



RGB-Farbmodell

Die Farbe eines Pixels kann man auf verschiedene Arten codieren. Eine davon ist das RGB-Farbmodell¹ (Rot-Grün-Blau):

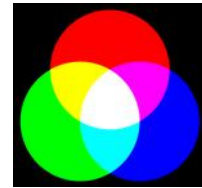


Abb. 1: Additive Farbmischung

Aufgabe:

1. Ergänze die folgende Tabelle:



Nutze dazu auch https://informatik.schule.de/rgb/RGB_farbmischer.html.

Farbe		Rot	Grün	Lila	Braun	Hellgrau	Dunkelgrau	Weiß	schwarz
Rotanteil	0-255	255							
	0.0-1.0	1.0							
Grünanteil	0-255	0							
	0.0-1.0	0.0							
Blauanteil	0-255	0							
	0.0-1.0	0.0							
Hexcode		FF0000							

2. Beschreibe, wie du eine Farbe heller machen kannst, ohne den Farbton zu ändern.



3. Beschreibe, was passiert, wenn du die einzelnen Farbanteile einer Farbe allmählich aneinander annäherst.



Farben in Java

Klasse Color

Farben werden in Objekten der Klasse `Color` gespeichert:

Farbe deklarieren: `Color farbe;`

Farbe initialisieren: `farbe = new Color(0,255,255); // Cyan`
`farbe = Color.GREEN; // geht für einige Farben`

Farbanteil ermitteln: `rotanteil = farbe.getRed(); // getBlue(),getGreen()`

Man kann einzelne Farbanteile einer Farbe nicht ändern, nur eine komplette neue Farbe initialisieren.

Dokumentation

<https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/awt/Color.html>

Aufgabe

4. Deklariere zwei Variablen `f1` und `f2` vom Typ `Color` in Java. Initialisiere sie auf zwei verschiedene Arten, so dass sie die Farbe Lila repräsentieren.



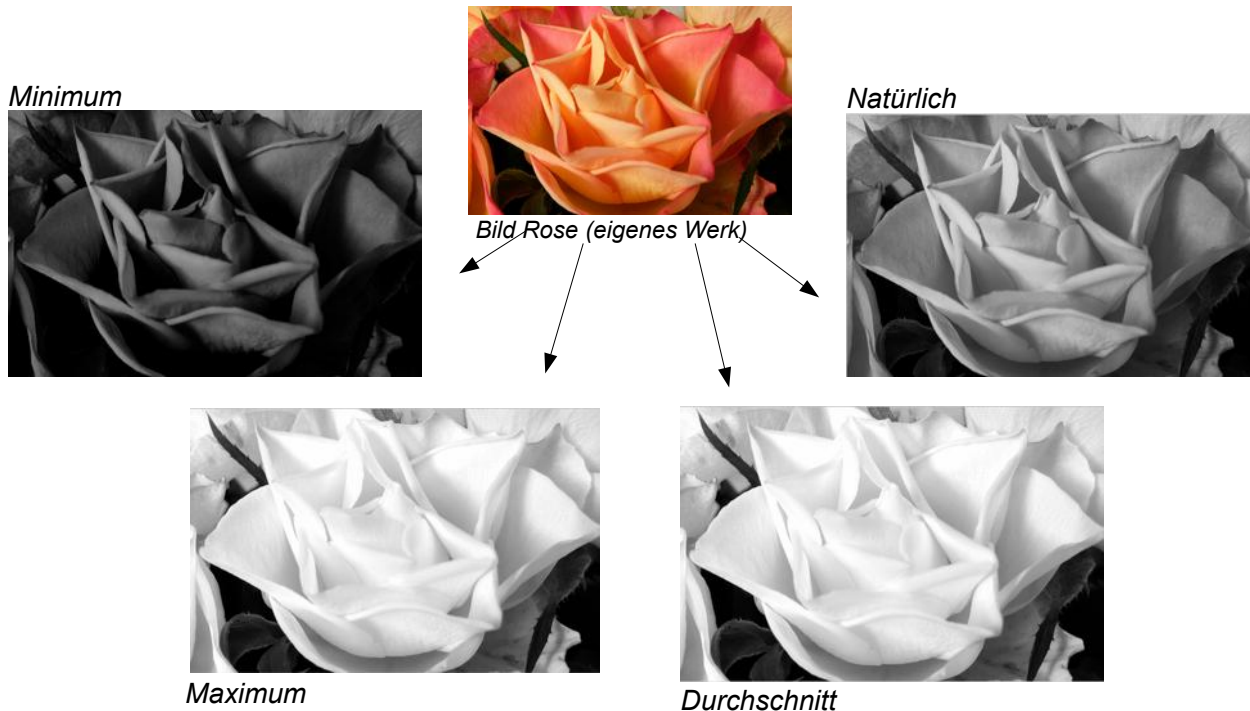
5. Gib die notwendigen Befehle an, um `f1` eine neue Farbe zuzuweisen, deren Blauanteil auf 0 und deren Grünanteil auf 128 gesetzt wird. Der Rotanteil soll dem bisherigen Rotanteil von `f1` entsprechen. Gib an, welche Farbe nun repräsentiert wird.



