

# Prüfung 3: Processing – Lösungen

## Aufgabe 1: Bildbearbeitungsprogramm lesen (5 Punkte)

Sie finden in einer Sammlung von Bildbearbeitungseffekten folgendes Processing-Programm – wie so oft in der Praxis gänzlich frei von Kommentaren:

```
public void setup() {
    original = loadImage("flickr photos mike 6083251958.jpg");
    original.loadPixels();

    bearbeitet = new PImage(original.width, original.height);
    bearbeitet.loadPixels();

    for (int y = 0; y < bearbeitet.height; y++) {
        for (int x = 0; x < bearbeitet.width; x++) {
            if (istUngerade(y)) {
                int originalFarbe = getColor(original, x, y);
                setColor(bearbeitet, x, y, originalFarbe);
            } else {
                int schwarz = color(0, 0, 0);
                setColor(bearbeitet, x, y, schwarz);
            }
        }
    }
    bearbeitet.updatePixels();

    size(original.width + bearbeitet.width, bearbeitet.height);
}

boolean istUngerade(int i) {
    return i % 2 == 1;
}
```

- Der Einfachheit halber ist oben nur die Methode `setup()` dargestellt, die das Original-Bild einliest und das neue Bild berechnet.
- Die oben ebenfalls dargestellte Methode `istUngerade(i)` gibt `true` zurück, falls `i` eine ungerade Zahl ist.
- Die Methode `draw()` – hier nicht gezeigt – stellt links das Originalbild und rechts das in `setup()` berechnete Bild dar.
- Die Methode `getColor(bild, x, y)` liefert wie im Skript den Farbwert im Bild `bild` an der Koordinate `(x, y)`.
- Die Methode `setColor(bild, x, y, farbe)` setzt den Farbwert `farbe` im Bild `bild` an Koordinate `(x, y)`.

### Aufgabe 1.1: Programm simulieren (2 Punkte)

Um das Programm zu analysieren, spielen Sie es von Hand auf Papier durch.

Der Einfachheit halber verwendet das Bild nur schwarz (s) und weiss (w) als Farben. Das Eingabebild für die Analyse ist 4x3 Pixel gross:

s	w	w	s
w	s	s	w
w	w	w	s

Zeichnen Sie unten das resultierende Bild hin, in gleicher Art wie das Originalbild links: Jedes Kästchen ist ein Pixel:

### Aufgabe 1.2: Programm beschreiben (2 Punkte)

Beschreiben Sie in wenigen Worten präzise den Bildbearbeitungseffekt, den das Programm darstellt: Was geschieht mit welchen Bildpunkten?

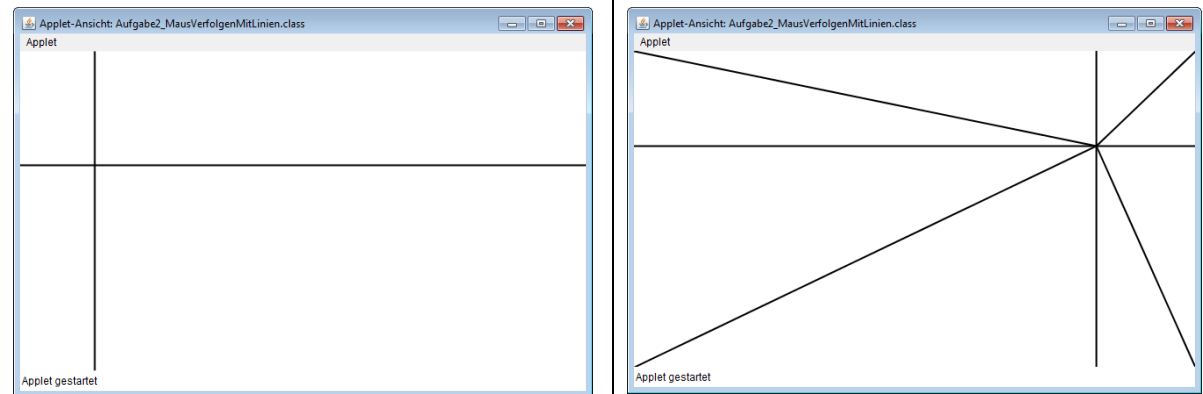
### Aufgabe 1.3: Programm anpassen (1 Punkt)

Das Programm verändert die Farben des Bildes in einer Richtung (horizontal oder vertikal).

1. Markieren Sie auf dem Aufgabenblatt im obigen Programm deutlich, welche Programmzeilen Sie ändern müssen, damit das Programm den gleichen Bildbearbeitungseffekt auf die andere Richtung (also vertikal oder horizontal) darstellen würde.
2. Schreiben Sie anschliessend auf, wie diese (und nur diese) Programmzeilen neu lauten müssen (Sie brauchen nicht das ganze Programm abzuschreiben):

Schreiben Sie ein Programm, das die Maus mit einem "Fadenkreuz" verfolgt:

- |                                                                                                                                        |                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Das Programm soll die unten abgebildeten Linien darstellen. Im Screenshot befindet sich die Maus am Schnittpunkt der beiden Linien: | 2. Wenn der Benutzer eine Maustaste drückt, sollen zusätzlich die unten abgebildeten Linien dargestellt werden, von der Mausposition zu den Ecken hin: |
|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|



Aufgabe 2.1 Programm schreiben (4 Punkte)

Der Anfang ist bereits gemacht – ergänzen Sie die fehlenden Befehle, damit sich das Programm wie beschrieben verhält:

