

# KURS JĘZYKA C++

## KOLOROWE PİKSELE

Instytut Informatyki Uniwersytetu Wrocławskiego

*Paweł Rzechonek*

---

---

RGB to jeden z modeli przestrzeni barw, opisywanej współrzędnymi RGB. Jego nazwa powstała ze złożenia pierwszych liter angielskich nazw trzech barw: R – red (czerwonej), G – green (zielonej) i B – blue (niebieskiej). Jest to model wynikający z właściwości sensorycznych ludzkiego oka, w którym wrażenie widzenia dowolnego koloru można wywołać przez zmieszanie w ustalonych proporcjach trzech wiązek światła o barwie czerwonej, zielonej i niebieskiej.

### **Zadanie.**

Zdefiniuj klasę `kolor` reprezentującą kolor zapisany w systemie RGB. W klasie tej umieść trzy pola, które będą pamiętać nasycenie daną barwą (wartości z zakresu  $0 \dots 255$ ). Zdefiniuj dwakonstruktory: jeden bezargumentowy, który będzie ustawiać kolor na czarny (same zera) oraz drugi, który zainicjalizuje wartości poszczególnych składowych koloru podanymi wartościami (w konstruktorze tym zgłaszaj wyjątki, gdy któraś z podanych wartości wykroczy poza zakres  $0 \dots 255$ ). Klasa ma też posiadać gettery i settery odpowiednio do odczytania i zapisania określonej barwy składowej, a także metody do rozjaśniania i przyciemniania całego koloru. Dodatkowo w klasie umieść metodę statyczną do łączenia kolorów (średnia arytmetyczna poszczególnych barw).

Następnie zdefiniuj klasę `kolortransparentny`, która uzupełni klasę `kolor` o współczynnik alfa (wartość z zakresu  $0 \dots 255$ ), określający przezroczystość (0 oznacza całkowitą przezroczystość a 255 pełny kolor). Zdefiniuj także klasę `kolornazwany`, która doloży do klasy `kolor` nazwę mnemoniczną (łańcuch znakowy typu `string`) — domyślnie nazwa ma być pusta, ale jeśli już zostanie podana, to ma się składać tylko z małych liter alfabetu angielskiego. Obie te klasy (`kolortransparentny` oraz `kolornazwany`) mają dziedziczyć publicznie po klasie `kolor`.

Kolejna klasa `kolornt` to klasa, która ma posiadać zarówno współczynnik alfa jak również nazwę mnemoniczną. W tym przypadku zastosuj dziedziczenie wielobazowe po klasach `kolortransparentny` i `kolornazwany`; zadbaj o to, by w klasie tej nie powtarzały się informacje dotyczące barw składowych koloru (pochodzące z klasy `kolor`).

Dalej zdefiniuj klasę `piksel`, reprezentującą pojedynczy piksel na ekranie; piksel ma być zdefiniowany na ekranie za pomocą współrzędnych `x` i `y` typu `int`. Przy inicjalizacji i przekształceniach sprawdzaj, czy współrzędne piksela nie wychodzą poza ekran (muszą być mniejsze od rozmiaru ekranu i nieujemne). Rozmiar ekranu określ za pomocą pól statycznych w klasie `piksel` (ustal na stałe pewien wybrany standard rozdzielczości, na przykład SVGA, XGA, SXGA+ dla proporcji 4:3, albo HDTV720, HDTV1080 dla proporcji 16:9). Dodatkowo w klasie umieść składowe funkcje instancyjne do obliczenia odległości piksela od lewego, prawego, górnego i dolnego brzegu ekranu.

Następna klasa `pikselkolorowy` ma rozszerzać klasę `piksel` o możliwość ustalenia koloru piksela. Właściwość koloru dla piksela określ za pomocą typu `kolortransparentny`. Dodaj w tej klasie funkcjonalność związaną z przemieszczaniem pikseli o zadany wektor (gdyby nowe współrzędne miały wyjść poza ekran, to należy zgłosić wyjątek).

Zdefiniuj także funkcję globalną liczącą odległość pomiędzy pikselami (wynik zaokrąglony do najbliższej liczby całkowitej) przekazanymi przez referencje/wskaźniki:

```
int odleglosc(const piksel &p, const piksel &q);  
int odleglosc(const piksel *p, const piksel *q);
```

Na koniec napisz program, który rzetelnie przetestuje wszystkie zdefiniowane klasy. Bardzo przydatne przy testowaniu klas mogą okazać się operatory strumieniowe.

### **Elementy w programie, na które należy zwracać uwagę.**

- Podział programu na pliki nagłówkowe (osobno z definicjami klas dla kolorów i osobno z definicjami klas dla pikseli) i źródłowe (osobny plik źródłowy z funkcją `main.cpp` testującą klasy kolorów i pikseli).
- Implementacja dziedziczenia, w tym dziedziczenia wielobazowego i wirtualnego.
- W funkcjach składowych i w konstruktorach zgłaszaj błędy za pomocą instrukcji `throw`.
- Wykorzystanie automatycznego rzutowania wskaźników/referencji do klas pochodnych na wskaźniki/referencje do klas bazowych w wywołaniu funkcji `odleglosc()`.
- W funkcji `main()` należy rzetelnie przetestować wszystkie zdefiniowane klasy.