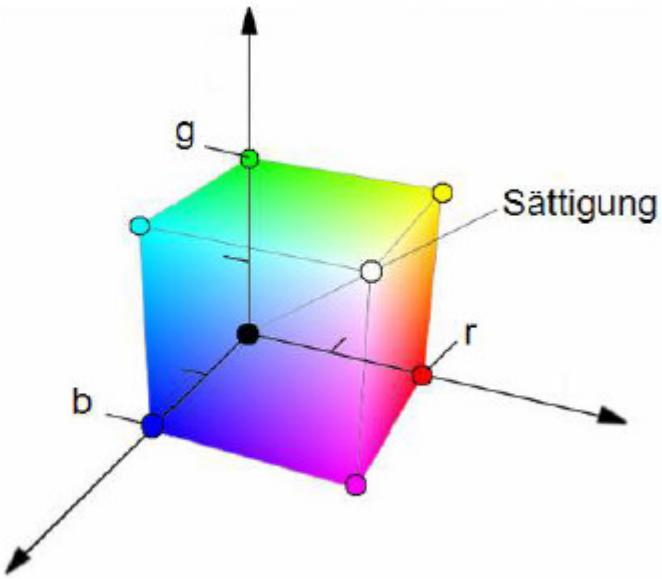
Freiwillige Offline-Aufgabe 10-01 (INF & WI & MCD): Klasse `RGB_Color`

(geübte C++ Konstrukte: Klassen, Attribute, Methoden)

Der *RGB-Farbraum* ist ein additiver Farbraum, der Farbwahrnehmungen durch das additive Mischen dreier Grundfarben (Rot, Grün und Blau) nachbildet.

Jede Farbe kann aus drei Grundfarben gemischt werden, z.B. aus rot, grün und blau. Diese Situation kann mithilfe des sogenannten RGB-Einheitswürfels (R = rot, G = grün, B = blau) graphisch dargestellt werden:



Im RGB Farbraum hat Weiß z.B. die Koordinaten (1,1,1), schwarz (0,0,0) und blau (0,0,1).

Beispielsweise ist Gelb ein Mix aus gleichen Rot- und Grünanteilen, wohingegen es sich bei Cyan (= Türkis) um einen Mix aus gleichen Grün- und Blauanteilen handelt. Auf der Geraden durch die beiden Ecken, die Schwarz und Weiß darstellen, liegen alle Graustufen beginnend bei Schwarz, Dunkelgrau bis Hellgrau und Weiß.

In einem Programm kann eine Farbe mithilfe von Koordinaten `(r, g, b)` dargestellt werden, wobei `0 <= r, g, b <= 255`. *D.h. hier ist der Wertebereich der Farbanteile nicht von `0` bis `1`, sondern von `0` bis `255`.*

Beispielsweise sind die Koordinaten von Gelb `(255, 255, 0)`, die von Cyan `(0, 255, 255)`, die von Schwarz `(0, 0, 0)`, die von Blau `(0, 0, 255)` und die von Weiß `(255, 255, 255)`.

Definieren Sie in einer Headerdatei `RGB_Color.h` die C++-Klasse `RGB_Color` zur Darstellung einer Farbe. Folgende Attribute und Methoden sind zu definieren:

