

Übungsblatt 14 JavaFx

Digitale Bildverarbeitung

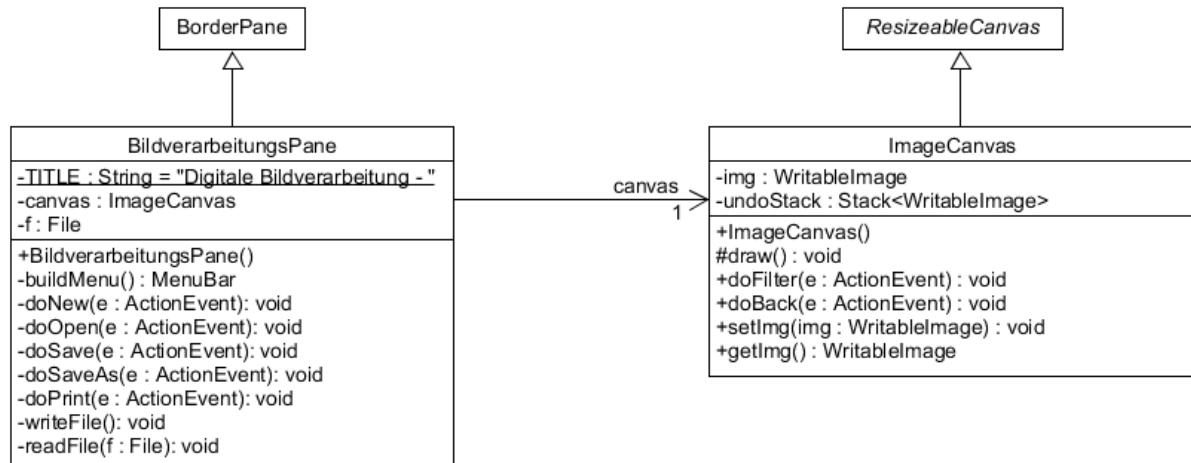
Es soll ein Programm zur Anzeige und zur Verarbeitung digitaler Bilder entworfen werden.



Die Menüeinträge haben folgendes Aussehen:

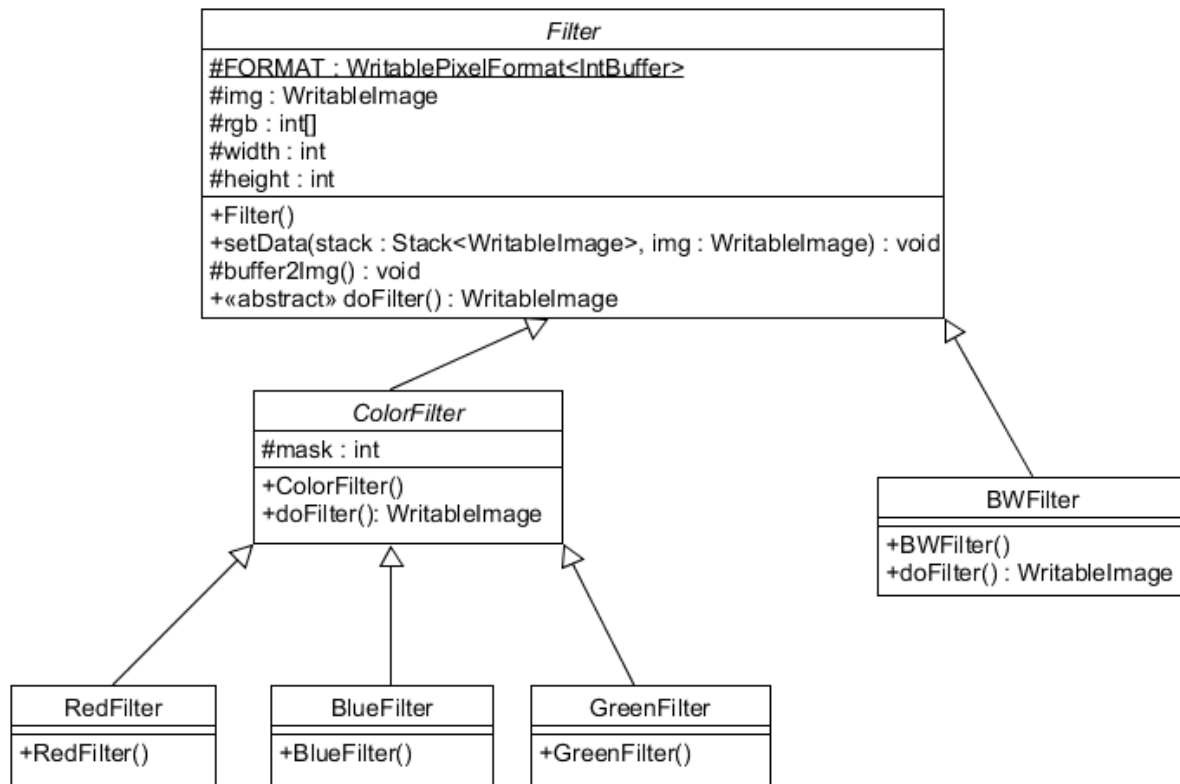
Datei		Bearbeiten	Filter
Neu	Ctrl+N		Rot
Öffnen	Ctrl+O		Grün
Sichern	Ctrl+S		Blau
Sichern unter...			Negativ
Drucken	Ctrl+P		Graustufen
Beenden	Ctrl+Q		Sepia
			Horizontal spiegeln
			Vertical spiegeln
			90° im Uhrzeigersinn drehen
			Weichzeichner
			Scharfzeichner
			SobelX
			SobelY
			Kantendetektor
		Zurück	
		Ctrl+Z	

