

## Laboratorium 5A (10pkt)

---

W zadaniu należy przygotować klasę **piksel** opisujący jeden piksel obrazu, oraz klasę **obraz** przedstawiającą obraz złożony z pikseli.

Przygotowany jest plik o nazwie **kolor** zawierający funkcje umożliwiające zmianę koloru tekstu wyświetlanego w konsoli.

Plik programu także jest już przygotowany i **nie można** go modyfikować poza zakomentowywaniem i odkomentowywaniem fragmentów kodu.

### Etap 1 (4.0 pkt)

Klasa **piksel** zawiera wskaźnik na tablicę trzelementową typu `int`, w której będą przechowywane wartości koloru w standardzie **RGB**. Czyli w zerowej komórce tablicy przechowujemy kolor czerwony (R), w pierwszej zielony (G), a w ostatniej niebieski (B). Wartości kolorów powinny się mieścić w przedziale 0 – 255, jednak nie trzeba sprawdzać poprawności podawanych wartości, zakładamy, że wszystkie wartości będą poprawne. Drugie pole klasy przechowuje wyświetlany znak, który będzie reprezentował nasz piksel.

Należy zaimplementować:

- Konstruktor – należy sprawdzać poprawność alokacji pamięci, przechowywany znak to 'O'
- Destruktor – zwalnia zaalokowaną tablicę
- Operator strumienia wyjściowego wypisujący piksel. Podczas wypisywania piksela należy wyświetlić znak w odpowiednim kolorze, a następnie wrócić do koloru domyślnego. (Funkcje z pliku `kolor.h`)  
Ponieważ piksel wyświetlany będzie w konsoli, mamy tam ograniczoną paletę kolorów. Dlatego też użyjemy następującego algorytmu:
  - Wartość koloru w danym kanale zaokrąglamy albo do 0, albo do 255. Jeśli np.  $r < 128$  – przyjmujemy 0, jeśli  $r \geq 128$  – przyjmujemy 255.
  - Używając **switch** (kara za użycie tu ifa)
    - ustawiamy odpowiedni kolor,
    - wyświetlamy znak,
    - ustawiamy kolor domyślny

Kolory:

$r = 0, g = 0, b = 0$  – szary  
 $r = 255, g = 0, b = 0$  – czerwony  
 $r = 0, g = 255, b = 0$  – zielony  
 $r = 0, g = 0, b = 255$  – niebieski  
 $r = 255, g = 255, b = 0$  – żółty  
 $r = 255, g = 0, b = 255$  – fioletowy  
 $r = 0, g = 255, b = 255$  – turkusowy  
 $r = 255, g = 255, b = 255$  – biały

- operator[] – pozwala ustawić wartość koloru w podanym kanale
- operator++ przedrostkowy – zmienia jasność piksela (rozjaśnia) dodając do każdego kanału wartość 1.
- operator-- przyrostkowy – zmienia jasność piksela (przyciemnia) odejmując do każdego kanału wartość 1

### Etap 2 (6.0 pkt)

Klasa **obraz**. Przechowuje rozmiar obrazu, czyli ilość wierszy i kolumn, oraz wskaźnik na dwuwymiarową tablicę wskaźników do pikseli. Proszę tu się przyjrzeć definicji tego pola.

Należy zaimplementować:

- Konstruktor – jeśli któryś z rozmiarów jest zerowy, to nic nie robi. W przeciwnym wypadku tworzy dwuwymiarową tablicę i w każdą komórkę wpisuje dynamicznie utworzony piksel domyślny. Należy też sprawdzić poprawność alokacji.
- Destruktor – zwalnia całą pamięć.
- Operator wypisania – wypisuje piksele wierszami.
- Operator funkcyjny – w podanych indeksach obrazka (pierwsze dwa argumenty) ustawia kolor piksela podany w pozostałych trzech argumentach. Nie tworzy nowego piksela, modyfikuje oryginalny.
- Konstruktor kopiujący.

## Wyniki:

[illegible]