- Die Attribute `red`, `green`, `blue` für die Farbanteile der aktuellen Farbe, jeweils mit dem Datentyp `int`. Diese Attribute dürfen von außen *nicht* zugreifbar sein.
- Ein Default-Konstruktor, der ein Objekt vom Typ `RGB_Color` mit Weiß initialisiert,
- Ein Konstruktor mit drei Parametern für den Rot-, Grün- und Blauanteil einer Farbe, der den Default-Konstruktor überlädt.
- Die Methode `set_color()` mit drei Parametern für den Rot-, Grün- und Blauanteil einer Farbe.
- Die Getter-Methoden `get_red()`, `get_green()` und `get_blue()`, die den Rot-, Grün- bzw. Blauanteil einer Farbe zurückliefern.
- Die entsprechenden Setter-Methoden. Die zu setzenden Werte sind auf den Wertebereich `0 … 255` zu überprüfen und nur im Falle ihrer Gültigkeit zu setzen (anderenfalls keine Änderung, aber auch keine Meldung).
- `display()` zur Ausgabe der Rot-, Grün- und Blauanteile einer Farbe im Format `(r, g, b)` auf dem Bildschirm. Die Methode hat keinen Parameter und keinen Return-Wert.
  Beispiel: `rot: 0 gruen: 255 blau: 255`
- `input_color()`, die den Benutzer auffordert, den Rot-, Grün- und Blauanteil einer Farbe einzugeben. Die Eingaben sind auf den Wertebereich `0 … 255` zu überprüfen. Nur, wenn korrekte Werte eingegeben wurden, werden die eingegebenen Farbanteil-Werte gespeichert.
  Bitte beachten Sie diese Anforderung genau: Auch wenn (erst) der zweite oder dritte eingegebene Farbanteil falsch eingegeben wird,

darf sich keiner der vorherigen Farbanteile geändert haben.
Die Methode gibt `true` zurück, falls die Eingabe gültig ist, andernfalls `false`. Die Methode selbst gibt bei falscher Eingabe keine Meldung auf den Bildschirm aus; der Text `Falsche Eingabe!` aus dem Testlauf wird vom Hauptprogramm `main()` ausgegeben.
Beispiel (Eingaben unterstrichen):

```
rot (0...255): ? 209
gruen (0...255): ? 2
blau (0...255): ? 66
```

Implementieren Sie die umfassenderen Methoden der Klasse in einer `RGB_Color.cpp` Datei.

Schreiben Sie in einer dritten Datei `main.cpp` ein Hauptprogramm `main()` zum Testen der Klasse `RGB_Color`. Definieren Sie zu dem Zweck vier Objekte, die die Farben Weiß, Cyan `(r=0, g=255, b=255)`, Gelb `(r=255, g=255, b=0)` und Magenta `(r=255, g=0, b=255)` darstellen. Geben Sie den Rot-, Grün- und Blauanteil der Farben aus.

Definieren Sie ein Objekt `farbe1` mit den Farbanteilen `(r=255, g=255, b=128)`. Definieren Sie dann ein Objekt `farbe2`; dieses soll mittels des Farbobjekts `farbe1` initialisiert werden. Geben Sie zur Prüfung die RGB-Anteile von `farbe2` aus.

Lassen Sie dann die Farbanteile von `farbe2` durch den Benutzer mittels `input_color()` neu setzen. Geben Sie zur Prüfung die RGB-Anteile von `farbe2` erneut aus.

Testläufe (Benutzereingaben zur Verdeutlichung unterstrichen):

```
Weiss:
rot: 255 gruen: 255 blau: 255
Tuerkis:
rot: 0 gruen: 255 blau: 255
Gelb:
rot: 255 gruen: 255 blau: 0
Magenta:
rot: 255 gruen: 0 blau: 255

Farbe 1:
rot: 255 gruen: 255 blau: 128

Farbe 2:
rot: 255 gruen: 255 blau: 128
```

```
Neu-Eingabe von Farbe 2:
rot (0...255): ? 33
gruen (0...255): ? 44
blau (0...255): ? 55

Farbe 2 jetzt:
rot: 33 gruen: 44 blau: 55
Drücken Sie eine beliebige Taste . . .
```

```
Weiss:
rot: 255 gruen: 255 blau: 255
Tuerkis:
rot: 0 gruen: 255 blau: 255
Gelb:
rot: 255 gruen: 255 blau: 0
Magenta:
rot: 255 gruen: 0 blau: 255

Farbe 1:
rot: 255 gruen: 255 blau: 128

Farbe 2:
rot: 255 gruen: 255 blau: 128

Neu-Eingabe von Farbe 2:
rot (0...255): ? 33
gruen (0...255): ? 999
Falsche Eingabe!
Drücken Sie eine beliebige Taste . . .
```
