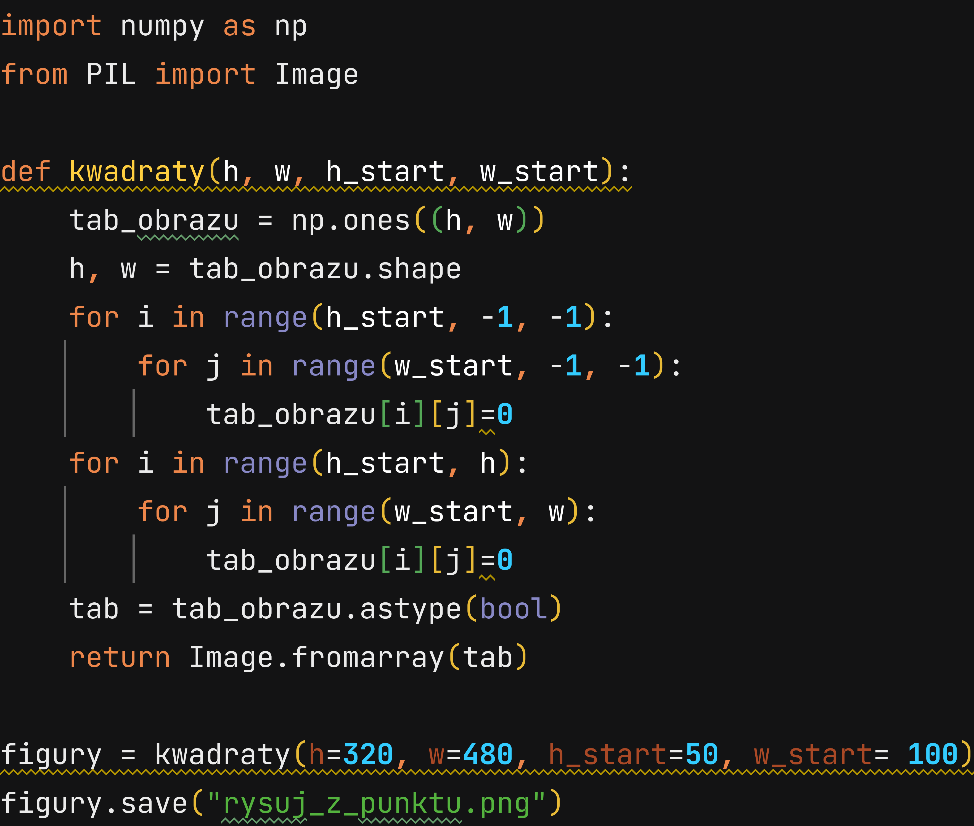
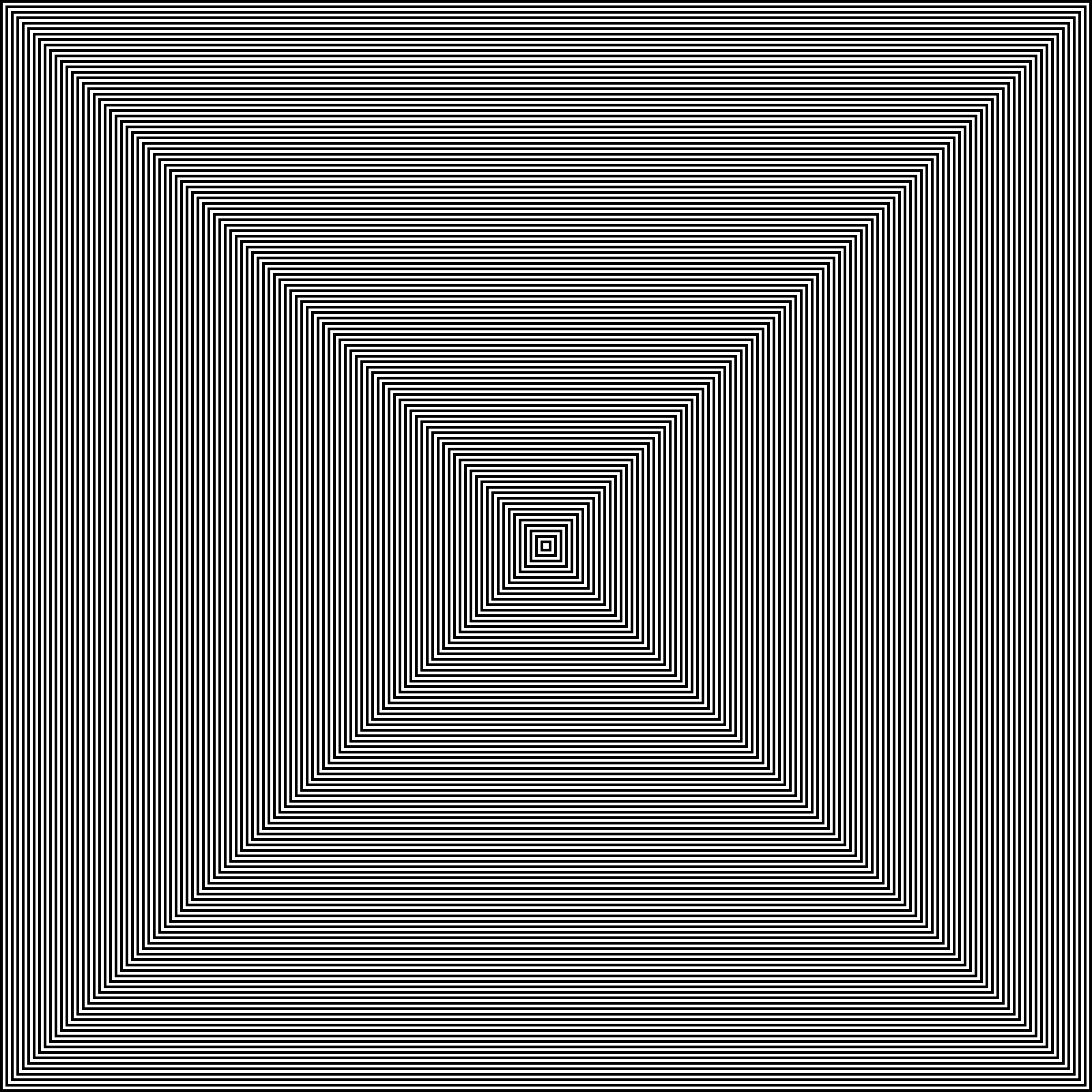
Jak widać, obie funkcje niczym się nie różnią, poza podanymi argumentami. W razie potrzeby można zmodyfikować kod, i dać możliwość zmiany takich parametrów użytkownikowi.

