

# Planung

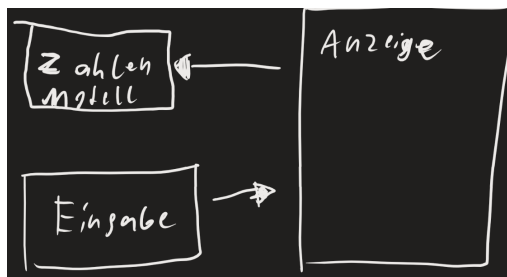
Erstmal habe ich die grundlegenden Sachen aus der Aufgabe genommen, die das Programm braucht und mir Kompetente überlegt, die man dazu brauchen könnte und was die Funktionen sind.

Aufgabe:

Es soll möglich sein eine Zahl an der X und Y-Achse zu spiegeln in einer Digitale Darstellung. Dabei soll es dann so aussehen wie auf der Zeichnung laut der Aufgabe:

	Y-ACHSE	
X-ACHSE	0234	PE50
	0534	hE20

**Komponenten, die ich benötige:**



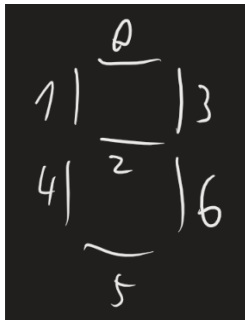
1. Anzeige-Klasse:
  - Speichert und verwaltet die Zahlen, enthält Methoden zum Hinzufügen und Spiegeln von Zahlen.
  - Implementiert die Darstellung der Zahlen
2. Eingabe-Komponente:
  - Kümmt sich um die Tastatureingaben und die Zahleneingabe.
  - Verarbeitet die Eingaben der Tasten 'x' und 'y'.
  - Leitet die Eingaben an die Anzeige-Komponente weiter.
3. Zahlen-Klasse:
  - Speichert die Daten für die entsprechende Darstellung der Zahlen.
  - Verwaltet die notwendigen Informationen über die Zahlen für die Anzeige.

**Entscheidung für die Programmiersprache:**

Ursprünglich hatte ich geplant, die Aufgabe in Java zu lösen. Allerdings stellte sich heraus, dass Java in einer Konsolen-Umgebung keine Tastatureingaben direkt unterstützt, außer man verwendet Enter. Da jedoch die Anforderung besteht, dass die Anzeige gespiegelt werden soll, je nachdem, ob die Tasten 'x' oder 'y' gedrückt werden, musste ich nach einer anderen Lösung suchen. Ich hätte Java mit einer Erweiterung, die das verwenden können,

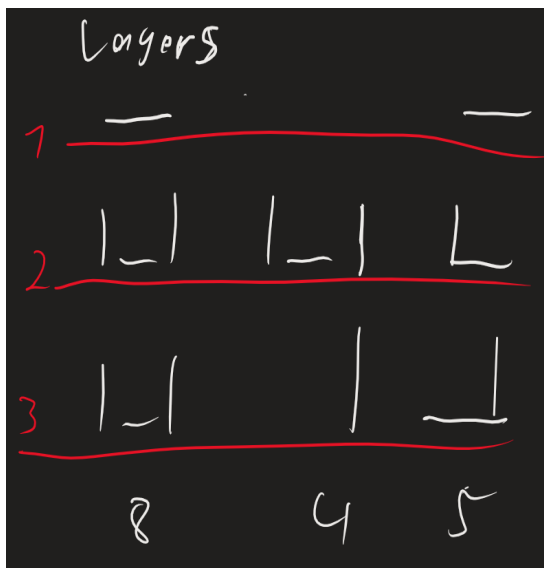
aber ich habe das schon mal mit Python gemacht mit einem Packet und aus diesem Grund habe ich mich für Python entschieden.

## Ideensammlung zur Darstellung der Zahlen:



Ich habe darüber nachgedacht, wie ich die Zahlen am besten darstellen kann. Dazu habe ich ein Modell entwickelt, bei dem die Zahlen in sieben verschiedene Segmente aufgeteilt werden. Die Darstellung wird aus drei Zeilen bestehen, wobei die Segmente in den Zeilen angeordnet werden, um die entsprechenden Zahlen zu repräsentieren.

