

do zdobycia: [10pkt]
deadline: 16-17 IV 2014

Zestaw 4

Tym razem przyjrzymy się już problemowi bardziej złożonemu. Zadaniem jest stworzenie programu (docelowo równoległego) służącego do przetwarzania bitmap. Aby nie komplikować sobie życia za bardzo, do obsługi bitmap wykorzystamy istniejącą już bibliotekę, mianowicie bibliotekę **EasyBMP**. Więcej informacji dotyczących tej biblioteki znaleźć można pod adresem:

<http://easybmp.sourceforge.net/>

Niezbędny przy realizacji tego ćwiczenia zasoby (źródła i dokumentację biblioteki **EasyBMP**) pobrać można z:

<http://www.mif.pg.gda.pl/homepages/swinczew/AR/EasyBMP.zip>

Na wstępie nasze zadanie nie będzie specjalnie złożone. Celem jest napisanie programu szeregowego wykonującego proste przetwarzanie bitmapy. Program ten powinien:

1. wczytać wskazany (wektor argumentów) przez użytkownika plik bitmapy,
2. obliczyć średnią luminancję bitmapy, luminancja pojedynczego piksela dana jest wyrażeniem:

$$Y = 0.299R + 0.587G + 0.114B,$$

3. przekonwertować bitmapę na czarno-białą (2-bitową, zachowując rozdzielczość), wynikowy kolor piksela powinien być uzależniony od wyjściowej luminancji piksela, pikselom o luminancji o wartości większej od średniej luminancji powinien zostać przypisany kolor biały, pozostałym pikselom kolor czarny,
4. zapisać bitmapę do pliku o nazwie określonej przez użytkownika (wektor argumentów).

Jako iż program ten będziemy rozwijać przez najbliższe 3-4 tygodnie (będzie on podstawą do realizacji kilku dalszych ćwiczeń), warto od samego początku zadbać o to, by był on napisany porządnie! Warto już teraz wyodrębnić funkcje służące do:

1. odczytu bitmapy z pliku,
2. obliczania średniej wartości składowych koloru R , G i B ,
3. obliczania luminancji pojedynczego piksela,
4. przetwarzania bitmapy,
5. zapisu bitmapy do pliku.

Warto również stworzyć własne typy służące do przechowywania informacji o pojedynczym pikselu, jak i całej bitmapie. Tworząc te typy należy mieć na uwadze fakt, iż w przyszłości pojawi się konieczność komunikowania informacji o bitmapie.

Kryteria oceniania:

1. prawidłowe działanie programu [4pkt],
2. estetyka, przejrzystość, uniwersalność oraz struktura kodu [6pkt].