Prüfung Programming: Bildbearbeitung

1. Programm bearbeitet Bildstruktur [5 Punkte]

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

```
public Picture bearbeite(Picture originalBild) {
	int breite = originalBild.getWidth();
	int hoehe = originalBild.getHeight();

	Picture neuesBild = new Picture(breite * 2, hoehe);

	for (int y = 0; y < hoehe; y++) {
		for (int x = 0; x < breite; x++) {
			Pixel originalPixel = originalBild.getPixel(x, y);

			Pixel neuerPixel = neuesBild.getPixel(x * 2, y);
			Pixel neuerPixel2 = neuesBild.getPixel(x * 2 + 1, y);

			kopiereFarben(originalPixel, neuerPixel);
			kopiereFarben(originalPixel, neuerPixel2);
		}
	}

	return neuesBild;
}

void kopiereFarben(Pixel originalPixel, Pixel neuerPixel) {
	neuerPixel.setRed(originalPixel.getRed());
	neuerPixel.setGreen(originalPixel.getGreen());
	neuerPixel.setBlue(originalPixel.getBlue());
}
```

Um das Programm zu analyiseren, spielen Sie es von Hand auf Papier durch. Zeichnen Sie rechts das resultierende Bild "neuesBild" hin:

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

S	W	S
W	S	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:

Lösung [1 Punkt wenn alle Felder korrekt]

S	S	W	W	S	S
W	W	S	S	W	W
W	W	W	W	S	S

Beschreiben Sie in wenigen Worten präzise den Bildbearbeitungseffekt, den das Programm darstellt:

Mögliche Lösung [2 Punkte, 1 für horizontale Richtung, 1 für Verdoppelung Spalten]

Das Programm streckt das Bild in horizontaler Richtung um den Faktor zwei. Dazu werden alle Spalten des Bildes verdoppelt.

Das Programm verändert die Bildstruktur in einer Richtung (horizontal oder vertikal). **Geben Sie alle Programmzeilen an, die Sie ändern müssten (und nur diese, Sie brauchen nicht das ganze Programm abzuschreiben), damit es den gleichen Bildbearbeitungseffekt auf die andere Richtung (also vertikal oder horizontal) darstellt:**

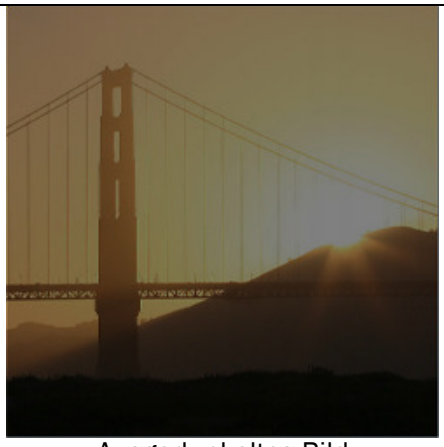
Lösung [2 Punkte, 1 Punkt für Grössenangaben, 1 Punkt für beide Koordinaten korrekt]

```
Picture neuesBild = new Picture(breite, hoehe * 2);

                    Pixel neuerPixel = neuesBild.getPixel(x, y* 2);
                    Pixel neuerPixel2 = neuesBild.getPixel(x, y * 2 + 1);
```

2. Programm verändert Bildfarben [5 Punkte]

Schreiben Sie ein Programm, das ein Bild ausdunkelt:

 Originalbild	 Ausgedunkeltes Bild

Das Bild soll ausgedunkelt werden, indem die Farbe der Pixel in ungeraden Spalte des Bildes (also Spalten mit Koordinaten x=1,3,5,...) auf schwarz gesetzt werden.

Ergänzen Sie im folgenden Programm die Methode bearbeiteFarben (nächste Seite):

- **Um zu prüfen, ob eine Zahl x ungerade ist, können Sie entweder if (x % 2 == 1) { ...} verwenden oder die Methode istUngerade(x) aufrufen.**
- **Falls Sie weitere Methoden brauchen, können Sie die Methoden unterhalb von bearbeiteFarben hinschreiben.**

```
public Picture bearbeite(Picture originalBild) {
        int breite = originalBild.getWidth();
        int hoehe = originalBild.getHeight();
        Picture neuesBild = new Picture(breite, hoehe);
        for (int y = 0; y < hoehe; y++) {
                for (int x = 0; x < breite; x++) {
                        Pixel originalPixel = originalBild.getPixel(x, y);
                        Pixel neuerPixel = neuesBild.getPixel(x, y);
                        bearbeiteFarben(x, originalPixel, neuerPixel);
                }
        }
        return neuesBild;
}

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

void kopiereFarben(Pixel originalPixel, Pixel neuerPixel) {
        neuerPixel.setRed(originalPixel.getRed());
        neuerPixel.setGreen(originalPixel.getGreen());
        neuerPixel.setBlue(originalPixel.getBlue());
}
```

```
void bearbeiteFarben(int x, Pixel originalPixel, Pixel neuerPixel) {




}
```

Mögliche Lösung [3 Punkte, 1 Punkt für korrekten Aufruf von istUngerade / if-Anweisung, 1 Punkt für Aufruf von kopiereFarben, 1 Punkt für schwarz färben]

```
void bearbeiteFarben(int x, Pixel originalPixel, Pixel neuerPixel) {
	if (istUngerade(x)) {
		kopiereFarben(originalPixel, neuerPixel);
	} else {
		macheSchwarz(neuerPixel);
	}
}

void macheSchwarz(Pixel neuerPixel) {
	neuerPixel.setRed(0);
	neuerPixel.setGreen(0);
	neuerPixel.setBlue(0);
}
```

Betrachten wir eine andere Art von Ausdunkeln. Unten sehen Sie das obige Programm leicht angepasst: Die Methode bearbeiteFarben erhält neu auch die Breite des Bildes mitgeteilt, und sie ist bereits ausprogrammiert:

```
// [...]
                        bearbeiteFarben(x, breite, originalPixel, neuerPixel);
// [...]

void bearbeiteFarben(int x, int breite, Pixel originalPixel, Pixel neuerPixel) {
        neuerPixel.setRed(originalPixel.getRed() * x / (breite – 1));
        neuerPixel.setGreen(originalPixel.getGreen() * x / (breite – 1));
        neuerPixel.setBlue(originalPixel.getBlue() * x / (breite – 1));
```

Um das Programm zu analyiseren, spielen Sie es von Hand auf Papier durch. Zeichnen Sie rechts das resultierende Bild "neuesBild" hin:

Der Einfachkeit halber verwendet das Bild nur Grauwerte (rot=grün=blau) als Farben; im Bild unten ist daher pro Pixel nur ein Farbwert angegeben. Das Eingabebild für die Analyse sei 3x3 Pixel gross:

10	4	6
17	28	24
11	10	18

Zeichnen Sie unten das resultierende Bild hin, in gleicher Art wie das Originalbild links: Jedes Kästchen ist ein Pixel. Sie können ebenfalls pro Farbe nur den Grauwert hinschreiben:

Lösung [1 Punkt wenn in mindestens einer Zeile alle Werte richtig]

0	2	6
0	14	24
0	5	18

Beschreiben Sie in wenigen Worten präzise den Bildbearbeitungseffekt, den das Programm darstellt:

Mögliche Lösung [1 Punkt, falls von links nach rechts zunehmende Farbstärke genannt wird]

Das Programm setzt die Farbwerte von links nach rechts stärker: Links 0% des Originalwertes, rechts 100% des Originalwertes. Die einzelnen Farbanteile rot, grün, blau werden dabei unabhängig voneinander betrachtet.