¹ Abbildung 1: Additive Farbmischung, Quark67 (Lizenz: CC BY-SA 3.0) via Wikimedia Commons, URL: <https://commons.wikimedia.org/wiki/File:Synthese%2B.svg>.



Bildbearbeitungsalgorithmen 2: Umwandlung in Graustufenbilder



1. Erzeuge analog zu der Klasse `GeometrischeBildoperationen` eine Klasse



`Pixeloperationen`.

Implementiere die Methode `public Picture graustufenDurchschnitt(Picture originalbild)`, die für jedes Pixel des Originalbildes den Mittelwert der Farbanteile berechnet und diesen als Helligkeit für die Graustufe nimmt.

Gehe dabei wie bei den Algorithmen zum Spiegeln vor:

- Deklariere und initialisiere zweidimensionale `Color`-Arrays für das Originalbild und das neue Bild.
- Mit Hilfe von zwei `for`-Schleifen wird für jedes Pixel Folgendes gemacht:
 - Hole Rot-, Grün- und Blauanteil aus dem Originalpixel heraus und speichere die Werte in lokalen Variablen.
 - Berechne den Durchschnitt der drei Werte.
 - Erzeuge ein neues `Color`-Objekt, das einen Grauwert repräsentiert, dessen Helligkeit dem berechneten Durchschnitt entspricht.
 - Speichere diese im `Pixel-Array` des neuen Bildes.
- Erstelle aus dem `Pixel-Array` ein neues Bild und gib es zurück.

2. Implementiere analog zu `graustufenDurchschnitt` zwei weitere Methoden



`graustufenMin` und `graustufenMax`, die das Minimum, bzw. Maximum der drei Farbanteile als Helligkeit der Graustufe nehmen.

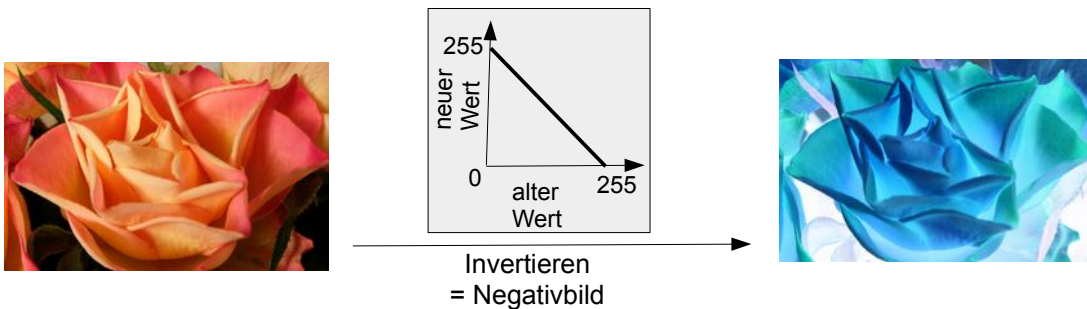
3. Unser Auge ist nicht für alle drei Farben gleich empfindlich. Daher sollten die Farbanteile