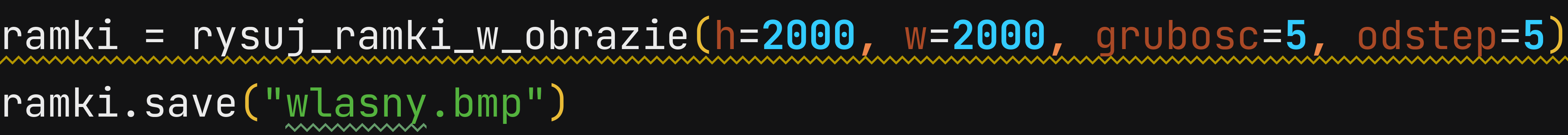
Funkcja ta rysuje czarne pasy na białym tle. Jest możliwość ustawienia odstępu między tymi pasami, jednak domyślnie jest ustawiony na 10, gdyż przy wielkości   
0 wyszedłby po prostu czarny obraz. Przyjęte parametry to:

h=320, w=480, grubość=20, odstęp=20

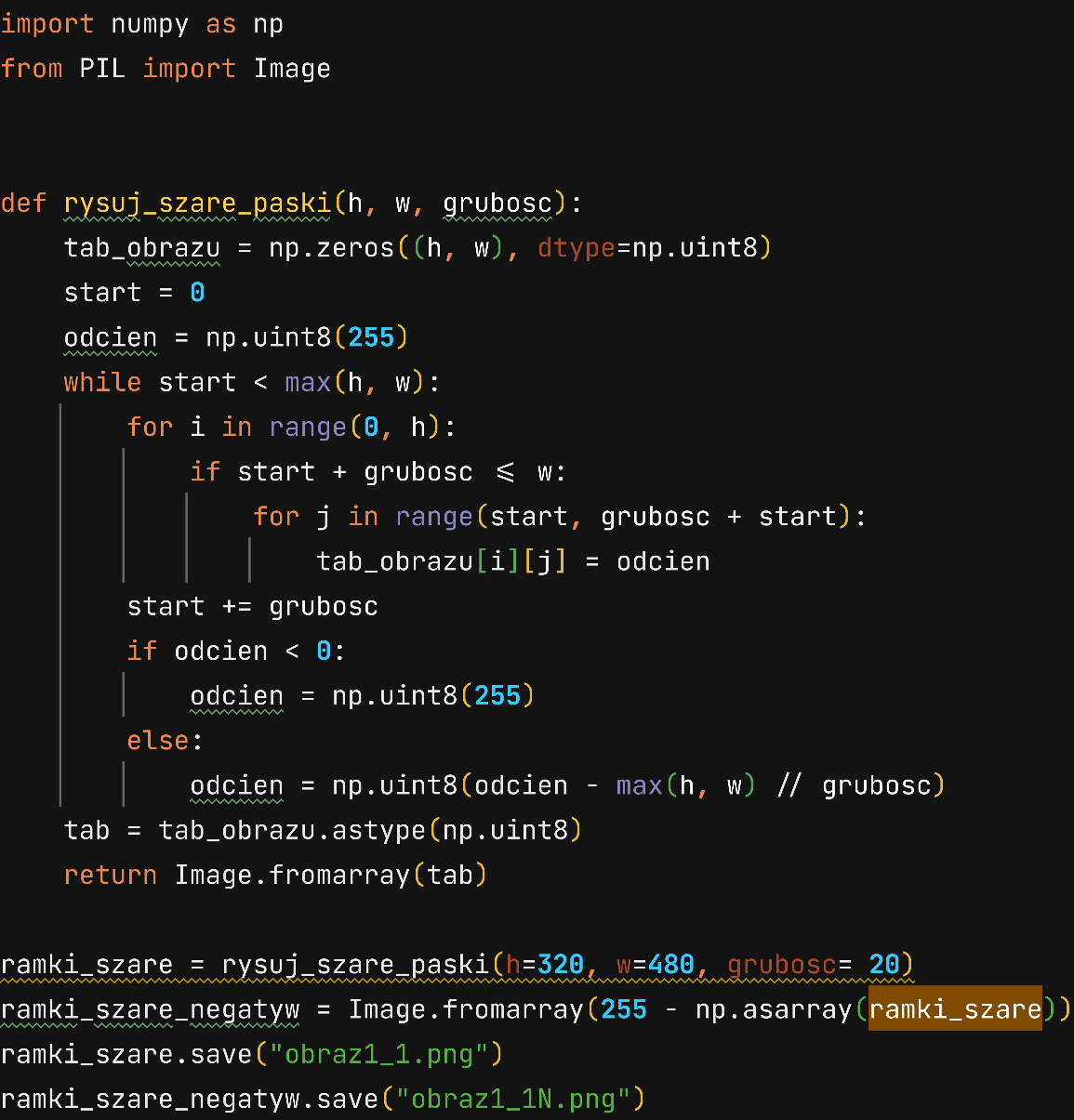


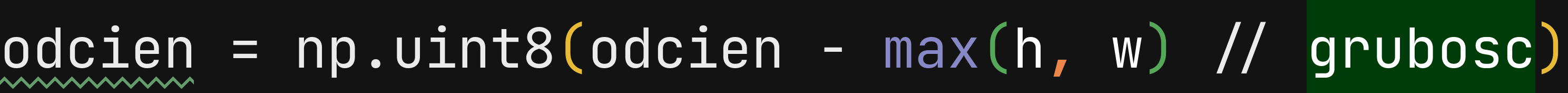
Paski o grubości 20 i odstępie 20

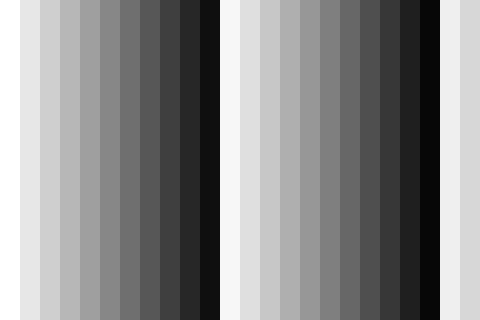
Ta funkcja rysuje dwa czarne prostokąty stykające się w podanym punkcie. Jako parametry przyjęto h=320, w=480 i współrzędne startowe nazwane tu h\_start=50 oraz w\_start=100.Prostokąty stykające się w punkcie 100x50



Obraz o parametrach powyżej, ciekawe złudzenie, jakby były tam różne kolory, a cały obraz to po prostu coraz mniejsze ramki.

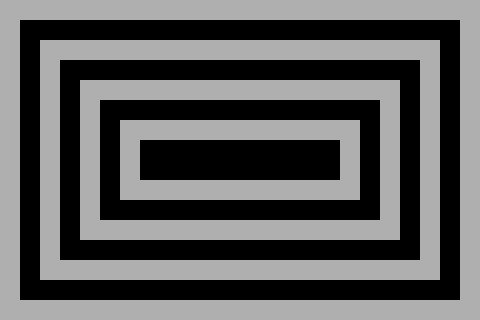
**Obrazy w trybie L**

Funkcja tworzy pasy w odcieniach szarości o wybranej grubości. Na początku przyjęto, że pierwszy pas będzie biały, a kiedy wartość odcienia spadnie poniżej zera zostanie ponownie ustawiony na wartość 255. W innych warunkach wartość odcienia wyliczana będzie według wzoru:

Czyli przy każdym przejściu pętli obecna wartość odcienia będzie pomniejszana o wynik dzielenia bez reszty maksymalnej wartości h lub w podzielonej przez ustaloną grubość. Obraz zawierający zrzut ekranu, czarne, biały, design

Opis wygenerowany automatycznie  
Obraz o parametrach h=320, w=480, grubość=20 Negatyw obrazu obok

* 1. 
  2. Kod rysuje w zależności od parametru odstęp albo ramkę, albo wiele ramek coraz mniejszych oddalonych od siebie o wybrany parametr odstep i o wybranej grubości.
  3. Przykładowy wynik i jego negatyw przy podanych parametrach. Tym razem odcień ustawiono jako stałą wartość i nie zmienia się ona w programie.
  4. 
  5. **Obraz zawierający Prostokąt, sztuka, Symetria, wzór

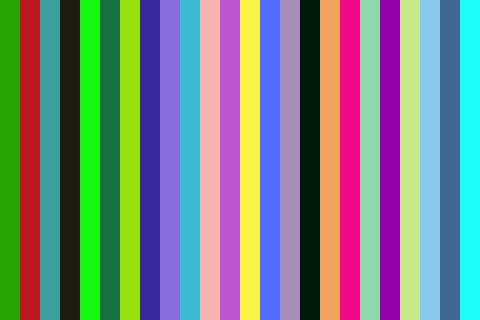
     Opis wygenerowany automatycznie** 
  6. Obraz powstały przy podanych parametrach Negatyw tego obrazu
  7. **Obrazy w trybie RGB**