Klassendiagramm:



- Implementieren Sie, die in den Klassendiagrammen gezeigte Struktur.
- Ergänzen Sie die Menüeinträge wie aufgeführt. Legen Sie ein zweidimensionales statisches Array von Strings als Konstante mit Namen „TXT“ in der Klasse BildverarbeitungsPane an. In „TXT“ werden die Menüeinträge des Filtermenüs und die zugehörigen Filternamen verwaltet.
- Wählt der Benutzer den Menüeintrag Bearbeiten→ Zurück sollte der vorherige Zustand wieder hergestellt werden. Auf dem undoStack werden nahezu beliebig viele (bis der Arbeitsspeicher voll ist) Bilder gespeichert.
- Bei der Auswahl der Farbfiler soll das Bild in RGB-Ebenen zerlegt werden. Im Array `rgb[]` besteht Zugriff auf die einzelnen Bildpunkte. `0xFF7C3467` zeigt die Hexdarstellung eines Pixels. Die ersten zwei Ziffern stellen den Alphakanal dar. `7C` ist die Rotkomponente, `34` die Grünkomponente, `67` die Blaukomponente. Für den RotFilter werden demnach alle anderen Komponenten durch eine Bitweise-Und-Verknüpfung ausgeblendet.
- Zur Berechnung des Negativbildes werden die Komplementärfarben durch die Differenz zu 255 bestimmt. Anschließend werden die Farbkomponenten Bitweise-Oder wieder zusammen gesetzt.
- Für die Umrechnung in Graustufen stehen unterschiedliche Algorithmen zur Wahl:
 Average: $val = (r+g+b)/3$
 Lightness: $val = (\max(r, g, b) + \min(r, g, b)) / 2$
 Luminosity: $val = 0.21 * r + 0.71 * g + 0.07 * b$
 Experimentieren Sie.
- Finden Sie im Internet einen Algorithmus für die Umrechnung in Sepia.
- Ein einfacher Weichzeichner- oder Blur-Filter summiert alle 4 Nachbapixel und den aktuellen Bildpunkt und teilt dann durch 5. (RGB Komponenten separat bearbeiten)
- Ein einfacher Scharfzeichnerfilter subtrahiert alle 4 Nachbarn vom 5 fachen aktuellen Bildpunkt.

Die Klassen SepiaFilter, BlurFilter, SharpenFilter, InvertFilter können sinngemäß ergänzt werden.



Weitere Filter zur Kantendetektion können bei Bedarf ergänzt werden. (siehe Sobel-Operator):

$$H_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \text{ oder } H_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}$$