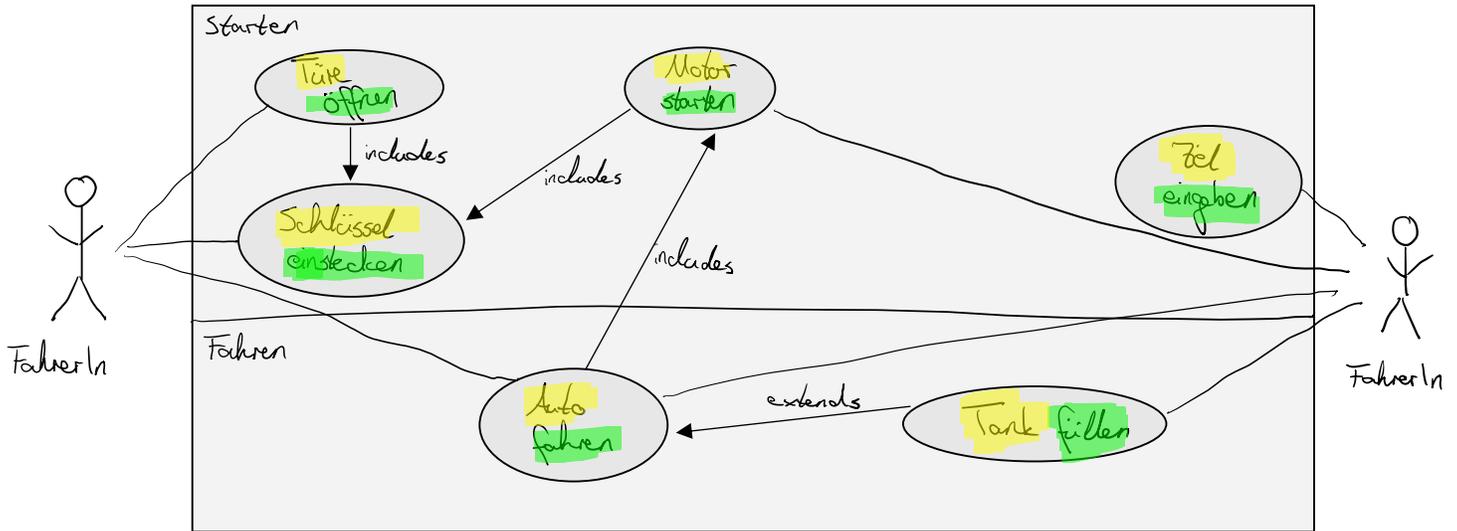
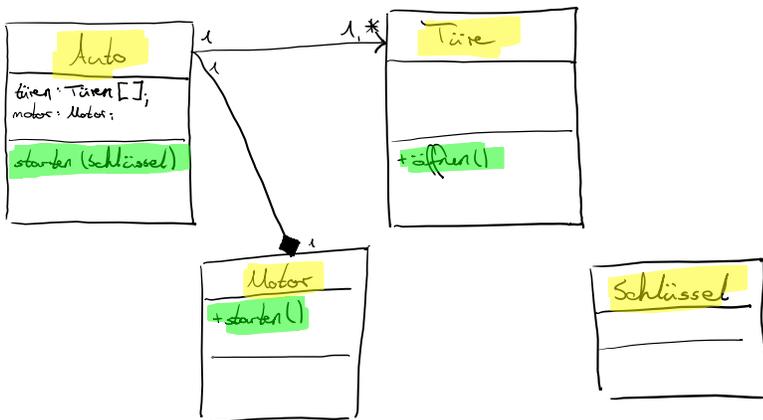


.....
 Vorname, Name, Matrikelnummer

1. Erstellen Sie ein Use Case Diagramm für ein Auto



Dazugehöriges Klassendiagramm:

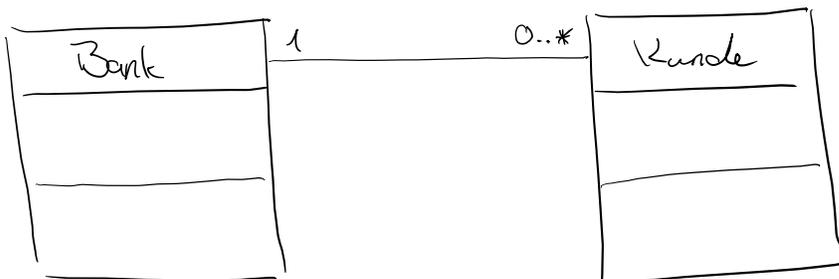


```

class Auto {
    public Auto() {
        türen [0] = new Tür();
        türen [1] = new Tür();
        ...
    }
    public Auto(türen[]) {
        ...
    }
}
    
```

2. Eine Bank hat viele Kunden. Malen Sie das Klassendiagramm für diese Beziehung.

Wie kann man diese Beziehung in der Implementation in Java darstellen?



3. Wie kommen Sie vom Use Case Diagramm zum Klassendiagramm?

-Nomen aus dem Use-Case-Diagramm sind mit hoher Wahrscheinlichkeit nützlich für die Applikation und können deshalb als erste Klassen übernommen werden

-anschließend überlegt man sich die Beziehungen zwischen den Klassen

-Verben aus dem Use-Case-Diagramm können als Startpunkt für Methoden genutzt werden (je nach Methode kann es jedoch sinnvoll sein, die Methoden in anderen Klassen anzusiedeln, z.B. einstecken nicht in der Klasse Schlüssel, sondern in der Klasse Auto implementieren)

-während Implementierung kommen weitere Klassen und Methoden dazu, die sich nicht direkt aus dem Use-Case-Diagramm ergeben, aber ein Anfang ist gemacht

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

-bei einer Komposition können die einzelnen Teile nicht ohne einander existieren (z.B. kann kein Auto ohne Motor und kein Motor ohne Auto existieren)

-bei einer Aggregation können einzelne Teile auch ohne andere Teile auskommen

5. Definieren Sie eine Klasse Game mit ihren Attributen



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

→ Da hat sich wohl jemand erwischen lassen ...

8. Hausaufgabe für jedes Team: in der Übung Use Case und Klassendiagramm für Ihr Projekt erstellen! Wir schauen uns die ersten Schritte morgen an im Review.