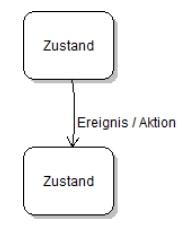
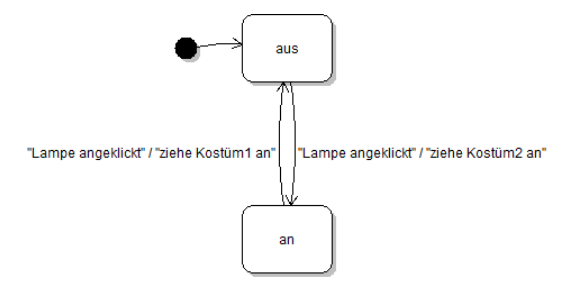
# Zustandsautomaten

Wenn der Automat im aktuellen Zustand eine bestimmte Eingabe erhält, dann erzeugt er eine Ausgabe und wechselt in einen Folgezustand (wobei dies auch der aktuelle Zustand sein kann).

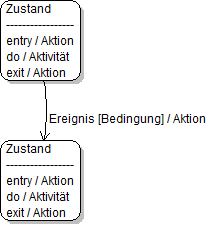


Einfaches BSP zu Ereignisgesteuerte Zustandsübergänge:

Ein/Ausschalten einer Lampe. Zwei Zustände (On/Off) und wird mittels Schalter geändert.



Eine weitere Verallgemeinerung des Automatenkonzepts erhält man, wenn Automaten innerhalb der Zustände Aktionen bzw. Aktivitäten ausführen können.



Eine Entry-Aktion wird beim Eintritt in einen Zustand ausgeführt, eine Exit-Aktion beim Verlassen eines Zustands.

Eine Do-Aktivität wird während der gesamten Zeit durchgeführt, in der sich das System im betreffenden Zustand befindet.

Beachte, dass hier zwischen Aktionen und Aktivitäten unterschieden wird.

Eine Aktion ist ein Vorgang im System, der nur sehr kurz dauert und für die Modellierung des Verhaltens unbedeutend ist. Aktionen können als (nahezu) zeitlose Vorgänge angesehen werden. Eine Aktivität ist dagegen ein zeitbehafteter Vorgang im System.

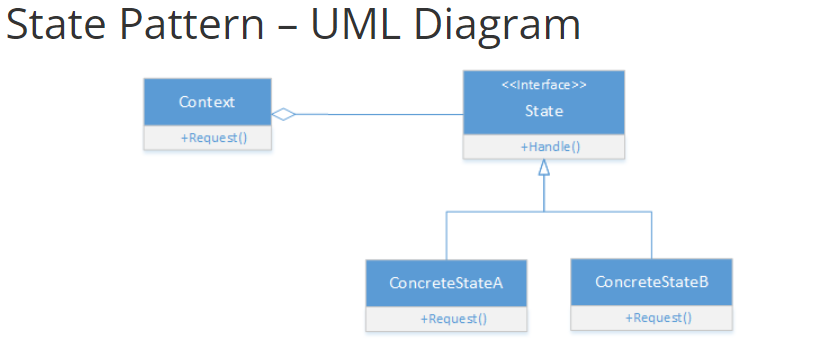
Einfache Darstellung in Scratch -> Siehe File: C:\Users\vkell\Desktop\AUD\_Zustandsautomaten

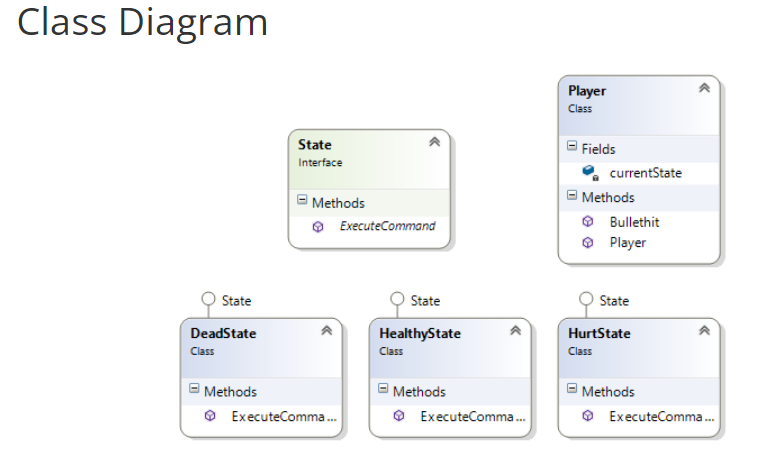
# Moore- und Mealy-Automat: Unterschied

Hast du schon mal etwas vom [Mealy-Automat](https://studyflix.de/informatik/mealy-automat-1217)gehört? Dieser ist dem Moore-Automaten sehr ähnlich. Der eigentliche Hauptunterschied zwischen Moore- und Mealy-Automat ist, dass die Ausgaben des Moore-Automaten nur davon abhängen, in welchem Zustand er sich befindet.

Im Mealy-Automaten ist die Ausgabe sowohl mit dem Zustand als auch mit der aktuellen Eingabe verbunden.

# State Pattern C#





C# Program-Path: C:\Users\vkell\Desktop\AUD\AUD\_Zustandsautomaten\AUD\_StatePattern