## Struktur der Zahlen



Nachdem ich die Elemente der Zahl hatte, habe ich mir Gedanken darüber gemacht, wie ich sie in der Konsole anzeigen kann. Jede Ziffer wird als Liste von 7 Werten dargestellt, wobei waagerechte Striche mit "\_" dargestellt werden, senkrechte Striche mit "|", und leere Stellen mit Leerzeichen " ".

In der ersten Zeile muss immer entweder ein "\_" sein oder nur Leerzeichen. Die Leerzeichen am Anfang und Ende der ersten Linie müssen also immer manuell hinzugefügt werden. Ab Zeile 2-3 bestehen die Zahlen einfach aus 123 und 456 der Zahl.

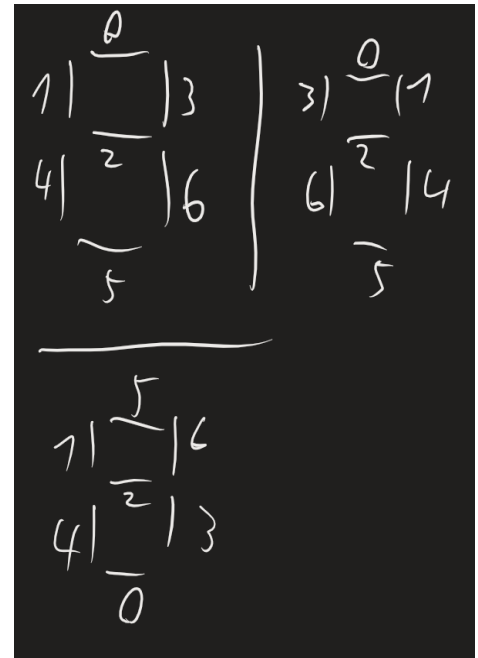
## Spiegeln der Zahlen

Anschließend habe ich darüber nachgedacht, wie ich die Zahlen horizontal und vertikal spiegeln kann. Ich habe einfach die Positionen in der Liste entsprechend angepasst. Zudem habe ich erkannt, dass bei mehrstelligen Zahlen auch diese gespiegelt werden müssen.

## Erstellen der DigitNumber Klasse und BigNumber Klasse:

Daraufhin habe ich eine DigitNumber-Klasse erstellt, um die Ziffern abzuspeichern. Die Klasse hat eine String-Liste und zwei Funktionen, die sich darum kümmern, dass die Zahl gespiegelt wird. Dabei werden die Positionen in der Liste getauscht, um die Spiegelung zu erreichen und ein neues Nummern Objekt erstellt welches dann diese Liste hat.

Anschließend habe ich eine weitere Klasse erstellt, die 4 Digits verwaltet und Big Number genannt wird. Sie enthält eine Liste von Zahlen und verwaltet die Zahlen, wenn die Methoden mirror\_x oder mirror\_y aufgerufen werden. Diese Methoden rufen die entsprechenden Spiegelungsmethoden der DigitNumber-Klasse auf, und bei einer Y-Spiegelung wird auch die Reihenfolge in der Liste entsprechend angepasst und es wird dann eine neue BigNumber Objekte erstellt welches dann die gespiegelten Zahlen enthält.



## Eingabe der Zahl:

Danach habe ich eine Methode geschrieben, die die Eingabe von Zahlen in der Konsole verarbeitet. Dabei wird die Eingabe so lange eingelesen, bis der Benutzer eine Zahl mit 1-4 Stellen angibt. Ich verwende Regex, um zu überprüfen, ob eine valide Zahl eingegeben wurde.

## Erstellung der Anzeige-Klasse:

Nachdem ich die Eingabe der Zahl abgeschlossen hatte, entwickelte ich eine Klasse, um die Anzeige zu verwalten. Diese Klasse enthält eine BigNumber, die die vom Benutzer eingegebene Zahl repräsentiert, sowie zwei Werte, um festzuhalten, ob die Taste 'x' oder 'y' gedrückt wurde. Darüber hinaus enthält die Klasse eine Liste von Digit Numbers für alle Zahlen von 0 bis 9.