Obraz zawierający tekst, zrzut ekranu

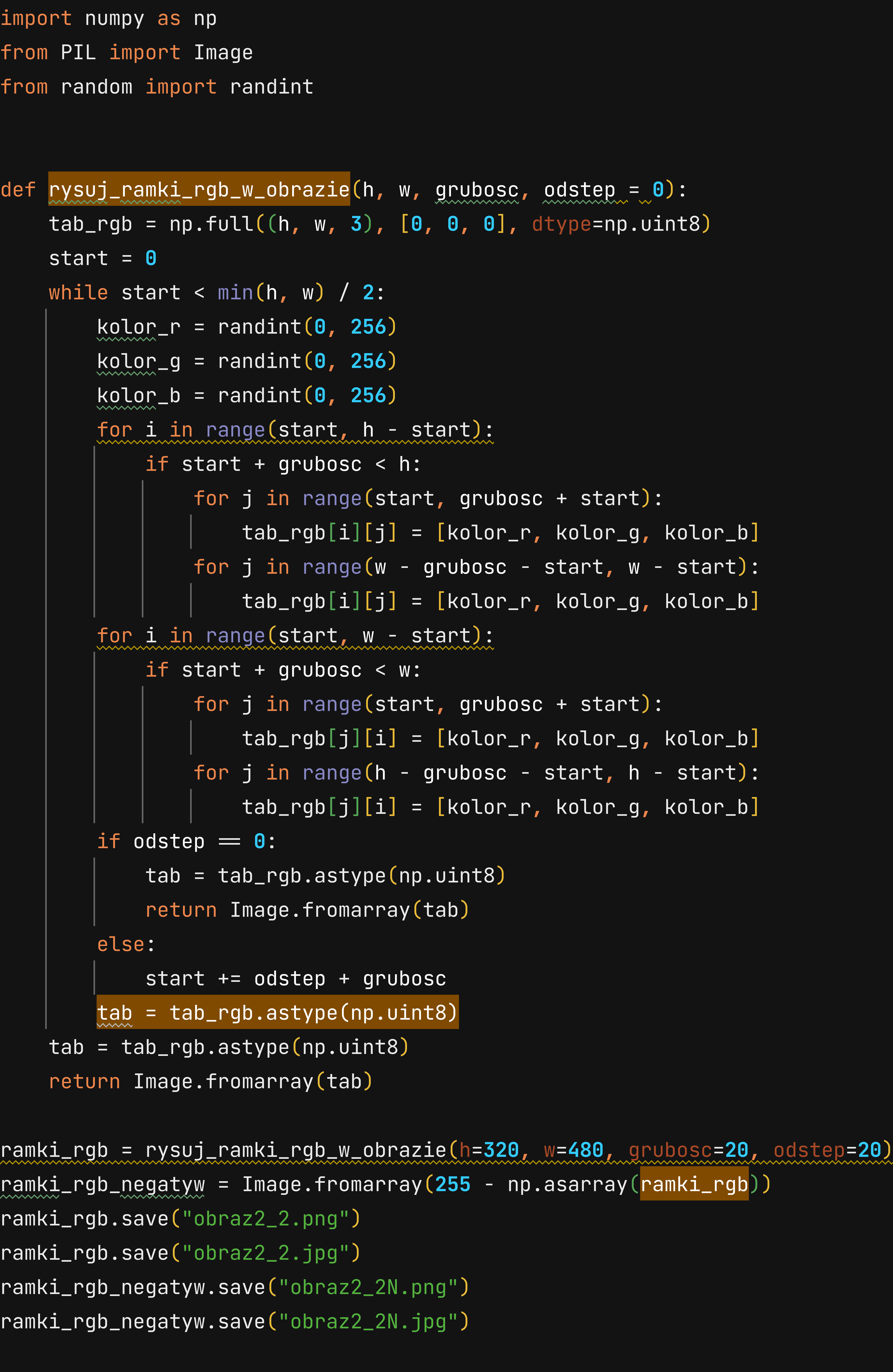
Opis wygenerowany automatycznie

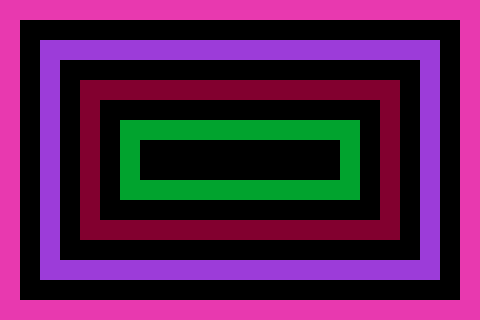
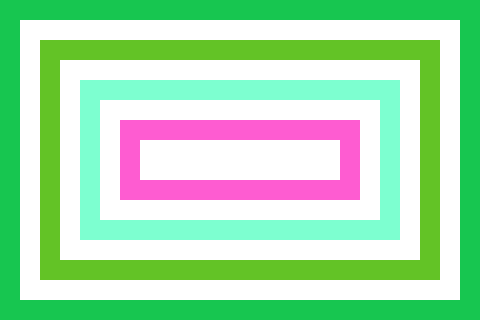
Funkcja podobnie jak rysująca szare pasy rysuje tym razem kolorowe RGB o podanej przez użytkownika grubości.

