

Aufgabe 2

Leider ist das Klassendiagramm der folgenden Klassen verloren gegangen. Führen Sie ein Reverse Engineering durch und erstellen Sie aus dem Quellcode ein vollständiges UML-Klassendiagramm inklusive aller Klassen, Schnittstellen, Attribute, Methoden, Konstruktoren, Sichtbarkeiten, Assoziationen, Rollenamen, Multiplizitäten, Navigationspfeilen und evtl. Stereotypen. Der Quellcode innerhalb von Methoden und Konstruktoren soll nicht übertragen werden, wohl aber die Methodensignaturen. Assoziationsnamen und deren Leserichtung lassen sich aus dem Quellcode nur schwer erraten und sollen deshalb ebenfalls weggelassen werden.

```
3 public abstract class Display implements PixelPainter {
4     protected HardwareMatrix hardwareMatrix;
5     protected int lastPaintedX;
6     protected int lastPaintedY;
7
8     public Display(HardwareMatrix hardwareMatrix) {
9         this.hardwareMatrix = hardwareMatrix;
10    }
11
12    public int getWidth() {
13        return hardwareMatrix.getWidth() / getWidthFactor();
14    }
15
16    public int getHeight() {
17        return hardwareMatrix.getHeight() / getHeightFactor();
18    }
19
20    public void clear() {
21        // some longer code
22    }
23
24    protected abstract int getWidthFactor();
25
26    protected abstract int getHeightFactor();
27
28 }
```

[github: raw](#)

```
3 import java.awt.Color;
4
5 public interface PixelPainter {
6     void set(int x, int y, Color color);
7
8     int getHeight();
9
10    int getWidth();
11 }
```

[github: raw](#)

```
3 public interface HardwareMatrix {
4     void set(int x, int y, int v);
5
6     int getWidth();
7
8     int getHeight();
9 }
```

github: raw

```
3 import java.awt.Color;
4
5 @SuppressWarnings({ "unused" })
6 public class DisplayUnion extends RGBDisplay {
7     public static final int MAX_DISPLAY_COUNT = 50;
8
9     private int currentDisplayCount;
10
11     private Display[] displays;
12
13     public DisplayUnion(Display[] displays) {
14         super(null);
15     }
16
17     public int getDisplayCount() {
18         return 0;
19     }
20
21     protected int getWidthFactor() {
22         return 1;
23     }
24
25     protected int getHeightFactor() {
26         return 1;
27     }
28
29     public void set(int x, int y, Color color) {
30     }
31 }
```

github: raw

```
3 import java.awt.Color;
4
5 public class RGBDisplay extends Display {
6     public RGBDisplay(HardwareMatrix matrix) {
7         super(matrix);
8     }
9
10    public void set(int x, int y, Color color) {
11    }
12
13    protected int getWidthFactor() {
14        return 3;
15    }
16
17    protected int getHeightFactor() {
18        return 1;
19    }
20 }
```

github: raw