Die Klasse verfügt auch über Methoden, die aufgerufen werden können, wenn die Tasten 'x' oder 'y' gedrückt werden. Diese Methoden rufen dann die entsprechende Funktion in der BigNumber-Klasse auf, um die Nummer zu spiegeln.

## Erstellung einer Funktion zur Konvertierung einer Zahl in eine BigInteger:

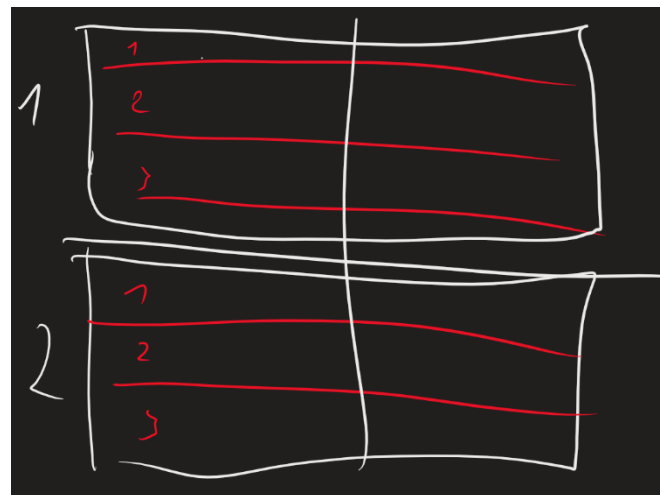
Im nächsten Schritt habe ich eine Funktion in der Anzeige-Klasse erstellt, die einen gültigen String erhält und daraus eine BigInteger erstellt. Wenn ich das Klassenobjekt später verwende, kann diese Methode verwendet werden. Zuerst wird ein Offset eingestellt, um anzuzeigen, ab wann keine 0 mehr zur BigInteger hinzugefügt werden sollen. Wenn kein führendes Nullen vorhanden sind oder waren, werden die Zahlen aus der Nummernliste der Anzeigen-Klasse genommen.

## Zeichnen der Zahlen:

Zur Darstellung musste ich es mir überlegen, wie ich es mit den Achsen mache. Und habe mich entschieden das Beste ist es einfach 2 Listen zu erstellen die die Zahlen haben. Einmal die oberhalb der X-Achse und die unterhalb dieser. Die Gespiegelten Varianten der Zahl werden einfach mit den Mirror Funktionen der Zahl geholt.

Diese beiden Listen werden an die Methode `draw_numbers` Zahlen übergeben und macht es dann, wie ich es mir am Anfang überlegt habe einen durchlaufe der drei Layer und füge für jede Zahl den entsprechenden String für den Layer hinzu.

Im ersten Layer füge ich einfach das erste Element aus dem Array ein, gefolgt von 2 Leerzeichen darum. Für die anderen beiden Layer verbinde ich die nächsten 3 Strings, um die Zahl zu zeichnen.



## Testen der Funktionen und grundlegender Programmablauf:

Anschließend habe ich die Anzeige-Klasse mit der Eingabe-Funktion in der Main-Methode verbunden, die Klasse getestet und dabei eventuelle Fehler behoben. Zuerst habe ich die Spiegel-Methoden nur durch einen Aufruf in der Klasse getestet, bevor ich den Listener im nächsten Schritt geschrieben habe.

## Hinzufügen von Tastatur-Listenern für 'x' und 'y':

Nachdem ich die Anzeige funktionsfähig hatte und die meisten Fehler behoben waren, habe ich Tastatur-Listener für die Tasten 'x' und 'y' hinzugefügt, um die Anzeige horizontal oder vertikal zu spiegeln. Hierfür habe ich die Pynput-Bibliothek verwendet, um die Keylistener zu implementieren. Die Keylistener rufen dann die entsprechende Methode in der Anzeige-Klasse auf.

## Erstellen der Anwendung:

Schließlich habe ich alles zum Laufen gebracht und am Ende eine Python-Exe erstellt, die ausgeführt werden kann.

Durch diese Schritte konnte ich eine Anzeige-Klasse entwickeln, die es dem Benutzer ermöglicht, eine Zahl einzugeben und diese an der X und Y-Achse wie auf der Abbildung zu spiegeln, je nachdem, ob die Tasten 'x' oder 'y' gedrückt werden.