

Diskussion am: 15.01.2025

Aufgabe 1: Wiederholung

- a) Welche SOLID Kriterien gibt es?
- b) Wie unterscheiden sich DIP und OCP?

Aufgabe 2: SOLID-Kriterien

Die neue Bildbearbeitungssoftware IntelliPhoto ist ein interaktives Tool zum Anzeigen und Bearbeiten von Bildern. Jedes Bild wird durch ein zweidimensionales Array von Bytes repräsentiert, wobei jeder Byte-Wert für einen Farbwert des Bildpunktes steht. Der Benutzer soll in der Lage sein, die Bilddimensionen abzufragen. Es sollen zwei verschiedene Arten von Bildern repräsentiert werden können: RasterImage und ShapedImage, wobei letzteres eine Spezialform vom RasterImage ist. Ein ShapedImage besitzt eine nicht-rechteckige Form (Polygon), wobei die Bytes im Array angeben, ob die jeweiligen Punkte transparent oder opak dargestellt werden sollen. Darüber hinaus soll die Software einfache Manipulationen von Bildern erlauben. So soll das Drehen, als auch das Vergrößern und Verkleinern von Bildern, das Setzen neuer Farbwerte im Bild und das Zusammenfügen zweier Bilder zu einem neuen Bild innerhalb von 0,2 Sekunden möglich sein.

Ihre Auftraggebenden haben das Tool leider etwas unfertig auf den Markt gebracht, und möchten nun nachträglich Funktionen ausbessern und neue anfügen. Für unterschiedliche Ansätze wurden bereits Prototypen geschaffen, die Ihnen vorgelegt werden sollen. Sie werden gebeten, die Prototypen zu bewerten und einzuordnen, ob und welche Änderungen vonnöten seien.

Aufgabe 2.1

Bevor neue Features implementiert werden, teilen Ihre Auftraggebenden die aktuelle Implementierung des Öffnens der Raster und ShapedImages mit Ihnen. Dieses erfolgt direkt mit dem Starten des Editors. Sie wollen sichergehen, dass die Basis des Projektes vernünftig ist, bevor weitere Anpassungen gemacht werden.

```
1  public class ImageEditor {
2      private EditorDisplay editorDisplay;
3
4      public ImageEditor(EditorDisplay editorDisplay, Image image) {
5          this.editorDisplay = editorDisplay;
6
7          image.display(editorDisplay);
8      }
9  }
10
11 public abstract class Image {
12     Information getRenderInfo(); // Information is a placeholder for
    the actual return type
13     void display(EditorDisplay editorDisplay);
14 }
15
16 public class RasterImage extends Image {
17     private int height;
18     private int width;
19
20     public Information getRenderInfo() {
21         return (<image information>);
22     }
23
24     public void display(EditorDisplay editorDisplay) {
25         editorDisplay.renderImage(<image information>);
26     }
27 }
28
29 public class ShapedImage extends Image {
30     private int height;
31     private int width;
32
33     public Information getRenderInfo() {
34         return (<image information>);
35     }
36
37     public void display(EditorDisplay editorDisplay) {
38         editorDisplay.renderImage(<image information>);
39     }
40 }
```

- a) Gegen welches SOLID-Prinzip verstößt der gegebene Prototyp? Begründen Sie Ihre Antwort.
- b) Passen Sie den Prototypen dem SOLID-Prinzip entsprechend an. Hierfür können Sie entweder (a) Quelltext oder (b) eine genaue Beschreibung der Änderungen, die am Prototypen vorgenommen werden müssen, angeben.

Aufgabe 2.2

Ihre Auftraggebenden haben die Implementierung der Öffnung nun angepasst und möchten erneut Ihre Einschätzung. Es wird betont, dass zu späteren Zeitpunkten weitere Bildtypen angefügt werden sollen.

```
1 public class ImageEditor {
2     private EditorDisplay editorDisplay;
3
4     public ImageEditor(EditorDisplay editorDisplay, RasterImage
5 rasterImage, ShapedImage shapedImage) {
6         this.editorDisplay = editorDisplay;
7
8         if (rasterImage != null) {
9             editorDisplay.renderImage(rasterImage.getRenderInfo());
10        }
11        else if (shapedImage != null) {
12            editorDisplay.renderImage(shapedImage.getRenderInfo());
13        }
14    }
15
16    public class RasterImage {
17        // Information is a placeholder for the actual return type
18        public Information getRenderInfo() {
19            return (<image information>);
20        }
21    }
22
23    public class ShapedImage {
24        public Information getRenderInfo() {
25            return (<image information>);
26        }
27    }
```

- a) Gegen welches SOLID-Prinzip verstößt der gegebene Prototyp? Begründen Sie Ihre Antwort.
- b) Passen Sie den Prototypen dem SOLID-Prinzip entsprechend an. Hierfür können Sie entweder (a) Quelltext oder (b) eine genaue Beschreibung der Änderungen, die am Prototypen vorgenommen werden müssen, angeben.

