

Erste Idee: Digitale Darstellung als String-Array (schlecht)

Zunächst dachte ich daran, die Ziffern als Arrays von Strings zu repräsentieren, wie sie auf der Konsole dargestellt werden sollen.

Zum Beispiel würde die Ziffer 1 so aussehen: [" ", " ", " |", " "]

Dann wollte ich die Ziffern von 1 bis 5 entlang der x- und y-Achse spiegeln und einen Algorithmus finden, um die gespiegelten Ziffern zu berechnen. Dies erwies sich jedoch als sehr kompliziert und schwer umzusetzen, insbesondere wegen der Konsolendarstellung, die nur mit "_", "|" oder " " arbeiten sollte.

Zweite Idee: Ähnlichkeit der Spiegelbilder (schlecht und ineffizient)

Meine nächste Überlegung war, dass manche Ziffern die gleichen oder ähnliche Spiegelbilder haben. Zum Beispiel bleibt die Ziffer 1 an der x-Achse 1 und an der y-Achse 1, 2 ist 5 und 5 ist 2 etc pp. Daher dachte ich, dass es eine bestimmte Menge von Spiegelbildern und ihren Urbildern gibt. Diese Menge könnte ich speichern und die Berechnung somit umgehen.

Dritte Idee: 7-Segment-Darstellung (gut und effizient)

Am Ende war ich jedoch zu faul, um die zweite Idee umzusetzen. Ich erinnerte mich daran, dass ich die Darstellung von Ziffern in digitalen Uhren schon in der Schule gelernt hatte. Jede Ziffer kann als Kombination von 7 Segmenten dargestellt werden, was die Spiegelung sehr einfach und deutlich macht.

Ich habe alles auf Papier getestet, Pseudocode geschrieben und während der Implementierung natürlich die Clean Code und SOLID-Prinzipien verfolgt. Jede Klasse erfüllt genau eine Aufgabe. Zum Beispiel kümmert sich die Klasse MirrorPrinter nur um die Ausgabe des Ergebnisses, während andere Klassen keine Druckaufgaben übernehmen.

Fazit

Durch die Umsetzung dieser Ideen habe ich schließlich eine effiziente und nachvollziehbare Lösung gefunden, um die Spiegelung von Ziffern darzustellen.