

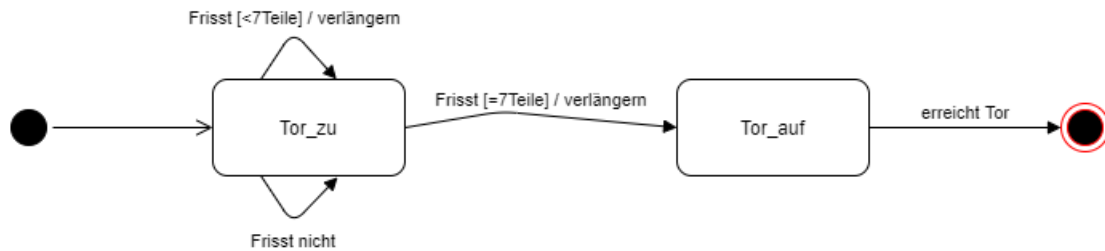
A1: Grundwissen Zustandsdiagramm:

Arbeiten Sie -wenn nötig- auf der Webseite

<https://www.inf-schule.de/modellierung/zustandsmodellierung> die Abschnitte 1.3 und 3.1 durch.

Zeichnen Sie dann ein Zustandsdiagramm zu folgender Beschreibung:

Das Spiel „Snake“ beginnt in einem bestimmten Level mit einer Schlange der Länge 5. Genau dann, wenn die Schlange einen „Futter“-Block erreicht und frisst, wird sie um ein Rumpfteil länger, wenn sie nicht frisst bleibt die Länge gleich. Erreicht die Schlange die Länge 8, öffnet sich ein „Ausgangstor“. Erreicht die Schlange das Ausgangstor, ist dieses Level des Spiels beendet

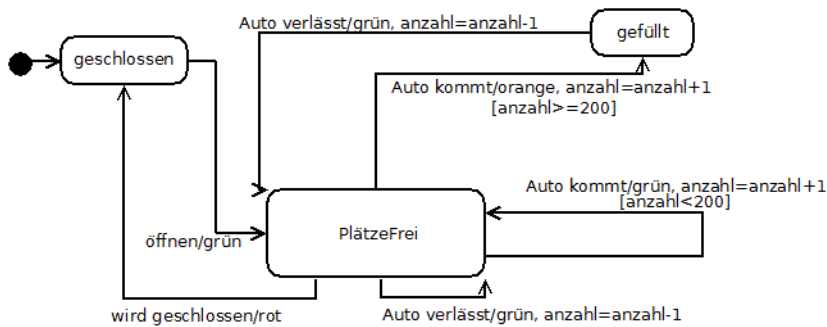


Natürlich kann man als Zustände auch die Snake-Länge wählen, aber so ist's halt kürzer.

Wichtig: Formalia einhalten! Abgerundete Rechtecke für Status; was passiert, wenn die Schlange NICHT frisst; es gibt „Start“- und „Endzustand“

Hier am interessantesten: z.b. Frisst [< 7 Teile] / verlängern ← **Ereignis [Bedingung] / Aktion**

A2: Grundwissen Zustandsdiagramm:



Betrachten Sie dieses Zustandsdiagramm. Beschreiben Sie die modellierte Situation.

Es handelt sich um ein Parkhaus. Am Anfang ist es geschlossen und ein Licht leuchtet rot. Sobald es öffnet wird das Licht grün (= es sind darin Plätze frei). Kommt ein Auto an, wird die Anzahl der belegten Plätze um 1 erhöht. Solange weniger als 200 Plätze belegt sind, bleibt die Signalanlage auf „grün“, sind mindestens 200 Plätze belegt zeigt die Signalanlage „orange“ an und das Parkhaus ist „gefüllt“. Jedes Auto, das das Parkhaus verlässt, verringert die Anzahl der belegten Plätze um 1, die Signalanlage wechselt dann stets auf „grün“. Beim Schließen des Parkhauses wird das Licht der Signalanlage auf „rot“ gestellt. Das Haus ist geschlossen.

A3: Grundwissen Programmierung:

Markiere im folgendem Greenfoot(Java) -Codeschnippel jeweils mindestens (a) zwei Deklarationen, (b) einen Konstruktor, (c) zwei Attribute, (d) eine Variable, die nur während der Ausführung einer Methode existiert, (e) eine sondierende Methode (GetterMethode) und (f) eine Methode, die mindestens ein Attribut ändert.

Häufig nicht in Ordnung:

Parameter sind im eigentlichen Sinn KEINE Variablen

Ein Konstruktor heißt genau so wie die Klasse

```
class Eintrittskarte extends Actor ▶  
  
private static final double MWST ← 19.0      a) c)  
private String zustand                       a) c)  
private int nummer                           a) c)  
private double preis                          a) c)  
  
.  
.  
public Eintrittskarte(String neueOper, double neuPreis)  b)  
    zustand ← "verfügbar"  
    nummer ← 143  
    preis ← neuPreis  
  
.  
.  
public double preisGeben ()                          e)  
    return preis  
  
-  
.  
public void preisAendern (double nPreis)              f)  
    preis ← nPreis  
  
.  
.  
public double gebenMWST ()                            d)  
    var double mwstWert  
    mwstWert ← MWST * preis  
    return mwstWert
```