Aufgabe 2.3

Die Auftraggebenden haben sich nach Ihren Anmerkungen für die Implementierung eines Interfaces entschieden, um beide Image Arten zu implementieren und das Anfügen weiterer Typen leichter zu gestalten.

```
1  public interface Image {
2      // Information is a placeholder for the actual return type
3      Information getRenderInfo();
4      Polygon getPolygonShape();
5  }
6
7  public class RasterImage implements Image {
8      @Override
9      public Information getRenderInfo() {...}
10
11     @Override
12     public Polygon getPolygonShape() {
13         throw new UnsupportedOperationException("RasterImage does not
14         support polygon shapes");
15     }
16
17     public class ShapedImage implements Image {
18         private Polygon polygonShape;
19
20         @Override
21         public Information getRenderInfo() {...}
22
23         @Override
24         public Polygon getPolygonShape() {
25             return this.polygonShape; // Example placeholder
26         }
27     }
```

- a) Gegen welches SOLID-Prinzip verstößt der gegebene Prototyp? Begründen Sie Ihre Antwort.
- b) Passen Sie den Prototypen dem SOLID-Prinzip entsprechend an. Hierfür können Sie entweder (a) Quelltext oder (b) eine genaue Beschreibung der Änderungen, die am Prototypen vorgenommen werden müssen, angeben.

Aufgabe 2.4

Die Auftraggebenden glauben, alle Kritiken verinnerlicht zu haben und wagen sich an eine komplexere Funktionalität: Die Anwendung von Filtern. Hierfür werden erstmal beispielhaft zwei Filter (Greyscale und Sepia) implementiert, die aber später erweitert werden sollen.

```
1  public enum FilterType {
2      GRAYSCALE,
3      SEPIA
4  }
5
6  public class ImageEditor {
7      public void applyFilter(Image image, FilterType filterType) {
8          switch (filterType) {
9              case GRAYSCALE:
10                 System.out.println("Applying grayscale filter...");
11                 image.applyGrayscaleFilter();
12                 break;
13              case SEPIA:
14                 System.out.println("Applying sepia filter...");
15                 image.applySepiaFilter();
16                 break;
17              default:
18                 System.out.println("Unsupported filter type!");
19                 break;
20          }
21      }
22  }
23
24  public abstract class Image {
25      public abstract void applyGrayscaleFilter();
26      public abstract void applySepiaFilter();
27  }
```

- a) Gegen welches SOLID-Prinzip verstößt der gegebene Prototyp? Begründen Sie Ihre Antwort.
- b) Passen Sie den Prototypen dem SOLID-Prinzip entsprechend an. Hierfür können Sie entweder (a) Quelltext oder (b) eine genaue Beschreibung der Änderungen, die am Prototypen vorgenommen werden müssen, angeben.

Aufgabe 2.5

Dieses Mal sind die Auftraggeber*innen überzeugt davon, dass Sie sich an alle Ihre Anweisungen gehalten haben. Sie haben ihren Prototypen nun um Bilder der Art *CurvedImage* erweitert. Was halten Sie vom neuesten Prototypen?

```
1  public class ImageEditor {
2      public ImageEditor(EditorDisplay editorDisplay, Image image) {
3          image.display(editorDisplay);
4      }
5  }
6
7  public abstract class Image {
8      // Information is a placeholder for the actual return type
9      public abstract Information getRenderInfo();
10 }
11
12 public class RasterImage extends Image {
13     @Override
14     public Information getRenderInfo() {
15         return (<image information>);
16     }
17 }
18
19 public class ShapedImage extends RasterImage {
20     private Polygon polygonShape;
21
22     @Override
23     public Information getRenderInfo() {
24         return this.polygonShape.getRenderInfo();
25     }
26 }
27
28 public class CurvedImage extends Image {
29     @Override
30     public Information getRenderInfo() {
31         throw new UnsupportedOperationException("Render info for
CurvedImage is not yet implemented");
32     }
33 }
```

- a) Gegen welches SOLID-Prinzip verstößt der gegebene Prototyp? Begründen Sie Ihre Antwort.
- b) Passen Sie den Prototypen dem SOLID-Prinzip entsprechend an. Hierfür können Sie entweder (a) Quelltext oder (b) eine genaue Beschreibung der Änderungen, die am Prototypen vorgenommen werden müssen, angeben.