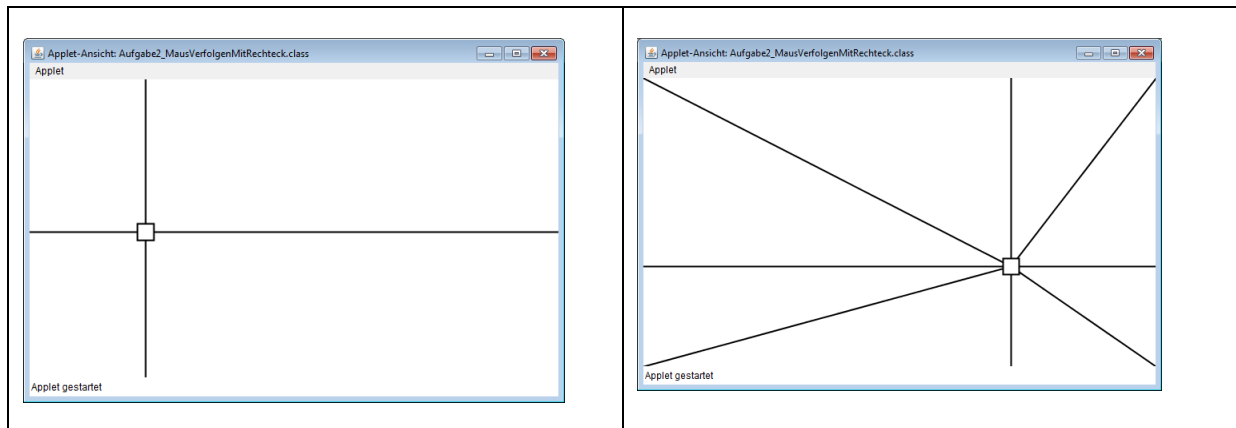
[illegible]

Zur Erinnerung:

- Die Variablen width, height geben die Breite und Höhe der Zeichnungsfläche an.
- Die Variablen mouseX, mouseY geben die aktuelle Mauskoordinate (x, y) an.
- Die Variable mousePressed wird auf true gesetzt, wenn Benutzer Maustaste drückt.

## Aufgabe 2.2 Programm ergänzen (1 Punkt)

Erweitern Sie das Programm so, dass die Methode `draw()` über die Linien noch ein Quadrat der Breite 20 zeichnet, dessen Mittelpunkt die Mausposition  $(x, y)$  ist. (Sie können für die Berechnung der Koordinaten des Quadrats ignorieren, dass dessen Koordinaten ausserhalb des sichtbaren Bereichs zu liegen kommen, wenn Sie mit der Maus an den Rand des Fensters gehen: Processing ignoriert das auch und stellt einfach nur den sichtbaren Teil des Quadrates dar.)



1. Markieren Sie auf im obigen Programm deutlich, wo Sie das Programm ergänzen müssen.
2. Schreiben Sie anschliessend den zu ergänzenden Programmteil auf:

# Lösungen

## Aufgabe 1: Bildbearbeitungsprogramm lesen (5 Punkte)

### Aufgabe 1.1: Programm simulieren (2 Punkte)

2 Punkte, falls alle vier Spalten korrekt; 1 Punkt, wenn mind. 1 Spalte schwarz gefärbt wurde

Der Einfachheit halber verwendet das Bild nur schwarz (s) und weiss (w) als Farben. Das Eingabebild für die Analyse sei 3x3 Pixel gross:

s	w	w	s
w	s	s	w
w	w	w	s

Zeichnen Sie unten das resultierende Bild hin, in gleicher Art wie das Originalbild links: Jedes Kästchen ist ein Pixel. Sie können ebenfalls s und w als Abkürzung für die Farben verwenden:

s	w	w	s
s	s	s	w
w	w	w	s

### Aufgabe 1.2: Programm beschreiben (2 Punkte)

Mögliche Lösung: 1 Punkt für die ungeraden Spalten; 1 Punkt für die geraden Spalten

Das Programm setzt die Farben der Pixel aller ungeraden Spalten auf Schwarz. Die Farben der Pixel der geraden Spalten bleiben unverändert.

("Bonuspunkt" für: Es resultiert eine Verdunkelung des Bildes.)

### Aufgabe 1.3: Programm anpassen (1 Punkt)

**Original:**

```
if (istUngerade(y)) {
```

**Angepasst:**

```
if (istUngerade(x)) {
```

## Aufgabe 2: Koordinaten, Formen, Interaktionen mit der Maus (5 Punkte)

Mögliche Lösung [4 Punkte – 1 Punkt für Fadenkreuz, 1 Punkt für if korrekt, 2 Punkte für Eckverbindungen ]

```
public void draw() {  
    background(255);  
    strokeWeight(2 + speed / 10);  
    stroke(0);  
  
    line(0, mouseY, width, mouseY);  
    line(mouseX, 0, mouseX, height);  
  
    if (mousePressed) {  
        line(0, 0, mouseX, mouseY);  
        line(width, 0, mouseX, mouseY);  
        line(0, height, mouseX, mouseY);  
        line(width, height, mouseX, mouseY);  
    }  
  
    // Hier die Ergänzung von Aufgabe 2.2  
}
```

### Aufgabe 2.2 Programm ergänzen (1 Punkt)

Mögliche Lösung – 1 Punkt, falls Mittelpunkt korrekt gesetzt ist:

```
float breite = 20;  
rect(mouseX - breite / 2, mouseY - breite / 2, breite, breite);
```