

## Arbeitsblatt: Immutability

Im Unterricht haben Sie die diversen Vorteile von unveränderbaren Objekten (Immutables) gesehen. In den allgemein verbreiteten OOP-Sprachen wie Java und C# werden Immutables nicht durch Sprach-Konstrukte oder das Typsystem unterstützt, sondern müssen selbst als Klassen gestaltet werden, die speziellen Regeln folgen. Ziel dieses Arbeitsblattes ist es, solche Regeln herauszufinden.

### 1. Immutable Line (unter Verwendung von `java.awt.Point`)

Gegeben ist die Klasse `MutableLine` (sie finden diese Klasse im Paket `patterns.clone.immutable` im Wochenprojekt auf dem AD).

```
public class MutableLine implements Cloneable {
    public Point start, end;

    public MutableLine(Point start, Point end) {
        this.start = start;
        this.end = end;
    }

    public Point getStartPoint() { return start; }
    public Point getEndPoint() { return end; }

    public void setStartPoint(Point start) { this.start = start; }
    public void setEndPoint(Point end) { this.end = end; }

    @Override
    public MutableLine clone() {
        try {
            MutableLine p = (MutableLine) super.clone();
            p.start = (Point)start.clone();
            p.end = (Point)end.clone();
            return p;
        } catch (CloneNotSupportedException e) {
            throw new InternalError();
        }
    }

    @Override
    public String toString() {
        return String.format("Line[start=%s, end=%s]", start, end);
    }
}
```

Entwerfen und programmieren Sie auf Basis von `MutableLine` eine neue Klasse `ImmutableLine`, deren Instanzen unveränderbar sind.

Da Immutables nicht modifiziert werden können, geben „modifizierende Methoden“ neue Instanzen zurück. Tipp: Betrachten Sie dazu die Signatur der Methoden `concat` und `substring` der unveränderbaren Klasse `String`.

Formulieren Sie die Vorkehrungen die Sie treffen, damit `ImmutableLine`-Instanzen immutable sind, als Regeln.

### 2. Immutable Line (unter Verwendung einer Immutable-Point Klasse)

Wie würde sich die Klasse `ImmutableLine` vereinfachen, wenn für Start- und Endpunkt anstelle einer veränderbaren Klasse eine unveränderbare Klasse `Point` verwendet wird?