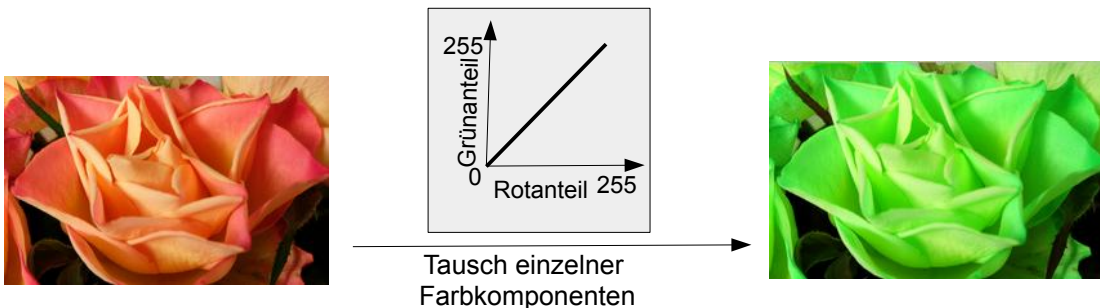
unterschiedlich gewichtet werden: Rot mit 29,9%, Grün mit 58,7% und Blau mit 11,4%. Implementiere eine Methode `graustufenNatuerlich`, die diese Vorgabe umsetzt.



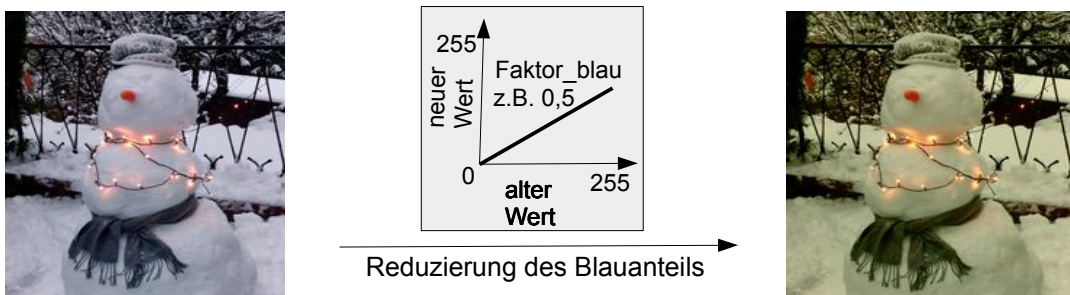
Bildbearbeitungsalgorithmen 3: RGB-Experimente



1. Implementiere eine Methode `invertiere`, die alle Farbanteile umdreht, d.h. aus Rotanteil 0 wird Rotanteil 255 und umgekehrt. Man erhält ein Bild, das dem Negativbild entspricht, das bei der analogen Fotografie bei der Belichtung des Films entsteht.



2. Implementiere eine Methode `tauschRotGruen`, die den Rot- und den Grünanteil jeder Farbe vertauscht. Analog kann man auch andere Farbkomponenten tauschen.



3. Implementiere eine Methode `public Picture farbaenderung (Picture originalbild, double faktor_r, double faktor_g, double faktor_b)`, die jede Farbkomponente mit dem angegebenen Faktor multipliziert.

Beachte dabei, dass durch die Multiplikation mit Faktoren größer als 1 oder kleiner als 0, Werte herauskommen können, die nicht zum gültigen Wertebereich $[0;255]$ gehören. Werte größer als 255 müssen auf 255 reduziert werden, Werte kleiner 0 müssen zu 0 werden.

4. Beschreibe, welchen Effekt es hat, wenn man die Methode mit `farbaenderung(beispiel1, 0.0, 1.0, 1.0)` aufruft. Was passiert bei `farbaenderung(beispiel1, 0.5, 0.5, 0.5)`?
5. Gib an, wie die Methode aufgerufen werden muss, um nur den Blauanteil eines Bildes zu bekommen. Gib an, wie die Methode aufgerufen werden muss, um ein Bild aufzuhellen.