Aby wybrać zaimportowano funkcję randint z biblioteki random i ustalono aby wybierał losowo z zakresu 0-256. To sprawi, że po narysowaniu danego pasa kolor zmienia się sposób losowy, a każde wywołanie programu daje różne wyniki. Przyjęto następujące parametry: h=320, w=480, grubość=20

Obraz powstały z podanych parametrów. Negatyw tego obrazu



Program rysuje ramkę przy parametrze odstep = 0 lub wiele ramek kiedy odstęp jest > 0. Podobnie jak poprzednio do ustalenia koloru zastosowano funkcję randint z bibliotemi random o parametrach 0-256. W odróżnieniu od odcieni szarości, tutaj każda ramka ma losowany kolor, więc przy każdym uruchomieniu programu wynik będzie inny. 

Obraz powstały przy podanych parametrach Negatyw tego obrazu

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

W powyższym kodzie wczytujemy w pierwszej kolejności znany z poprzednich lab1 plik inicjaly.bmp w formacie uint.8. Następnie przy pomocy funkcji stack z biblioteki NumPy zamieniamy tablicę na format RGB. Nadal mamy tam tylko 0 i 1. Zatem powstała funkcja konwertuj, która porówna wartości w tablicy i zamieni tak, że tam gdzie znajdzie wartości [1, 1, 1] zamieni na [255, 255, 255] czyli na biały. Następnie taką tablicę RGB dajemy jako argument do kolejnej funkcji, która w miejscu gdzie są narysowane inicjały zacznie rysować poziome pasy o ustalonej przez użytkownika grubości.



Obraz jpg Obraz png

Tak, widoczne są różnice między obrazami zapisanymi w formacie JPG i PNG. Główna przyczyna tych różnic wynika   
z różnicy w kompresji obu formatów.

* Format JPEG (JPG) jest formatem stratyowym, co oznacza, że kompresuje obraz przez usuwanie pewnych informacji, co może prowadzić do utraty jakości obrazu. Dlatego obrazy w formacie JPG są zazwyczaj mniejsze pod względem rozmiaru pliku, ale mogą wykazywać kompresję artefaktów, zwłaszcza przy niższych poziomach jakości.
* Format PNG jest formatem bezstratnym, co oznacza, że nie wprowadza kompresji stratowej i zachowuje pełną jakość obrazu. Obrazy w formacie PNG są zazwyczaj większe pod względem rozmiaru pliku, ale zachowują pełne szczegóły i jakość, dlatego są często wybierane do przechowywania obrazów, w których jakość jest ważniejsza niż rozmiar pliku.

Typ uint8 (unsigned 8-bit integer) w Pythonie i innych językach programowania oznacza, że liczby całkowite są reprezentowane w zakresie od 0 do 255. Jeśli podana wartość koloru przekracza 255, to typ uint8 "zawija się" i zaczyna od zera, co nazywa się przepełnieniem. Na przykład, 256 w typie uint8 jest reprezentowane jako 0. Jeśli wartość jest ujemna, typ uint8 nie obsługuje liczb ujemnych i znowu "zawija się", dlatego -1 w typie uint8 jest reprezentowane jako 255. To oznacza, że typ uint8 jest niemożliwy do reprezentowania wartości poza zakresem 0-255 i nie jest odpowiedni do reprezentacji kolorów lub innych danych, które wymagają wartości spoza tego zakresu. O to kod, który sprawdza pewne wartości w typie uint8:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

A to wyniki jakie zostały uzyskane:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie