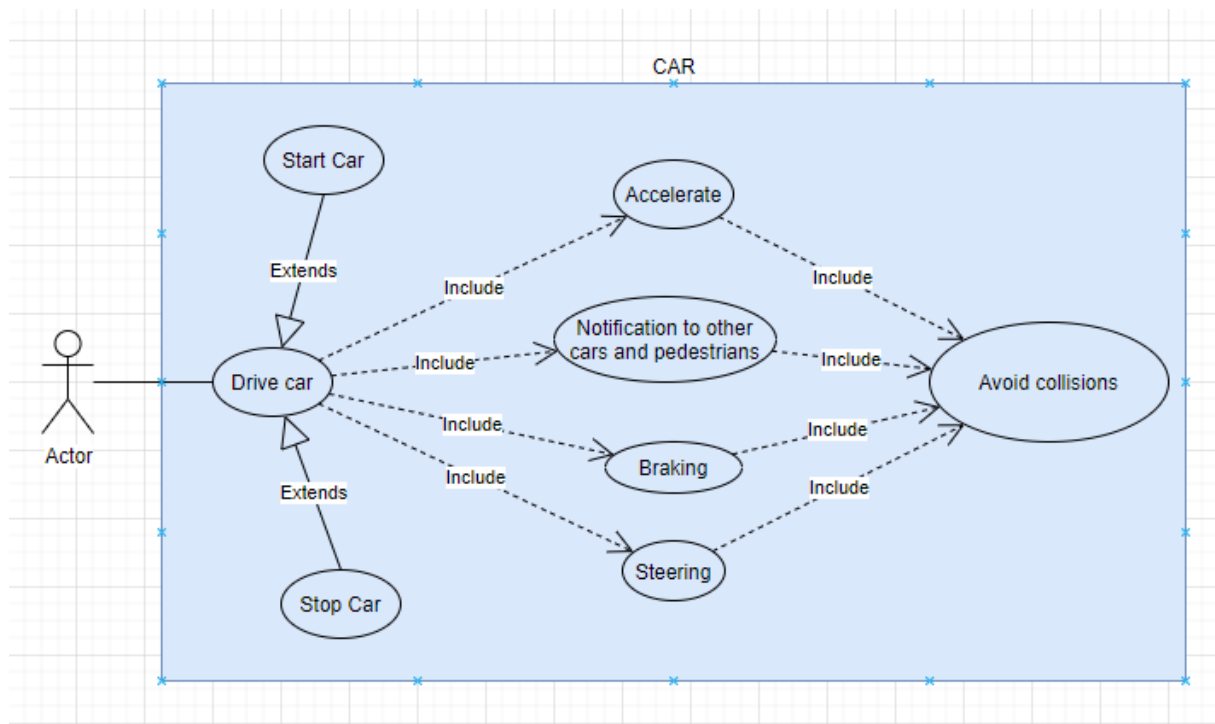


## Nachdenkzettel UML

1. Erstellen Sie ein Use Case Diagramm für ein Auto



2. Eine Bank hat viele Kunden. Malen Sie das Klassendiagramm für die Beziehungen. Wie kann man diese Beziehungen in der Implementation in Java darstellen?
3. Wie kommen Sie vom Use Case Diagramm zum Klassendiagramm?

Am sinnvollsten ist es, die Use Cases des Use Case Diagramms als Klassen im Klassendiagramm zu übertragen, und anschließend die Methoden und Variablen zu diskutieren.

4. Was ist der Unterschied zwischen Aggregation und Komposition? Was bedeutet der Unterschied in deiner Datenbank?

## Aggregation

Eine *Aggregation* ist eine Zusammensetzung von Objekten (Komponenten) zu einem zusammengesetzten Objekt. („Teil-Ganzes Beziehung“)

Objekte werden von dem besitzenden Objekt dominiert.

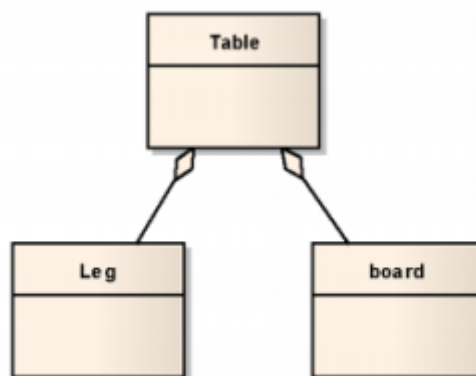
i.A. ist kein externer Zugriff auf aggregierte Objekte möglich.

Beispiel 1:



Ein Student nimmt an einer Vorlesung teil.

Beispiel 2:



Leg und Board sind Teile von Table, weil ja ein Tisch aus Beinen und einer Platte besteht.

Ein Pfeil von Table zu Leg macht keinen Sinn, weil ein Bein ja nicht aus einem Tisch besteht.

## Komposition

Eine *Komposition* ist eine spezielle Aggregation, bei der die Existenz der Teile (Komponenten) von der Existenz des Ganzen abhängt.

Objekte werden von dem besitzenden Objekt dominiert.

Die Lebensdauer (Instanziierung und Freigabe) werden vom dominierenden Objekt beherrscht.

Beispiel:



Wenn es kein Gebäude mehr gibt, existiert auch kein Raum mehr.



In einer Datenbank entspricht beispielsweise der Durchschnittswert eines Datensatzes einer *Aggregation*.

Eine Komposition ist eine Art Regel, die beispielsweise einen Eintrag in die DB nur dann zulässt, wenn zu einem Kunden auch eine Adresse existiert.

5. Definieren Sie eine Klasse Game mit ihren Attributen

```

public class Game {
    Player1 = player1;
    Player2 = player2;
    Menu = menu;
    Map = map;
    MiniMap = miniMap;
    GameManager = gameManager;
    RenderingEngine = renderingEngine;
    Sound = sound;
}
  
```

6. Was bedeutet es, wenn eine Klasse in UML ein „public“ Attribut hat?

Unbeschränkter Zugriff → alle anderen Klassen können auf die Klasse bzw. deren Variablen und Methoden zugreifen.