

**Hausarbeit Entwurf von Informationssystemen**  
**Universität Bremen, Informatik**  
**SoSe 2008**

**Kai Michael Poppe, 1695749, [kmpoppe@informatik.uni-bremen.de](mailto:kmpoppe@informatik.uni-bremen.de)**  
**Bastian Kammann, 1695733, [bkammann@informatik.uni-bremen.de](mailto:bkammann@informatik.uni-bremen.de)**

# Inhaltsverzeichnis

|                                   |    |
|-----------------------------------|----|
| Inhaltsverzeichnis.....           | 2  |
| Vorwort.....                      | 5  |
| Modellbeschreibung.....           | 5  |
| Klassenbeschreibungen .....       | 6  |
| CurrentDate.....                  | 6  |
| Attribute.....                    | 6  |
| Operationen .....                 | 6  |
| Invarianten .....                 | 6  |
| Definition im USE-Quelltext.....  | 6  |
| Brand.....                        | 7  |
| Attribute.....                    | 7  |
| Operationen .....                 | 7  |
| Invarianten .....                 | 7  |
| Definition im USE-Quelltext ..... | 7  |
| Station.....                      | 8  |
| Attribute.....                    | 8  |
| Operationen .....                 | 8  |
| Invarianten .....                 | 8  |
| Definition im USE-Quelltext ..... | 8  |
| Vehicle .....                     | 9  |
| Attribute.....                    | 9  |
| Operationen .....                 | 10 |
| Invarianten .....                 | 11 |
| Definition im USE-Quelltext ..... | 12 |
| Car.....                          | 14 |
| Attribute.....                    | 14 |
| Operationen .....                 | 14 |
| Invarianten .....                 | 14 |
| Definition im USE-Quelltext ..... | 14 |
| Truck.....                        | 14 |
| Attribute.....                    | 14 |
| Operationen .....                 | 14 |
| Invarianten .....                 | 14 |
| Definition im USE-Quelltext ..... | 14 |
| Driver.....                       | 15 |
| Attribute.....                    | 15 |

|                                                                |    |
|----------------------------------------------------------------|----|
| Operationen .....                                              | 15 |
| Invarianten .....                                              | 16 |
| Definition im USE-Quelltext .....                              | 16 |
| IdentityCard .....                                             | 17 |
| Attribute .....                                                | 17 |
| Operationen .....                                              | 17 |
| Invarianten .....                                              | 17 |
| Definition im USE-Quelltext .....                              | 17 |
| DrivingLicense .....                                           | 18 |
| Attribute .....                                                | 18 |
| Operationen .....                                              | 18 |
| Invarianten .....                                              | 18 |
| Definition im USE-Quelltext .....                              | 18 |
| Attribute .....                                                | 19 |
| Operationen .....                                              | 19 |
| Invarianten .....                                              | 20 |
| Definition im USE-Quelltext .....                              | 21 |
| Assoziationsbeschreibungen .....                               | 22 |
| VehicleOfBrand .....                                           | 22 |
| Assoziationsteile .....                                        | 22 |
| Definition im USE-Quelltext .....                              | 22 |
| VehicleOnStation .....                                         | 22 |
| Assoziationsteile .....                                        | 22 |
| Definition im USE-Quelltext .....                              | 22 |
| DriverDocuments .....                                          | 23 |
| Assoziationsteile .....                                        | 23 |
| Definition im USE-Quelltext .....                              | 23 |
| OpenRentalAgreement .....                                      | 23 |
| Assoziationsteile .....                                        | 23 |
| Definition im USE-Quelltext .....                              | 24 |
| Operationsbeschreibungen .....                                 | 25 |
| Klasse Brand Operation setBrandAttributes .....                | 25 |
| Definition im USE-Quelltext .....                              | 25 |
| Klasse Driver Operation associateDriverWithDocuments .....     | 25 |
| Definition im USE-Quelltext .....                              | 25 |
| Klasse RentalAgreement Operation checkOutRentalAgreement ..... | 26 |
| Definition im USE-Quelltext .....                              | 26 |
| Klasse RentalAgreement Operation checkInRentalAgreement .....  | 26 |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Definition im USE-Quelltext .....                               | 26 |
| Klasse Vehicle Operation insertVehicleIntoBrandAssociation..... | 27 |
| Definition im USE-Quelltext .....                               | 27 |
| Klasse Vehicle Operation checkInVehicle .....                   | 27 |
| Definition im USE-Quelltext .....                               | 27 |
| Klassendiagramm .....                                           | 28 |
| Testfälle .....                                                 | 29 |
| Quelltext des Start-Scripts .....                               | 29 |
| Kommandosequenzprotokoll .....                                  | 30 |
| Objektdiagramm nach dem Start-Script .....                      | 36 |
| Sequenz-Diagramm nach dem Start-Script .....                    | 37 |
| Testfall 1 (TC_01.cmd) .....                                    | 38 |
| Quelltext des Testfalls .....                                   | 38 |
| Kommandosequenzprotokoll .....                                  | 39 |
| Objekt Diagramm Testfall 1 nach dem Check-Out .....             | 42 |
| Objekt Diagramm Testfall 1 nach dem Check-In .....              | 43 |
| Testfall 2 (TC_02.cmd) .....                                    | 44 |
| Quelltext des Testfalls .....                                   | 44 |
| Kommandosequenzprotokoll .....                                  | 44 |
| Testfall 3 (TC_03.cmd) .....                                    | 45 |
| Quelltext des Testfalls .....                                   | 45 |
| Kommandosequenzprotokoll .....                                  | 45 |
| Testfall 4 (TC_04.cmd) .....                                    | 46 |
| Quelltext des Testfalls .....                                   | 46 |
| Kommandosequenzprotokoll .....                                  | 46 |
| Testfall 5 (TC_05.cmd) .....                                    | 47 |
| Quelltext des Testfalls .....                                   | 47 |
| Kommandosequenzprotokoll .....                                  | 47 |
| Testfall 6 (TC_06.cmd) .....                                    | 48 |
| Quelltext des Testfalls .....                                   | 48 |
| Kommandosequenzprotokoll .....                                  | 48 |

# Vorwort

Das Modell **CarRentalAgency** bildet einen kleinen Ausschnitt der Abläufe innerhalb einer Autovermietung ab.

Aus persönlichen Gründen verwenden wir die Abläufe und Bezeichnungen der Autovermietung „Europcar“, das Modell lässt sich aber auch auf jede andere Autovermietung übertragen, da die Arbeitsabläufe im wesentlichen identisch sind.

Das Modell erhebt selbstverständlich keinen Anspruch auf Vollständigkeit. Dafür bietet die vorliegende Hausarbeit nicht den notwendigen Umfang, die umfassende Bearbeitung dieses Themas wäre im Rahmen einer Diplomarbeit denkbar.

Um die Möglichkeit der Vergleiche für Datum und Uhrzeit herzustellen, wurde die Klasse `CurrentDate` eingeführt, deren Attribut `today` den aktuellen Unix-Date-Zeitstempel erhält, wenn die `start-dos.bat` (unter Windows-basierten Systemen) bzw. `start-linux.sh` (unter Linux-basierten Systemen) verwendet wird, um das Modell-Design-Tool USE mit dem entwickelten Modell zu starten.

## Modellbeschreibung

Das Modell beschreibt die Abläufe bei der Anmietung von Fahrzeugen.

Voraussetzungen für einen Mietvertrag sind: Ein Fahrzeug, welches für einen bestimmten Zeitraum angemietet werden kann, eine Person (im Modell als `Driver` bezeichnet) sowie zwei Stationen, die eine von der der Mietvertrag beginnt, die andere die das voraussichtliche Ziel für die Miete angibt.

Das Modell beachtet bei der Anmietung die Voraussetzungen des Vermietunternehmens. So ist für jedes Fahrzeug ein Mindestalter gespeichert, welches der Fahrer vorweisen muss um zur Miete des Fahrzeuges berechtigt zu sein. Ebenso darf ein Fahrzeug nur dann vermietet werden, wenn es durch einen Mitarbeiter zur Vermietung freigegeben wurde („`Ready For Rent`“) – der Anfangsstatus jedes Fahrzeuges wenn es aus einer Fahrzeugbewegung („`Movement`“) kommt ist „`To Check`“. Ferner gibt es Fahrzeuge, die dadurch dass sie ein bestimmtes „Alter“ (Tage seit In-Dienst-Stellung beim Unternehmen) oder einen bestimmten Kilometer-Stand erreicht haben als „`Requested For Sale`“ markiert werden. Nach dem Ende jedes Mietvertrages (beim „`Check In`“) wird geprüft, ob das Fahrzeug die genannten Kriterien erfüllt und dann automatisch als `Requested For Sale` markiert. Die Werte, die diese Kriterien beinhalten, ergeben sich aus der Fahrzeugmarke, welche als eigenes Objekt in dem Modell definiert wurden. Wenn ein Fahrzeug „`Blocked For Sale`“ ist, ist es für die Vermietung gesperrt.

Um einen Fahrer für die Vermietung verwenden zu können, müssen ihm seine Ausweisdokumente zugeordnet werden. Bei der Zuordnung wird überprüft, ob die angegebenen Dokumente bereits bzw. noch gültig sind. Ebenso wird bei den darauf folgenden Anmietungen selbst („`Check Out`“) geprüft ob die dem Fahrer zugeordneten Dokumente noch gültig sind.

Das Modell definiert Invarianten die die Konsistenz des Modells und der Abläufe erhalten sollen.

# Klassenbeschreibungen

Im Folgenden die Beschreibung der einzelnen Klassen im Modell und Ihre Bedeutung.

## ***CurrentDate***

Die Klasse *CurrentDate* dient zur Speicherung der aktuellen Datum/Uhrzeit Kombination als Integer-Wert in Sekunden seit der Unix-Epoche (01.01.1970 00:00:00 UTC).

### **Attribute**

| Attributname | Typ     | Beschreibung des Inhaltes     |
|--------------|---------|-------------------------------|
| today        | Integer | Sekunden seit der Unix-Epoche |

### **Operationen**

Diese Klasse enthält keine Operationen

### **Invarianten**

| Invariantenname            | Beschreibung der Funktion                                                                  |
|----------------------------|--------------------------------------------------------------------------------------------|
| CurrentDateGreaterThanZero | Das in dem Attribut <i>today</i> gespeicherte Datum in Sekunden muss größer als Null sein. |

### **Definition im USE-Quelltext**

```
class CurrentDate
  attributes
    today : Integer -- UnixDate
  end

  constraints

  context CurrentDate
    inv CurrentDateGreaterThanZero : CurrentDate.forAll-> ↵
      (self : CurrentDate | self.today > 0)
```

## Brand

Die Klasse Brand beschreibt die Eigenschaften der Fahrzeughersteller.

### Attribute

| Attributname       | Typ     | Beschreibung des Inhaltes                                  |
|--------------------|---------|------------------------------------------------------------|
| shortName          | String  | Kurzbezeichnung des Fahrzeugherstellers                    |
| mileageLimit       | Integer | vom Hersteller vorgeschriebene Kilometerbegrenzung         |
| daysInServiceLimit | Integer | vom Hersteller vorgeschriebene Begrenzung der Nutzungstage |

### Operationen

#### setBrandAttributes

Wird verwendet um die o.g. Attribute für das Brand-Objekt zu setzen. Daher sind auch die Attribute des Operationsaufrufes dieselben wie die Attribute des Objektes selbst.

#### PreConditions

|                                          |                                                                |
|------------------------------------------|----------------------------------------------------------------|
| <i>MileageLimitGreanterThanZero</i>      | Die eingegebene Kilometerbegrenzung muss größer als Null sein. |
| <i>DaysInServiceLimitGreaterThenZero</i> | Die Begrenzung der Nutzungstage muss größer als Null sein.     |

### Invarianten

Diese Klasse definiert keine Invarianten.

### Definition im USE-Quelltext

```
class Brand
attributes
  shortName : String
  mileageLimit : Integer
  daysInServiceLimit : Integer
operations
  setBrandAttributes (sShortName : String, iMileageLimit : Integer, ↵
    iDaysInServiceLimit : Integer)
end

constraints

context Brand :: setBrandAttributes (sShortName : String, iMileageLimit : ↵
  Integer, iDaysInServiceLimit : Integer)
  pre MileageLimitGreanterThanZero : iMileageLimit > 0
  pre DaysInServiceLimitGreaterThenZero : iDaysInServiceLimit > 0
```

## Station

Die Klasse Station dient als Beschreibung für die aktuelle Position (physisch und als Kostenstelle) für ein Fahrzeug bzw. für die Bezeichnung der voraussichtlichen Check-In-Station eines Mietvertrages.

### Attribute

| Attributname | Typ    | Beschreibung des Inhaltes                       |
|--------------|--------|-------------------------------------------------|
| shortName    | String | Kurzbezeichnung der Station („Six-Letter-Code“) |
| longName     | String | Bezeichnung der Station                         |

### Operationen

Diese Klasse enthält keine Operationen

### Invarianten

| Invariantenname        | Beschreibung der Funktion                                                                                                                                                                                                                                                                                                             |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stationShortNameUnique | Die Kurzbezeichnung der Station muss eindeutig sein, da über diese jede Station identifiziert wird. Diese Invariante ist ausreichend um eine Stationseindeutigkeit in dem Modell herzustellen. Der „longName“ der Stationen muss nicht eindeutig sein, da er nicht für die Operationen während Check-In und Check-Out verwendet wird. |

### Definition im USE-Quelltext

```
class Station
attributes
  shortName : String
  longName : String
end

constraints

context Station
  inv stationShortNameUnique : Station.allInstances-> ⚡
  forAll(s1, s2 | s1.shortName = s2.shortName implies s1 = s2)
```

## Vehicle

Vehicle ist eine abstrakte Klasse, welche alle für den Betrieb wichtigen Eigenschaften von Fahrzeugen beinhaltet. Sie wird an die Klassen „Car“ und „Truck“ (im deutschen „PKW“ und „LKW“) vererbt.

### Attribute

| Attributname           | Typ     | Beschreibung des Inhaltes                                                                                                                                                                                                                                                                                                   |
|------------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sipp                   | String  | Mietwagenklassifizierung                                                                                                                                                                                                                                                                                                    |
| minAgeForRental        | Integer | Mindestalter des Mieters in Jahren                                                                                                                                                                                                                                                                                          |
| mileage                | Integer | Aktueller Kilometerstand                                                                                                                                                                                                                                                                                                    |
| saleFlag               | Enum    | Angabe ob ein Fahrzeug nicht zum Verkauf steht, zum Verkauf angefragt oder zum Verkauf blockiert ist. Mögliche Werte: #NotForSale (Fahrzeug steht nicht zum Verkauf), #RequestedForSale (Fahrzeug zum Verkauf angefragt), #BlockedForSale (Fahrzeug zum Verkauf blockiert)                                                  |
| theftFlag              | Boolean | Angabe ob ein Fahrzeug als gestohlen markiert ist oder nicht. Mögliche Werte: false (Fahrzeug ist nicht gestohlen), true (Fahrzeug ist gestohlen und befindet sich nur noch im System um die ihm zugeordneten Informationen nicht zu verlieren)                                                                             |
| registrationNumber     | String  | Polizeiliches Kennzeichen                                                                                                                                                                                                                                                                                                   |
| fuelType               | Enum    | Kraftstofftyp. Mögliche Werte: #Diesel, #Super                                                                                                                                                                                                                                                                              |
| inServiceDate          | Integer | Unix-Date des Tages der In-Dienst-Stellung des Fahrzeuges beim Unternehmen                                                                                                                                                                                                                                                  |
| unitNumber             | Integer | Eindeutige Fahrzeugnummer                                                                                                                                                                                                                                                                                                   |
| modelDesc              | String  | Menschenlesbare Modellbezeichnung des Fahrzeuges                                                                                                                                                                                                                                                                            |
| vehicleMovementType    | Enum    | Angabe ob und welchen Typ von Fahrzeugbewegung das Fahrzeug zurzeit macht. Mögliche Werte: #NOM (No Movement, Fahrzeug bewegt sich nicht), #ORA (Open Rental Agreement, Laufender Mietvertrag), #CTM (Car Transfer Movement, Fahrzeugbewegung von einer zu einer anderen Station durch einen Angestellten des Unternehmens) |
| vehicleStatusType      | Enum    | Angabe ob ein Fahrzeug zur Vermietung freigegeben ist. Mögliche Werte: #TC (To be Checked, Fahrzeug muss von einem Mitarbeiter auf seine Vermietfähigkeit hin überprüft werden), #RR (Ready for Rent, Fahrzeug ist zur Vermietung freigegeben)                                                                              |
| vehicleLastMovementEnd | Integer | Ende der letzten Fahrzeugbewegung als Unix-Date.                                                                                                                                                                                                                                                                            |

## Operationen

### checkInVehicle

Wird verwendet um dem Fahrzeug welches aus einem laufenden Mietvertrag kommt seinen neuen Kilometerstand zuzuordnen, sowie es der aktuellen Station in der Assoziation „VehicleOnStation“ zuzuordnen.

#### Attribute

|                       |                                                                            |
|-----------------------|----------------------------------------------------------------------------|
| <i>newMileage</i>     | Neuer Kilometerstand des Fahrzeugs                                         |
| <i>checkInStation</i> | Station an der das Fahrzeug eingecheckt wird                               |
| <i>curDate</i>        | Aktuelles Datum um es in „vehicleLastMovementEnd“ des Fahrzeugs zu setzen. |

### insertVehicleIntoBrandAssociation

Wird verwendet um ein Fahrzeug einem Fahrzeughersteller zuzuordnen, damit auf die Verkaufsvoraussetzungen des Herstellers zugegriffen werden kann.

#### Attribute

|              |                                                             |
|--------------|-------------------------------------------------------------|
| <i>Brand</i> | Objekt vom Typ Brand die den Fahrzeughersteller beschreibt. |
|--------------|-------------------------------------------------------------|

### checkSaleVehicle

Diese Operation wird verwendet, um beim Check-In des Fahrzeugs den Verkaufs-Status des Fahrzeuges neu zu setzen. In der Regel würde die Prüfung, ob das Fahrzeug seine Nutzungstagegrenze erreicht hat, in einem realen System periodisch jeden Tag erneut geprüft werden und nicht erst, nachdem ein Fahrzeug eine Fahrzeugbewegung beendet hat. Da sich hier jedoch keine voranschreitende Zeit und keine periodischen sich wiederholenden Abläufe modelliert werden können, wurde diese Verwendung gewählt.

#### Attribute

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <i>curDate</i> | Aktuelles Datum um es in „vehicleLastMovementEnd“ des Fahrzeugs zu setzen. |
|----------------|----------------------------------------------------------------------------|

#### Rückgabewert

Die Rückgabe der Funktion ist ein Wert aus der Enumeration SaleFlags. Wenn der aktuelle Kilometerstand größer oder gleich der durch die Fahrzeugmarke vorgegebenen Kilometerbegrenzung oder das Delta zwischen dem aktuellen Datum (angegeben durch das Objekt curDate vom Typ CurrentDate) und dem Datum der In-Dienst-Stellung größer oder gleich der Nutzungstage-Beschränkung welche ebenfalls durch die Fahrzeugmarke vorgegeben wird, ist der Rückgabewert #RequestedForSale, anderenfalls #NotForSale. Zu beachten ist hier, dass das Delta in Sekunden angegeben wird, die Nutzungstage-Beschränkung jedoch in Tagen. Daher werden die Tage in der Funktion mit den Sekunden pro Tag (86400) multipliziert.

## **automaticReadyForRentAfterFiveDays**

Diese Funktion wird verwendet, um vor dem Beginn eines Check-Outs ein Fahrzeug, das noch den vehicleStatus #TC hat automatisch zur Vermietung freizugeben, wenn die letzte Fahrzeugbewegung vor 5 Tagen ( $5 * 86400$  Sekunden) beendet wurde, da davon ausgegangen werden kann, dass ein Fahrzeug innerhalb von 5 Tagen mindestens einmal überprüft wurde und damit auch als vermietfähig gilt.

### Attribute

*curDate*

Aktuelles Datum für den Vergleich mit der letzten Fahrzeugbewegung

### Rückgabewert

Die Rückgabe der Funktion ist ein Wert aus der Enumeration VehicleStatusTypes.

## **Invarianten**

### **Invariantenname**

### **Beschreibung der Funktion**

mileageNotNegative

Der aktuelle Kilometerstand eines Fahrzeugs darf nicht negativ sein. Diese Invariante sollte immer OK sein, da auf Grund der PreConditions von kilometerstandsverändernden Funktionen der zu addierende Kilometerstand immer positiv ist. Diese Invariante schlägt also nur fehl, wenn der Kilometerstand manuell auf einen negativen Wert gesetzt wird.

vehicleUnitNumberUnique

Die vehicleUnitNumber ist die eindeutige Bezeichnung eines Fahrzeugs (die registrationNumber kann auf Grund von länderspezifischen Gegebenheiten nicht als eindeutige Identifizierung verwendet werden) und muss daher im Modell eindeutig sein.

vehicleToBeCheckedImpliesNoOpenRentalAgreement

Wenn der vehicleStatusType #TC (To be Checked) ist, darf der vehicleMovementType nicht #ORA (Open Rental Agreement) sein. Durch eine PreCondition der Check-Out-Operation wird verhindert, dass Fahrzeuge, die noch nicht zur Vermietung freigegeben sind, für einen Mietvertrag verwendet werden. Diese Invariante schlägt also nur fehl, wenn der Kilometerstand manuell auf einen negativen Wert gesetzt wird.

vehicleBlockedForSaleImpliesNoOpenRentalAgreement

Wie vorstehend – allerdings mit dem Attribut saleFlag des Fahrzeugs das den Wert #BlockedForSale hat.

vehicleOnTheRoadImpliesNotOnAStation

Wenn sich das Fahrzeug in einem laufenden Mietvertrag befindet (vehicleMovementType = #ORA) bedeutet das, dass es sich nicht auf einer Station (Assoziation „VehicleOnStation“) befinden kann, da beim Check-Out des Fahrzeugs diese Zuordnung gelöscht wird. Beim Check-In wird der VehicleMovementType auf #TC (To be Checked) gesetzt und die Zuordnung mit der Check-In-Station hergestellt. Schlägt diese Invariante fehl, wurde manuell eine Zuordnung hinzugefügt.

## Definition im USE-Quelltext

```
abstract class Vehicle
attributes
  sipp : String
  minAgeForRental : Integer -- Years
  mileage : Integer
  saleFlag : SaleFlags
  theftFlag : Boolean
  registrationNumber : String
  fuelType : FuelTypes
  inServiceDate : Integer -- UnixDate
  unitNumber : Integer
  modelDesc : String
  vehicleMovementType : VehicleMovementTypes
  vehicleStatusType : VehicleStatusTypes
  vehicleLastMovementEnd : Integer -- UnixDate
operations
  checkInVehicle(newMileage : Integer, checkInStation : Station, curDate : ←
  CurrentDate)
  insertVehicleIntoBrandAssociation (brand : Brand)
  checkSaleVehicle(curDate : CurrentDate) : SaleFlags = ←
    if (self.saleFlag = #NotForSale) and ((self.mileage >= ←
    self.brand.mileageLimit) or ((curDate.today-self.inServiceDate) >= ←
    (self.brand.daysInServiceLimit*86400))) then #RequestedForSale else ←
    self.saleFlag endif
  automaticReadyForRentAfterFiveDays(curDate : CurrentDate) : ←
  VehicleStatusTypes = ←
    if self.vehicleStatusType = #TC and (self.vehicleLastMovementEnd + ←
    (5*86400) <= curDate.today) then #RR else #TC endif

end

constraints

context Vehicle
  inv mileageNotNegative : Vehicle.allInstances->forall(self : Vehicle | ←
  self.mileage > 0)
  inv vehicleUnitNumberUnique : Vehicle.allInstances->forall(v1, v2 | ←
  v1.unitNumber = v2.unitNumber implies v1 = v2)
  inv vehicleToBeCheckedImpliesNoOpenRentalAgreement : ←
  Vehicle.allInstances->forall(self : Vehicle | self.vehicleStatusType = ←
  #TC implies self.vehicleMovementType = #NOM)
```

```
inv vehicleBlockedForSaleImpliesNoOpenRentalAgreement : ↵  
Vehicle.allInstances->forall(self : Vehicle | self.saleFlag = ↵  
#BlockedForSale implies self.vehicleMovementType = #NOM)  
inv vehicleOnTheRoadImpliesNotOnAStation : Vehicle.allInstances-> ↵  
forall(self : Vehicle | self.vehicleMovementType = #ORA implies ↵  
self.currentStation->isEmpty())
```

## **Car**

Diese Klasse erbt von der abstrakten Klasse Vehicle und wird für Fahrzeuge verwendet, die hauptsächlich dem Personentransport zugeordnet werden (PKW).

### **Attribute**

| <b>Attributname</b> | <b>Typ</b> | <b>Beschreibung des Inhaltes</b>             |
|---------------------|------------|----------------------------------------------|
| maxPassengers       | Integer    | Max. Anzahl an Fahrzeuginsassen inkl. Fahrer |

### **Operationen**

Diese Klasse erbt die Operationen der abstrakten Klasse Vehicle und definiert daher keine eigenen.

### **Invarianten**

Es gibt keine Invarianten die sich speziell auf den Kontext Car beziehen, denn die Invarianten für den Kontext Vehicle sind allgemeiner und daher ausreichend.

### **Definition im USE-Quelltext**

```
class Car < Vehicle
  attributes
    maxPassengers : Integer
end
```

## **Truck**

Diese Klasse erbt von der abstrakten Klasse Vehicle und wird für Fahrzeuge verwendet, die hauptsächlich dem Warentransport zugeordnet werden (LKW).

### **Attribute**

| <b>Attributname</b> | <b>Typ</b> | <b>Beschreibung des Inhaltes</b>       |
|---------------------|------------|----------------------------------------|
| maxLoadTons         | Real       | Zulässiges Gesamtgewicht des Fahrzeugs |

### **Operationen**

Diese Klasse erbt die Operationen der abstrakten Klasse Vehicle und definiert daher keine eigenen.

### **Invarianten**

Es gibt keine Invarianten die sich speziell auf den Kontext Truck beziehen, denn die Invarianten für den Kontext Vehicle sind allgemeiner und daher ausreichend.

### **Definition im USE-Quelltext**

```
class Truck < Vehicle
  attributes
    maxLoadTons : Real
end
```

## Driver

Die Klasse Driver wird verwendet um eine natürliche Person zu beschreiben die ein Fahrzeug mietet und fährt. Von den betrieblichen Feinheiten wie der Unterscheidung zwischen Mieter und Fahrer wird hier abgesehen.

### Attribute

| Attributname | Typ     | Beschreibung des Inhaltes               |
|--------------|---------|-----------------------------------------|
| driverID     | String  | Eindeutige Identifikation eines Fahrers |
| name         | String  | Vorname                                 |
| surname      | String  | Nachname                                |
| address      | String  | Anschrift                               |
| phoneNumber  | String  | Kontakttelefonnummer                    |
| birthDay     | Integer | Geburtsdatum in Sekunden (UnixDate)     |

### Operationen

#### associateDriverWithDocuments

Wird verwendet um einen Fahrer mit seinen Ausweisdokumenten (Ausweis der Klasse „IdentityCard“ und Führerschein der Klasse „DrivingLicense“, Beschreibungen folgen) in die Assoziation „DriverDocuments“ einzufügen

#### Attribute

|                   |                                                                                  |
|-------------------|----------------------------------------------------------------------------------|
| <i>idcard</i>     | Objekt der Klasse IdentityCard welches des Ausweis des Fahrers beschreibt        |
| <i>drivinglic</i> | Objekt der Klasse DrivingLicense welches den Führerschein des Fahrers beschreibt |
| <i>curDate</i>    | Objekt der Klasse CurrentDate um die Pre-Conditions prüfen zu können.            |

#### Pre-Conditions

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>driverHasReachedMinAge</i> | Die Unternehmensvorgaben sagen, dass ein Fahrer mindestens 19 Jahre alt sein muss um ein Fahrzeug mieten zu dürfen. Es wird nun also geprüft, ob das Delta zwischen dem heutigen Datum und dem eingetragenen Geburtsdatum des Fahrers gleich oder mehr als 19 Jahre (genauer $19 * 365 * 86400$ Sekunden) beträgt. Die Ungenauigkeit in der Berechnung, bedingt durch mindestens 4 Schaltjahre innerhalb von 19 Jahren wird durch die Prämisse eines aufmerksamen Systemanwenders in Kauf genommen. |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

*driverNotYoungerThanHisDocuments*

Durch Eingabefehler entstehen seltsame Ergebnisse, diese Vorbedingung verhindert, dass der Gültigkeitsbeginn der Dokumente des Fahrers vor seinem Geburtsdatum liegt.

*idCardCurrentlyValid, drivingLicenseCurrentlyValid*

Diese beiden Vorbedingungen prüfen, ob die Ausweisdokumente zum gegenwärtigen Zeitpunkt gültig sind.

## Invarianten

### Invariantenname

driverIDUnique

### Beschreibung der Funktion

Die Driver ID ist das eindeutige Erkennungsmerkmal eines Fahrers (die Kombination aus Name, Vorname und Geburtsdatum ist nicht ausreichend) und muss daher im Modell eindeutig sein.

## Definition im USE-Quelltext

```
class Driver
  attributes
    driverID : String
    name : String
    surname : String
    address : String
    phoneNumber : String
    birthDay : Integer -- UnixDate
  operations
    associateDriverWithDocuments(idcard : IdentityCard, drivinglic : ←
      DrivingLicense, curDate : CurrentDate)
  end

  constraints

  context Driver :: associateDriverWithDocuments(idcard : IdentityCard, ←
    drivinglic : DrivingLicense, curDate : CurrentDate)
    pre driverHasReachedMinAge : (curDate.today-(19*365*86400)) >= ←
      self.birthDay
    pre driverNotYoungerThanHisDocuments : self.birthDay < idcard.validFrom ←
      and self.birthDay < drivinglic.validFrom
    pre idCardCurrentlyValid : idcard.validFrom <= curDate.today and ←
      idcard.validTill >= curDate.today
    pre drivingLicenseCurrentlyValid : drivinglic.validFrom <= ←
      curDate.today and drivinglic.validTill >= curDate.today

  context Driver
    inv driverIDUnique : Driver.allInstances->forall(d1, d2 | d1.driverID = ←
      d2.driverID implies d1 = d2)
```

## ***IdentityCard***

Diese Klasse wird verwendet um die Daten des Ausweisdokuments eines Fahrers zu speichern.

### **Attribute**

| <b>Attributname</b>  | <b>Typ</b> | <b>Beschreibung des Inhaltes</b>         |
|----------------------|------------|------------------------------------------|
| country              | String     | Ausstellungsland                         |
| numberOfIdentityCard | String     | Ausweisnummer                            |
| validFrom            | Integer    | Gültigkeitsbeginn in Sekunden – UnixDate |
| validTill            | Integer    | Gültigkeitsende in Sekunden – UnixDate   |

### **Operationen**

Diese Klasse definiert keine Operationen.

### **Invarianten**

| <b>Invariantenname</b>     | <b>Beschreibung der Funktion</b>                                                                              |
|----------------------------|---------------------------------------------------------------------------------------------------------------|
| numberOfIdentityCardUnique | Die Ausweisnummer ist das eindeutige Erkennungsmerkmal des Ausweises und muss daher im Modell eindeutig sein. |

### **Definition im USE-Quelltext**

```
class IdentityCard
  attributes
    country : String
    numberOfIdentityCard : String
    validFrom : Integer -- UnixDate
    validTill : Integer -- UnixDate
  end

  constraints

  context IdentityCard
    inv numberOfIdentityCardUnique : IdentityCard.allInstances-> ⌊
      forAll(i1, i2 | i1.numberOfIdentityCard = i2.numberOfIdentityCard ⌊
        implies i1 = i2)
```

## DrivingLicense

Diese Klasse wird verwendet um die Daten des Fahrausweisdokuments eines Fahrers zu speichern.

### Attribute

| Attributname           | Typ     | Beschreibung des Inhaltes                |
|------------------------|---------|------------------------------------------|
| country                | String  | Ausstellungsland                         |
| numberOfDrivingLicense | String  | Fahrausweisnummer                        |
| validFrom              | Integer | Gültigkeitsbeginn in Sekunden – UnixDate |
| validTill              | Integer | Gültigkeitsende in Sekunden – UnixDate   |

### Operationen

Diese Klasse definiert keine Operationen.

### Invarianten

| Invariantenname              | Beschreibung der Funktion                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------|
| numberOfDrivingLicenseUnique | Die Fahrausweisnummer ist das eindeutige Erkennungsmerkmal des Ausweises und muss daher im Modell eindeutig sein. |

### Definition im USE-Quelltext

```
class DrivingLicense
  attributes
    country : String
    numberOfDrivingLicense : String
    validFrom : Integer -- UnixDate
    validTill : Integer -- UnixDate
  end

  constraints

  context DrivingLicense
    inv numberOf DrivingLicense Unique : DrivingLicense.allInstances-> ↵
    forAll(d1, d2 | d1.numberOf DrivingLicense = d2.numberOf DrivingLicense ↵
    implies d1 = d2)
```

## RentalAgreement

Diese Klasse speichert alle Informationen zu einem Mietvertrag.

### Attribute

| Attributname      | Typ     | Beschreibung des Inhaltes                                                                                                                                                                   |
|-------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RANo              | Integer | Eindeutige Nummer des Mietvertrages                                                                                                                                                         |
| PaymentType       | Enum    | Zahlungsmethode bei Anmietung. Mögliche Werte: #Cash (Bezahlung mit Bargeld), #ECCash (Bezahlung mit EurochequeCard), #CreditCard (Authorisierung einer Kreditkarte mit späterer Bezahlung) |
| distanceTravelled | Integer | Gefahrene Entfernung in Kilometern, zu Abrechnungszwecken                                                                                                                                   |
| checkOutDate      | Integer | Datum/Uhrzeit des Check-Outs des Mietvertrages – UnixDate                                                                                                                                   |
| checkInDate       | Integer | Datum/Uhrzeit des Check-Ins des Mietvertrages – UnixDate                                                                                                                                    |

### Operationen

#### checkOutRentalAgreement

Wird verwendet um den Check-Out-Prozess durchzuführen.

#### Attribute

|                       |                                                                                                                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>RANo</i>           | Eindeutige Nummer des Mietvertrages. Diese wird direkt im RentalAgreement-Objekt gespeichert.                                                                                                                                         |
| <i>veh</i>            | Objekt der Klasse Vehicle, da Vehicle eine abstrakte Klasse ist und ihre Eigenschaften an die Klassen Car und Truck vererbt, ein Objekt der Klasse Car oder Truck, welches das Fahrzeug beschreibt das im Mietvertrag verwendet wird. |
| <i>driver</i>         | Objekt der Klasse Driver, welches den Mieter und Hauptfahrer beschreibt.                                                                                                                                                              |
| <i>curDate</i>        | Objekt der Klasse CurrentDate um die Pre-Conditions prüfen zu können.                                                                                                                                                                 |
| <i>checkInStation</i> | Objekt der Klasse Station um in der Assoziation „OpenRentalAgreement“ die voraussichtliche Check-In-Station verknüpfen zu können.                                                                                                     |
| <i>checkInDate</i>    | Integer-Wert, Unix-Date des voraussichtlichen Check-In-Datum an der voraussichtlichen Check-In-Station.                                                                                                                               |
| <i>payType</i>        | Wert aus der Enumeration PaymentTypes, damit beim Check-In des Fahrzeugs eine korrekte Abrechnung erfolgen kann.                                                                                                                      |

### Pre-Conditions

|                                          |                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>driverFulfilsAgeRequirement</i>       | Die Unternehmensvorgaben sagen, dass ein Fahrer ein bestimmtes Mindestalter erreicht haben muss, um ein Fahrzeug einer bestimmten Klasse fahren zu dürfen. Dieses Alter ist in dem Fahrzeug im Attribut „minAgeForRental“ gespeichert. Es wird berechnet ob der Fahrer das Mindestalter (angegeben in Jahren) erreicht hat. |
| <i>vehicleIsNotOnTheRoad</i>             | Wenn sich ein Fahrzeug bereits in einer Fahrzeugbewegung befindet kann es nicht für einen Mietvertrag verwendet werden.                                                                                                                                                                                                     |
| <i>vehicleNotBlockedForSale</i>          | Zum Verkauf blockierte Fahrzeuge dürfen nicht für einen Mietvertrag verwendet werden.                                                                                                                                                                                                                                       |
| <i>vehicleMustBeReadyForRent</i>         | Das Fahrzeug muss durch einen Mitarbeiter zur Vermietung freigegeben worden sein.                                                                                                                                                                                                                                           |
| <i>driversDrivingLicenseStillValid</i>   | Das Fahrausweisdokument des Fahrers muss gültig sein.                                                                                                                                                                                                                                                                       |
| <i>driversIdCardStillValid</i>           | Auch das Ausweisdokument des Fahrers muss gültig sein.                                                                                                                                                                                                                                                                      |
| <i>checkInDateNotLessThanCurrentDate</i> | Das voraussichtliche Check-In-Datum darf nicht vor dem jetzigen Datum liegen.                                                                                                                                                                                                                                               |

### **checkInRentalAgreement**

Wird verwendet um den Check-In-Prozess durchzuführen.

#### Attribute

|                       |                                                                                                                                                                                  |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>checkInStation</i> | Objekt der Klasse Station, denn dir voraussichtlich Check-In-Station die bei Check-Out angegeben wurde muss nicht der realen Check-In-Station entsprechen. (sog. One-Way-Mieten) |
| <i>vehCurMileage</i>  | Integer-Wert der den aktuellen Kilometer-Stand des Fahrzeugs angibt.                                                                                                             |
| <i>curDate</i>        | Objekt der Klasse CurrentDate um dieses als Ende einer Fahrzeugbewegung in das Fahrzeug-Objekt speichern zu können.                                                              |

### Pre-Conditions

|                                                      |                                                                                           |
|------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <i>vehicleCurrentMileageGreaterThanAtRentalBegin</i> | Intuitiv, der aktuelle Kilometerstand beim Check-In muss größer sein, als beim Check-Out. |
|------------------------------------------------------|-------------------------------------------------------------------------------------------|

## **Invarianten**

### **Invariantenname**

numberOfRentalAgreementUnique

### **Beschreibung der Funktion**

Die Nummer des Mietvertrages muss im Modell eindeutig sein.

## Definition im USE-Quelltext

```
class RentalAgreement
attributes
  RANo : Integer
  PaymentType : PaymentTypes
  distanceTravelled : Integer
  checkOutDate : Integer -- UnixDate
  checkInDate : Integer -- UnixDate
operations
  checkOutRentalAgreement (RANo : Integer, veh : Vehicle, drv : Driver, ↵
    curDate : CurrentDate, checkInStation : Station, checkInDate : Integer, ↵
    payType : PaymentTypes)
  checkInRentalAgreement (checkInStation : Station, vehCurMileage : ↵
    Integer, curDate : CurrentDate)
end
constraints

context RentalAgreement :: checkOutRentalAgreement (RANo : Integer, ↵
  veh : Vehicle, drv : Driver, curDate : CurrentDate, checkInStation : ↵
  Station, checkInDate : Integer, payType : PaymentTypes)
  pre driverFulfilsAgeRequirement : (curDate.today - ↵
    (veh.minAgeForRental*365*86400)) >= drv.birthDay
  pre vehicleIsNotOnTheRoad : veh.vehicleMovementType = #NOM
  pre vehicleNotBlockedForSale : veh.saleFlag <> #BlockedForSale
  pre vehicleMustBeReadyForRent : veh.vehicleStatusType = #RR
  pre driversDrivingLicenseStillValid : drv.drivinglic->asSequence-> ↵
    at(1).validTill >= curDate.today
  pre driversIdCardStillValid : drv.idcard->asSequence-> ↵
    at(1).validTill >= curDate.today
  pre checkInDateNotLessThanCurrentDate : checkInDate >= curDate.today

context RentalAgreement :: checkInRentalAgreement (checkInStation : ↵
  Station, vehCurMileage : Integer, curDate : CurrentDate)
  pre vehicleCurrentMileageGreaterThanAtRentalBegin : ↵
    vehCurMileage > self.vehicleontheroad->asSequence->at(1).mileage

context RentalAgreement
  inv numberOfRentalAgreementUnique : RentalAgreement.allInstances-> ↵
    forAll(r1, r2 | r1.RANo = r2.RANo implies r1 = r2)
```

# Assoziationsbeschreibungen

Im Folgenden die Beschreibung der Assoziationen die In dem Modell verwendet werden.

## ***VehicleOfBrand***

Diese Assoziation wird verwendet um die Eigenschaften bezüglich der Nutzungsbeschränkungen eines Fahrzeugherstellers einem Fahrzeug zuzuordnen und so nach jeder Fahrzeugbewegung diese zu prüfen.

### **Assoziationsteile**

| Objekt  | Multiplizität | Beschreibung                                                |
|---------|---------------|-------------------------------------------------------------|
| Brand   | 1             | Einem Fahrzeughersteller ...                                |
| Vehicle | 0..*          | ... werden kein oder unbegrenzt viele Fahrzeuge zugeordnet. |

### **Definition im USE-Quelltext**

```
association VehicleOfBrand
between
    Vehicle[0..*] role vehicle
    Brand[1] role brand
end
```

## ***VehicleOnStation***

Diese Assoziation wird verwendet um die aktuelle physische (und Kostenstellen) Position eines Fahrzeugs zu speichern. Fahrzeuge werden beim Check-Out aus dieser Assoziation entfernt und beim Check-In wieder in die Assoziation mit ihrer neuen Station eingefügt.

### **Assoziationsteile**

| Objekt  | Multiplizität | Beschreibung                                                                                                       |
|---------|---------------|--------------------------------------------------------------------------------------------------------------------|
| Vehicle | 0..*          | Den Fahrzeugen (keinem bis unbegrenzt vielen) ...                                                                  |
| Station | 0..1          | ... werden keine (Fahrzeug in einer Fahrzeugbewegung) oder genau eine Station (keine Fahrzeugbewegung) zugeordnet. |

### **Definition im USE-Quelltext**

```
association VehicleOnStation
between
    Vehicle[0..*] role vehicle
    Station[0..1] role currentStation
end
```

## ***DriverDocuments***

Diese Assoziation wird verwendet um einem Fahrer sein Ausweis- und Fahrausweisdokument zuzuordnen.

### **Assoziationsteile**

| <b>Objekt</b>  | <b>Multiplizität</b> | <b>Beschreibung</b> |
|----------------|----------------------|---------------------|
| Driver         | 0..1                 | Fahrer              |
| IdentityCard   | 0..1                 | Ausweisdokument     |
| DrivingLicense | 0..1                 | Fahrausweisdokument |

Die Multiplizität 0..1 beschreibt den Umstand dass nicht notwendigerweise alle Dokumente dem Fahrer zugeordnet sein müssen. Ist dies nicht der Fall, werden die Vorbedingungen der Check-Out-Operation fehlschlagen und der Fahrer erhält (in der realen Welt) das Auto nicht.

### **Definition im USE-Quelltext**

```
association DriverDocuments
between
    Driver[0..1] role driver
    IdentityCard[0..1] role idcard
    DrivingLicense[0..1] role drivinglic
end
```

## ***OpenRentalAgreement***

Diese Assoziation enthält alle Informationen zu einem laufenden Mietvertrag. Wird ein Mietvertrag ausgecheckt, werden die Objekte entsprechend gesetzt. Beim Check-In des Mietvertrages werden nur die Informationen aus dieser Assoziation wieder verwendet die für die Abrechnung im Unternehmen wichtig sind, alle anderen Informationen haben dann keine Bedeutung mehr.

### **Assoziationsteile**

| <b>Objekt</b>   | <b>Multiplizität</b> | <b>Beschreibung</b>                                                                                                                                                                  |
|-----------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RentalAgreement | 0..1                 | Der Mietvertrag                                                                                                                                                                      |
| Vehicle         | 0..1                 | Das Fahrzeug                                                                                                                                                                         |
| Driver          | 0..*                 | Einem Mietvertrag können mehrere Fahrer zugeordnet sein. Dies wird nur nicht von den Operationen des Modells unterstützt, daher tritt dieser Fall nur bei manueller Bearbeitung auf. |
| Station         | 0..1                 | Rolle: checkOutStation – Station von der das Fahrzeug ausgecheckt wurde.                                                                                                             |
| Station         | 0..1                 | Rolle: checkInStation – Voraussichtliche Check-In-Station                                                                                                                            |

## Definition im USE-Quelltext

```
association OpenRentalAgreement
between
    RentalAgreement[0..1] role rentalagreement
    Vehicle[0..1] role vehicleontheroad
    Driver[0..*] role driver
    Station[0..1] role checkOutStation
    Station[0..1] role checkInStation
end
```

# Operationsbeschreibungen

Im Folgenden die Beschreibung der Operationen

## ***Klasse Brand Operation setBrandAttributes***

Setzen der Attribute für die Fahrzeughersteller.

### **Definition im USE-Quelltext**

```
-- setBrandAttributes (sShortName : String, iMileageLimit : Integer, ↵  
    iDaysInServiceLimit)  
  
!set self.shortName := sShortName  
!set self.mileageLimit := iMileageLimit  
!set self.daysInServiceLimit := iDaysInServiceLimit
```

## ***Klasse Driver Operation associateDriverWithDocuments***

Einfügen der Objekte der Klassen Driver, IdentityCard und DrivingLicense in die Assoziation DriverDocuments.

### **Definition im USE-Quelltext**

```
-- associateDriverWithDocuments(idcard : IdentityCard, drivinglic : ↵  
    DrivingLicense, curDate : CurrentDate)  
  
!insert (self, idcard, drivinglic) into DriverDocuments
```

## ***Klasse RentalAgreement Operation checkOutRentalAgreement***

Check-Out-Prozess. Setzen der Attribute im RentalAgreement-Objekt, Einfügen der Objekte in die OpenRentalAgreement Assoziation, Setzen der Werte im Vehicle-Objekt und Entfernen von Vehicle und Station aus der Assoziation VehicleOnStation.

### **Definition im USE-Quelltext**

```
-- checkOutRentalAgreement (RANo : Integer, veh : Vehicle, drv : ↵  
    Driver, curDate : CurrentDate, checkInStation : Station, ↵  
    checkInDate : Integer, payType : PaymentTypes)  
  
!set self.RANo := RANo  
!set self.PaymentType := payType  
!set self.checkOutDate := curDate.today  
!set self.checkInDate := checkInDate  
!insert (self, veh, drv, veh.currentStation, checkInStation) into ↵  
    OpenRentalAgreement  
!set veh.vehicleMovementType := #ORA  
!set veh.vehicleStatusType := #ON  
!delete (veh, veh.currentStation) from VehicleOnStation
```

## ***Klasse RentalAgreement Operation checkInRentalAgreement***

Check-In-Prozess. Setzen der Attribute im RentalAgreement-Objekt, Aufrufen der checkInVehicle Operation im Vehicle Objekt und Entfernen der Objekte aus der OpenRentalAgreement-Assoziation

### **Definition im USE-Quelltext**

```
-- checkInRentalAgreement (checkInStation : Station, vehCurMileage : ↵  
    Integer, curDate : CurrentDate)  
  
!set self.distanceTravelled := vehCurMileage - ↵  
    self.vehicleontheroad->asSequence->at(1).mileage  
!set self.checkInDate := curDate.today  
!openter self.vehicleontheroad->asSequence->at(1) ↵  
    checkInVehicle(vehCurMileage, checkInStation, curDate)  
read Vehicle_checkInVehicle.cmd  
!opexit  
!delete (self, self.vehicleontheroad->asSequence->at(1), self.driver-> ↵  
    asSequence->at(1), self.checkOutStation->asSequence->at(1), ↵  
    self.checkInStation->asSequence->at(1)) from OpenRentalAgreement
```

## ***Klasse Vehicle Operation insertVehicleIntoBrandAssociation***

Einfügen des Fahrzeugs mit seinem Fahrzeughersteller in die Assoziation VehicleOfBrand sowie setzen der anfänglichen Fahrzeug-Attribute.

### **Definition im USE-Quelltext**

```
-- insertVehicleIntoBrandAssociation (brand : Brand)
!insert (self, brand) into VehicleOfBrand
!set self.vehicleMovementType := #NOM
!set self.vehicleStatusType := #TC
!set self.vehicleLastMovementEnd := 0
```

## ***Klasse Vehicle Operation checkInVehicle***

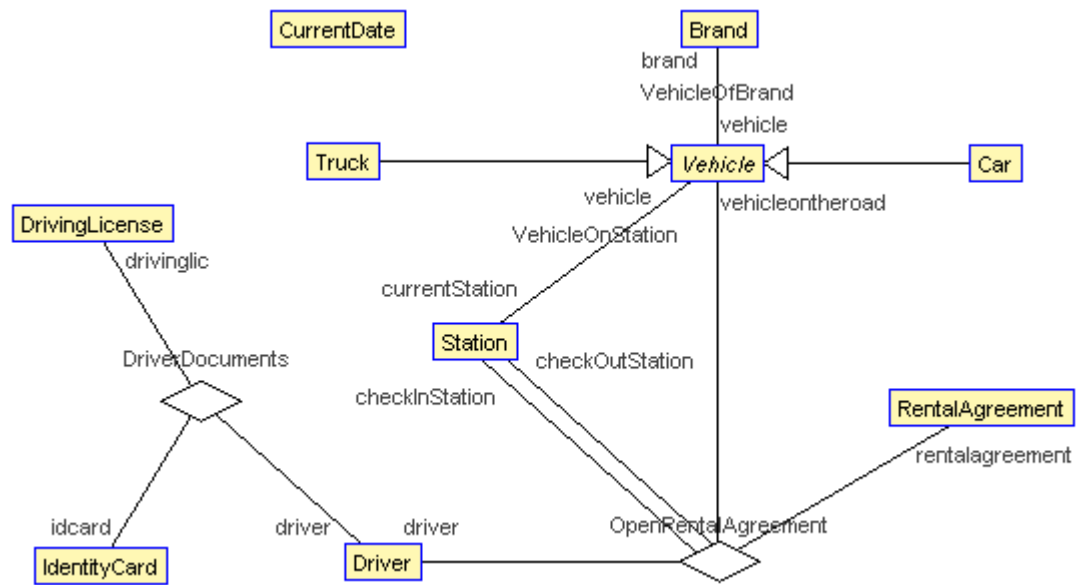
Setzen der Fahrzeug-Attribute sowie Einfügen des Fahrzeugs in die Assoziation VehicleOnStation mit der Check-In-Station.

### **Definition im USE-Quelltext**

```
-- checkInVehicle(newMileage : Integer, checkInStation : Station, ↵
    curDate : CurrentDate)

!set self.mileage := newMileage
!set self.vehicleMovementType := #NOM
!set self.vehicleStatusType := #TC
!set self.vehicleLastMovementEnd := curDate.today
!set self.saleFlag := self.checkSaleVehicle(curDate)
!insert (self, checkInStation) into VehicleOnStation
```

# Klassendiagramm



## Testfälle

Um die Testfälle zu erzeugen, wurde von uns ein Kommando-Script geschrieben, welches zunächst eine gewisse Menge von Objekten anlegt, die in den Test-Fällen verwendet werden. Alle Eigenschaften der Objekte hinterlassen einen gültigen Zustand (Alle Invarianten sind OK) in dem Modell.

Um einen negativen Test-Fall zu erzeugen werden dann bestimmte Attribute umgesetzt um die Funktion zu demonstrieren.

Das Kommando-Script wurde gemäß dem Modell CarRentalAgency-start.cmd genannt und befindet sich im Haupt-Verzeichnis der Modell-Quelltexte und greift auf die Dateien zu die sich im Ordner intern/ befinden.

### Quelltext des Start-Scripts

```
-- Reference for CurrentDate
!create Heute : CurrentDate
read intern/current-date.cmd

-- create Stations
read intern/CarRentalAgency-stations.cmd
-- create Brands
read intern/CarRentalAgency-brands.cmd
-- create Cars
read intern/CarRentalAgency-cars.cmd
-- create Drivers
read intern/CarRentalAgency-drivers.cmd

-- set Cars on Station

!insert (V80623665, BREC01) into VehicleOnStation
!insert (V79111294, BREC01) into VehicleOnStation
!insert (V80111875, BREC01) into VehicleOnStation
!insert (V81419972, BREC01) into VehicleOnStation
```

## Kommandosequenzprotokoll

```
CarRentalAgency-start.cmd> -- Reference for CurrentDate
CarRentalAgency-start.cmd> !create Heute : CurrentDate
CarRentalAgency-start.cmd> read intern/current-date.cmd
intern/current-date.cmd> !set Heute.today := 1221655085
intern/current-date.cmd>
CarRentalAgency-start.cmd>
CarRentalAgency-start.cmd> -- create Stations
CarRentalAgency-start.cmd> read intern/CarRentalAgency-stations.cmd
intern/CarRentalAgency-stations.cmd> -- create Stations
intern/CarRentalAgency-stations.cmd>
intern/CarRentalAgency-stations.cmd> !create BREC01 : Station
intern/CarRentalAgency-stations.cmd> !set BREC01.longName := 'Bremen City'
intern/CarRentalAgency-stations.cmd> !set BREC01.shortName := 'BREC01'
intern/CarRentalAgency-stations.cmd> !create BREL01 : Station
intern/CarRentalAgency-stations.cmd> !set BREL01.longName := 'Bremen Railway Station'
intern/CarRentalAgency-stations.cmd> !set BREL01.shortName := 'BREL01'
intern/CarRentalAgency-stations.cmd> !create BRET01 : Station
intern/CarRentalAgency-stations.cmd> !set BRET01.longName := 'Bremen Airport'
intern/CarRentalAgency-stations.cmd> !set BRET01.shortName := 'BRET01'
intern/CarRentalAgency-stations.cmd> !create OENC01 : Station
intern/CarRentalAgency-stations.cmd> !set OENC01.longName := 'Oldenburg City'
intern/CarRentalAgency-stations.cmd> !set OENC01.shortName := 'OENC01'
intern/CarRentalAgency-stations.cmd>
CarRentalAgency-start.cmd> -- create Brands
CarRentalAgency-start.cmd> read intern/CarRentalAgency-brands.cmd
intern/CarRentalAgency-brands.cmd> -- create Vehicle Brands
intern/CarRentalAgency-brands.cmd>
intern/CarRentalAgency-brands.cmd> !create MercedesBenz : Brand
intern/CarRentalAgency-brands.cmd> !create GeneralMotors : Brand
intern/CarRentalAgency-brands.cmd> !create Volkswagen : Brand
intern/CarRentalAgency-brands.cmd>
intern/CarRentalAgency-brands.cmd> !openter MercedesBenz setBrandAttributes('MB', 23800, 60)
precondition `MileageLimitGreanterThanZero' is true
precondition `DaysInServiceLimitGreaterThanZero' is true
intern/CarRentalAgency-brands.cmd> read Brand_setBrandAttributes.cmd
Brand_setBrandAttributes.cmd> -- setBrandAttributes (sShortName : String, iMileageLimit : Integer, iDaysInServiceLimit)
Brand_setBrandAttributes.cmd>
Brand_setBrandAttributes.cmd> !set self.shortName := sShortName
Brand_setBrandAttributes.cmd> !set self.mileageLimit := iMileageLimit
Brand_setBrandAttributes.cmd> !set self.daysInServiceLimit := iDaysInServiceLimit
Brand_setBrandAttributes.cmd>
intern/CarRentalAgency-brands.cmd> !opexit
intern/CarRentalAgency-brands.cmd> !openter GeneralMotors setBrandAttributes('GM', 25400, 180)
precondition `MileageLimitGreanterThanZero' is true
precondition `DaysInServiceLimitGreaterThanZero' is true
intern/CarRentalAgency-brands.cmd> read Brand_setBrandAttributes.cmd
```

```

Brand_setBrandAttributes.cmd> -- setBrandAttributes (sShortName : String, iMileageLimit : Integer, iDaysInServiceLimit)
Brand_setBrandAttributes.cmd>
Brand_setBrandAttributes.cmd> !set self.shortName := sShortName
Brand_setBrandAttributes.cmd> !set self.mileageLimit := iMileageLimit
Brand_setBrandAttributes.cmd> !set self.daysInServiceLimit := iDaysInServiceLimit
Brand_setBrandAttributes.cmd>
intern/CarRentalAgency-brands.cmd> !opexit
intern/CarRentalAgency-brands.cmd> !openter Volkswagen setBrandAttributes('VW', 25000, 180)
precondition `MileageLimitGreanterThanZero' is true
precondition `DaysInServiceLimitGreaterThanOrZero' is true
intern/CarRentalAgency-brands.cmd> read Brand_setBrandAttributes.cmd
Brand_setBrandAttributes.cmd> -- setBrandAttributes (sShortName : String, iMileageLimit : Integer, iDaysInServiceLimit)
Brand_setBrandAttributes.cmd>
Brand_setBrandAttributes.cmd> !set self.shortName := sShortName
Brand_setBrandAttributes.cmd> !set self.mileageLimit := iMileageLimit
Brand_setBrandAttributes.cmd> !set self.daysInServiceLimit := iDaysInServiceLimit
Brand_setBrandAttributes.cmd>
intern/CarRentalAgency-brands.cmd> !opexit
intern/CarRentalAgency-brands.cmd>
CarRentalAgency-start.cmd> -- create Cars
CarRentalAgency-start.cmd> read intern/CarRentalAgency-cars.cmd
intern/CarRentalAgency-cars.cmd> -- create Cars
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !create V80623665 : Car
intern/CarRentalAgency-cars.cmd> !set V80623665.unitNumber := 80623665
intern/CarRentalAgency-cars.cmd> !set V80623665.modelDesc := 'MERIV1.7D'
intern/CarRentalAgency-cars.cmd> !set V80623665.inServiceDate := 1214863200
intern/CarRentalAgency-cars.cmd> -- UnixDate for 2008-07-01
intern/CarRentalAgency-cars.cmd> !set V80623665.saleFlag := #NotForSale
intern/CarRentalAgency-cars.cmd> !set V80623665.theftFlag := false
intern/CarRentalAgency-cars.cmd> !set V80623665.mileage := 12859
intern/CarRentalAgency-cars.cmd> !set V80623665.sipp := 'IVMR'
intern/CarRentalAgency-cars.cmd> !set V80623665.minAgeForRental := 21
intern/CarRentalAgency-cars.cmd> !set V80623665.maxPassengers := 5
intern/CarRentalAgency-cars.cmd> !set V80623665.fuelType := #Super
intern/CarRentalAgency-cars.cmd> !set V80623665.registrationNumber := 'HH-AA9201'
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !openter V80623665 insertVehicleIntoBrandAssociation (GeneralMotors)
intern/CarRentalAgency-cars.cmd> read Vehicle_insertVehicleIntoBrandAssociation.cmd
Vehicle_insertVehicleIntoBrandAssociation.cmd> -- insertVehicleIntoBrandAssociation (brand : Brand)
Vehicle_insertVehicleIntoBrandAssociation.cmd> !insert (self, brand) into VehicleOfBrand
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleMovementType := #NOM
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleStatusType := #TC
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleLastMovementEnd := 0
Vehicle_insertVehicleIntoBrandAssociation.cmd>
intern/CarRentalAgency-cars.cmd> !opexit
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !create V79111294 : Car
intern/CarRentalAgency-cars.cmd> !set V79111294.unitNumber := 79111294
intern/CarRentalAgency-cars.cmd> !set V79111294.modelDesc := 'C 180K A'

```

```

intern/CarRentalAgency-cars.cmd> !set V79111294.inServiceDate := 1214863200
intern/CarRentalAgency-cars.cmd> -- UnixDate for 2008-07-01
intern/CarRentalAgency-cars.cmd> !set V79111294.saleFlag := #NotForSale
intern/CarRentalAgency-cars.cmd> !set V79111294.theftFlag := false
intern/CarRentalAgency-cars.cmd> !set V79111294.mileage := 18955
intern/CarRentalAgency-cars.cmd> !set V79111294.sipp := 'FDMR'
intern/CarRentalAgency-cars.cmd> !set V79111294.minAgeForRental := 25
intern/CarRentalAgency-cars.cmd> !set V79111294.maxPassengers := 5
intern/CarRentalAgency-cars.cmd> !set V79111294.fuelType := #Super
intern/CarRentalAgency-cars.cmd> !set V79111294.registrationNumber := 'HH-EC7301'
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !openter V79111294 insertVehicleIntoBrandAssociation (MercedesBenz)
intern/CarRentalAgency-cars.cmd> read Vehicle_insertVehicleIntoBrandAssociation.cmd
Vehicle_insertVehicleIntoBrandAssociation.cmd> -- insertVehicleIntoBrandAssociation (brand : Brand)
Vehicle_insertVehicleIntoBrandAssociation.cmd> !insert (self, brand) into VehicleOfBrand
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleMovementType := #NOM
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleStatusType := #TC
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleLastMovementEnd := 0
Vehicle_insertVehicleIntoBrandAssociation.cmd>
intern/CarRentalAgency-cars.cmd> !opexit
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !create V80111875 : Truck
intern/CarRentalAgency-cars.cmd> !set V80111875.unitNumber := 80111875
intern/CarRentalAgency-cars.cmd> !set V80111875.modelDesc := '211 KASTEN'
intern/CarRentalAgency-cars.cmd> !set V80111875.inServiceDate := 1214863200
intern/CarRentalAgency-cars.cmd> -- UnixDate for 2008-07-01
intern/CarRentalAgency-cars.cmd> !set V80111875.saleFlag := #NotForSale
intern/CarRentalAgency-cars.cmd> !set V80111875.theftFlag := false
intern/CarRentalAgency-cars.cmd> !set V80111875.mileage := 8243
intern/CarRentalAgency-cars.cmd> !set V80111875.sipp := 'VGHW'
intern/CarRentalAgency-cars.cmd> !set V80111875.minAgeForRental := 19
intern/CarRentalAgency-cars.cmd> !set V80111875.maxLoadTons := 3.5
intern/CarRentalAgency-cars.cmd> !set V80111875.fuelType := #Diesel
intern/CarRentalAgency-cars.cmd> !set V80111875.registrationNumber := 'HH-DB7723'
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !openter V80111875 insertVehicleIntoBrandAssociation (MercedesBenz)
intern/CarRentalAgency-cars.cmd> read Vehicle_insertVehicleIntoBrandAssociation.cmd
Vehicle_insertVehicleIntoBrandAssociation.cmd> -- insertVehicleIntoBrandAssociation (brand : Brand)
Vehicle_insertVehicleIntoBrandAssociation.cmd> !insert (self, brand) into VehicleOfBrand
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleMovementType := #NOM
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleStatusType := #TC
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleLastMovementEnd := 0
Vehicle_insertVehicleIntoBrandAssociation.cmd>
intern/CarRentalAgency-cars.cmd> !opexit
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !create V81419972 : Car
intern/CarRentalAgency-cars.cmd> !set V81419972.unitNumber := 81419972
intern/CarRentalAgency-cars.cmd> !set V81419972.modelDesc := 'TOUR 77DA'
intern/CarRentalAgency-cars.cmd> !set V81419972.inServiceDate := 1214863200
intern/CarRentalAgency-cars.cmd> -- UnixDate for 2008-07-01

```

```

intern/CarRentalAgency-cars.cmd> !set V81419972.saleFlag := #NotForSale
intern/CarRentalAgency-cars.cmd> !set V81419972.theftFlag := false
intern/CarRentalAgency-cars.cmd> !set V81419972.mileage := 8580
intern/CarRentalAgency-cars.cmd> !set V81419972.sipp := 'IWMR'
intern/CarRentalAgency-cars.cmd> !set V81419972.minAgeForRental := 21
intern/CarRentalAgency-cars.cmd> !set V81419972.maxPassengers := 5
intern/CarRentalAgency-cars.cmd> !set V81419972.fuelType := #Diesel
intern/CarRentalAgency-cars.cmd> !set V81419972.registrationNumber := 'HH-CI7655'
intern/CarRentalAgency-cars.cmd>
intern/CarRentalAgency-cars.cmd> !openter V81419972 insertVehicleIntoBrandAssociation (Volkswagen)
intern/CarRentalAgency-cars.cmd> read Vehicle_insertVehicleIntoBrandAssociation.cmd
Vehicle_insertVehicleIntoBrandAssociation.cmd> -- insertVehicleIntoBrandAssociation (brand : Brand)
Vehicle_insertVehicleIntoBrandAssociation.cmd> !insert (self, brand) into VehicleOfBrand
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleMovementType := #NOM
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleStatusType := #TC
Vehicle_insertVehicleIntoBrandAssociation.cmd> !set self.vehicleLastMovementEnd := 0
Vehicle_insertVehicleIntoBrandAssociation.cmd>
intern/CarRentalAgency-cars.cmd> !opexit
intern/CarRentalAgency-cars.cmd>
CarRentalAgency-start.cmd> -- create Drivers
CarRentalAgency-start.cmd> read intern/CarRentalAgency-drivers.cmd
intern/CarRentalAgency-drivers.cmd> -- Drivers
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create D57713381 : Driver
intern/CarRentalAgency-drivers.cmd> !set D57713381.driverID := '57713381'
intern/CarRentalAgency-drivers.cmd> !set D57713381.name := 'Kai'
intern/CarRentalAgency-drivers.cmd> !set D57713381.surname := 'Poppe'
intern/CarRentalAgency-drivers.cmd> !set D57713381.address := 'Bibliothekstr. 1, 28359 Bremen'
intern/CarRentalAgency-drivers.cmd> !set D57713381.phoneNumber := '+491234567890'
intern/CarRentalAgency-drivers.cmd> !set D57713381.birthDay := 0
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 1970-01-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create L57713381 : DrivingLicense
intern/CarRentalAgency-drivers.cmd> !set L57713381.country := 'DE'
intern/CarRentalAgency-drivers.cmd> !set L57713381.numberOfDrivingLicense := 'I123456789012'
intern/CarRentalAgency-drivers.cmd> !set L57713381.validFrom := 978300000
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2001-01-01
intern/CarRentalAgency-drivers.cmd> !set L57713381.validTill := 1293832800
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2011-01-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create I57713381 : IdentityCard
intern/CarRentalAgency-drivers.cmd> !set I57713381.country := 'DE'
intern/CarRentalAgency-drivers.cmd> !set I57713381.numberOfIdentityCard := '1234567890'
intern/CarRentalAgency-drivers.cmd> !set I57713381.validFrom := 1062367200
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2003-09-01
intern/CarRentalAgency-drivers.cmd> !set I57713381.validTill := 1222812000
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2008-10-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !openter D57713381 associateDriverWithDocuments(I57713381, L57713381, Heute)
precondition `driverHasReachedMinAge` is true

```

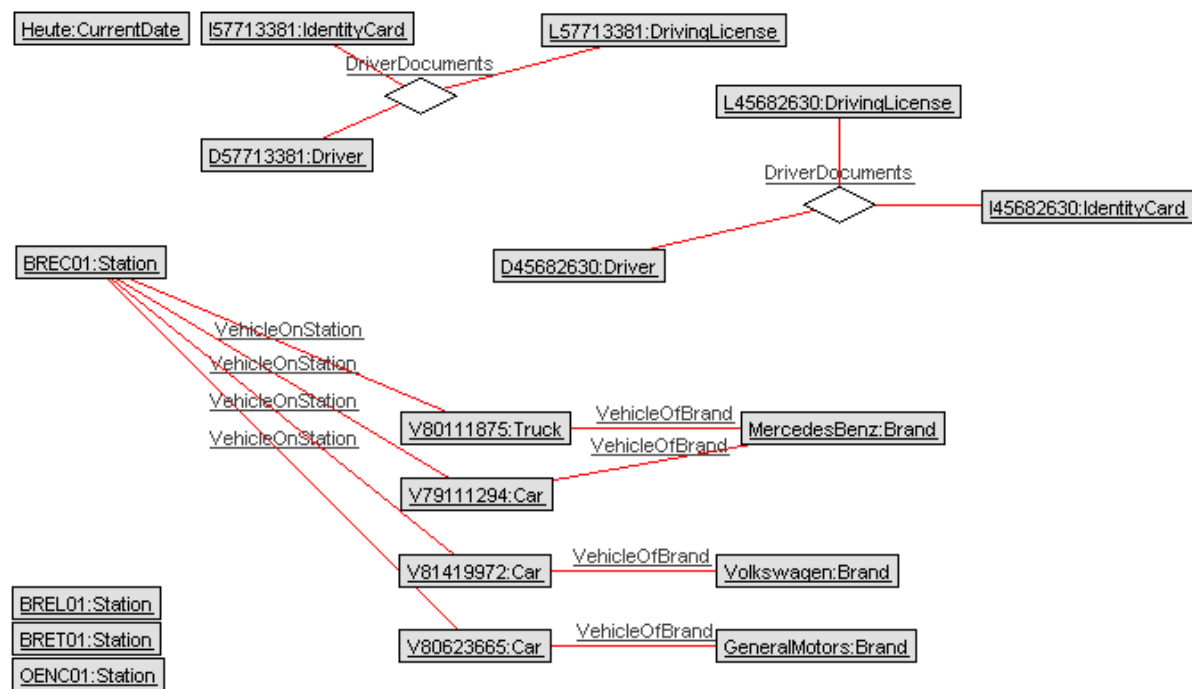
```

precondition `driverNotYoungerThanHisDocuments' is true
precondition `idCardCurrentlyValid' is true
precondition `drivingLicenseCurrentlyValid' is true
intern/CarRentalAgency-drivers.cmd> read Driver_associateDriverWithDocuments.cmd
Driver_associateDriverWithDocuments.cmd> -- associateDriverWithDocuments(idcard : IdentityCard, drivinglic : DrivingLicense, curDate :
CurrentDate)
Driver_associateDriverWithDocuments.cmd>
Driver_associateDriverWithDocuments.cmd> !insert (self, idcard, drivinglic) into DriverDocuments
Driver_associateDriverWithDocuments.cmd>
intern/CarRentalAgency-drivers.cmd> !opexit
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create D45682630 : Driver
intern/CarRentalAgency-drivers.cmd> !set D45682630.driverID := '45682630'
intern/CarRentalAgency-drivers.cmd> !set D45682630.name := 'Bastian'
intern/CarRentalAgency-drivers.cmd> !set D45682630.surname := 'Kammann'
intern/CarRentalAgency-drivers.cmd> !set D45682630.address := 'Bibliothekstr. 1, 28359 Bremen'
intern/CarRentalAgency-drivers.cmd> !set D45682630.phoneNumber := '+491234567890'
intern/CarRentalAgency-drivers.cmd> !set D45682630.birthDay := 0
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 1970-01-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create L45682630 : DrivingLicense
intern/CarRentalAgency-drivers.cmd> !set L45682630.country := 'DE'
intern/CarRentalAgency-drivers.cmd> !set L45682630.numberOfDrivingLicense := 'I123456789012'
intern/CarRentalAgency-drivers.cmd> !set L45682630.validFrom := 978300000
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2001-01-01
intern/CarRentalAgency-drivers.cmd> !set L45682630.validTill := 1293832800
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2011-01-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !create I45682630 : IdentityCard
intern/CarRentalAgency-drivers.cmd> !set I45682630.country := 'DE'
intern/CarRentalAgency-drivers.cmd> !set I45682630.numberOfIdentityCard := '1234567890'
intern/CarRentalAgency-drivers.cmd> !set I45682630.validFrom := 1062367200
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2003-09-01
intern/CarRentalAgency-drivers.cmd> !set I45682630.validTill := 1222812000
intern/CarRentalAgency-drivers.cmd> -- UnixDate for 2008-10-01
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd> !openter D45682630 associateDriverWithDocuments(I45682630, L45682630, Heute)
precondition `driverHasReachedMinAge' is true
precondition `driverNotYoungerThanHisDocuments' is true
precondition `idCardCurrentlyValid' is true
precondition `drivingLicenseCurrentlyValid' is true
intern/CarRentalAgency-drivers.cmd> read Driver_associateDriverWithDocuments.cmd
Driver_associateDriverWithDocuments.cmd> -- associateDriverWithDocuments(idcard : IdentityCard, drivinglic : DrivingLicense, curDate :
CurrentDate)
Driver_associateDriverWithDocuments.cmd>
Driver_associateDriverWithDocuments.cmd> !insert (self, idcard, drivinglic) into DriverDocuments
Driver_associateDriverWithDocuments.cmd>
intern/CarRentalAgency-drivers.cmd> !opexit
intern/CarRentalAgency-drivers.cmd>
intern/CarRentalAgency-drivers.cmd>

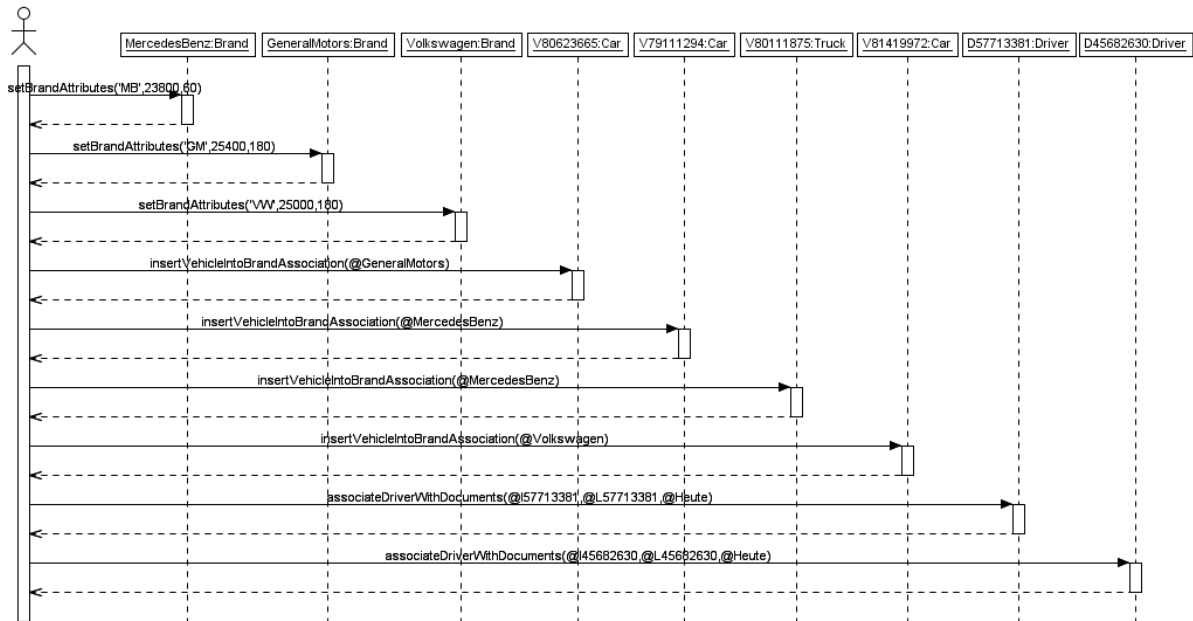
```

```
CarRentalAgency-start.cmd>
CarRentalAgency-start.cmd> -- set Cars on Station
CarRentalAgency-start.cmd>
CarRentalAgency-start.cmd> !insert (V80623665, BREC01) into VehicleOnStation
CarRentalAgency-start.cmd> !insert (V79111294, BREC01) into VehicleOnStation
CarRentalAgency-start.cmd> !insert (V80111875, BREC01) into VehicleOnStation
CarRentalAgency-start.cmd> !insert (V81419972, BREC01) into VehicleOnStation
CarRentalAgency-start.cmd>
```

## Objektdiagramm nach dem Start-Script



## Sequenz-Diagramm nach dem Start-Script



## **Testfall 1 (TC\_01.cmd)**

Ein normaler Ablauf zweier Anmietungen und Abgaben durch zwei unterschiedliche Fahrer, zwei unterschiedliche Fahrzeuge mit einem Check-In an einer anderen Station als die die voraussichtlich beim Check-Out eingetragen wurde.

### **Quelltext des Testfalls**

```
!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V79111294.vehicleStatusType := ←
    V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, ←
    D57713381, Heute, BRET01, 1223589600, #CreditCard)
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!create RA1050123457 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V80623665.vehicleStatusType := ←
    V80623665.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123457 checkOutRentalAgreement(1050123457, V80623665, ←
    D45682630, Heute, BREL01, 1223589600, #CreditCard)
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit

!openter RA1050123457 checkInRentalAgreement(OENC01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

## Kommandosequenzprotokoll

```
TC_01.cmd> !create RA1050123456 : RentalAgreement
TC_01.cmd>
TC_01.cmd> -- No periodical executions possible, so this needs to be done before the
TC_01.cmd> -- actual check Out of the vehicle starts.
TC_01.cmd>
TC_01.cmd> !set V79111294.vehicleStatusType := V79111294.automaticReadyForRentAfterFiveDays(Heute)
TC_01.cmd> !openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, D57713381, Heute, BRET01, 1223589600, #CreditCard)
precondition `driverFulfilAgeRequirement' is true
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is true
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is true
precondition `checkInDateNotLessThanCurrentDate' is true
TC_01.cmd>
TC_01.cmd> -- 1223589600 UnixDate for 2008-10-10
TC_01.cmd> read RentalAgreement_checkOutRentalAgreement.cmd
RentalAgreement_checkOutRentalAgreement.cmd> -- checkOutRentalAgreement (RANo : Integer, veh : Vehicle, drv : Driver, curDate : CurrentDate,
checkInStation : Station, checkInDate : Integer, payType : PaymentTypes)
RentalAgreement_checkOutRentalAgreement.cmd>
RentalAgreement_checkOutRentalAgreement.cmd> !set self.RANo := RANo
RentalAgreement_checkOutRentalAgreement.cmd> !set self.PaymentType := payType
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkOutDate := curDate.today
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkInDate := checkInDate
RentalAgreement_checkOutRentalAgreement.cmd> !insert (self, veh, drv, veh.currentStation, checkInStation) into OpenRentalAgreement
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleMovementType := #ORA
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleStatusType := #ON
RentalAgreement_checkOutRentalAgreement.cmd> !delete (veh, veh.currentStation) from VehicleOnStation
RentalAgreement_checkOutRentalAgreement.cmd>
TC_01.cmd> !opexit
TC_01.cmd>
TC_01.cmd> !create RA1050123457 : RentalAgreement
TC_01.cmd>
TC_01.cmd> -- No periodical executions possible, so this needs to be done before the
TC_01.cmd> -- actual check Out of the vehicle starts.
TC_01.cmd>
TC_01.cmd> !set V80623665.vehicleStatusType := V80623665.automaticReadyForRentAfterFiveDays(Heute)
TC_01.cmd> !openter RA1050123457 checkOutRentalAgreement(1050123457, V80623665, D45682630, Heute, BREL01, 1223589600, #CreditCard)
precondition `driverFulfilAgeRequirement' is true
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is true
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is true
precondition `checkInDateNotLessThanCurrentDate' is true
TC_01.cmd>
```

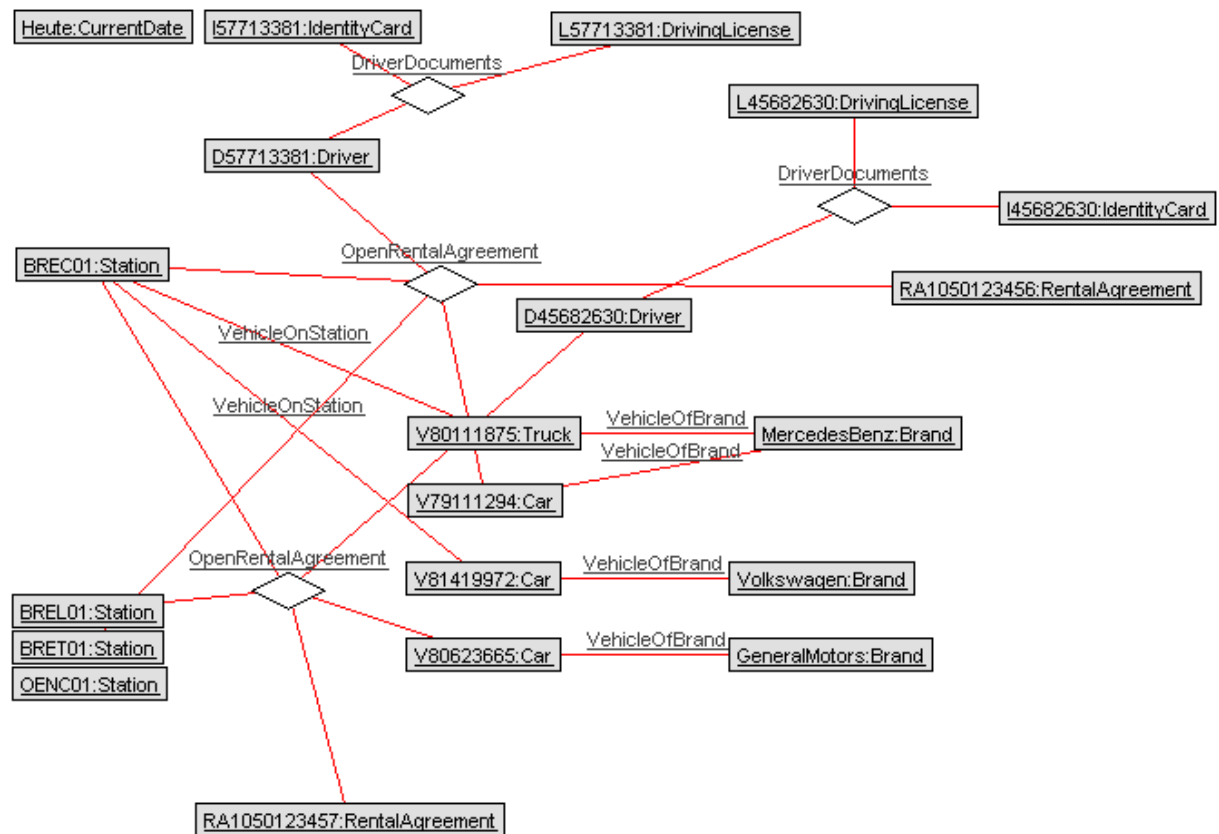
```

TC_01.cmd> -- 1223589600 UnixDate for 2008-10-10
TC_01.cmd> read RentalAgreement_checkOutRentalAgreement.cmd
RentalAgreement_checkOutRentalAgreement.cmd> -- checkOutRentalAgreement (RANo : Integer, veh : Vehicle, drv : Driver, curDate : CurrentDate,
checkInStation : Station, checkInDate : Integer, payType : PaymentTypes)
RentalAgreement_checkOutRentalAgreement.cmd>
RentalAgreement_checkOutRentalAgreement.cmd> !set self.RANo := RANo
RentalAgreement_checkOutRentalAgreement.cmd> !set self.PaymentType := payType
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkOutDate := curDate.today
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkInDate := checkInDate
RentalAgreement_checkOutRentalAgreement.cmd> !insert (self, veh, drv, veh.currentStation, checkInStation) into OpenRentalAgreement
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleMovementType := #ORA
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleStatusType := #ON
RentalAgreement_checkOutRentalAgreement.cmd> !delete (veh, veh.currentStation) from VehicleOnStation
RentalAgreement_checkOutRentalAgreement.cmd>
TC_01.cmd> !opexit
TC_01.cmd>
TC_01.cmd> !openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
precondition `vehicleCurrentMileageGreaterThanAtRentalBegin' is true
TC_01.cmd> read RentalAgreement_checkInRentalAgreement.cmd
RentalAgreement_checkInRentalAgreement.cmd> -- checkInRentalAgreement (checkInStation : Station, vehCurMileage : Integer, curDate : CurrentDate)
RentalAgreement_checkInRentalAgreement.cmd>
RentalAgreement_checkInRentalAgreement.cmd> !set self.distanceTravelled := vehCurMileage - self.vehicleontheroad->asSequence->at(1).mileage
RentalAgreement_checkInRentalAgreement.cmd> !set self.checkInDate := curDate.today
RentalAgreement_checkInRentalAgreement.cmd> !openter self.vehicleontheroad->asSequence->at(1) checkInVehicle(vehCurMileage, checkInStation,
curDate)
RentalAgreement_checkInRentalAgreement.cmd> read Vehicle_checkInVehicle.cmd
Vehicle_checkInVehicle.cmd> -- checkInVehicle(newMileage : Integer, checkInStation : Station, curDate : CurrentDate)
Vehicle_checkInVehicle.cmd>
Vehicle_checkInVehicle.cmd> !set self.mileage := newMileage
Vehicle_checkInVehicle.cmd> !set self.vehicleMovementType := #NOM
Vehicle_checkInVehicle.cmd> !set self.vehicleStatusType := #TC
Vehicle_checkInVehicle.cmd> !set self.vehicleLastMovementEnd := curDate.today
Vehicle_checkInVehicle.cmd> !set self.saleFlag := self.checkSaleVehicle(curDate)
Vehicle_checkInVehicle.cmd> !insert (self, checkInStation) into VehicleOnStation
Vehicle_checkInVehicle.cmd>
RentalAgreement_checkInRentalAgreement.cmd> !opexit
RentalAgreement_checkInRentalAgreement.cmd> !delete (self, self.vehicleontheroad->asSequence->at(1), self.driver->asSequence->at(1),
self.checkOutStation->asSequence->at(1), self.checkInStation->asSequence->at(1)) from OpenRentalAgreement
RentalAgreement_checkInRentalAgreement.cmd>
TC_01.cmd> !opexit
TC_01.cmd>
TC_01.cmd> !openter RA1050123457 checkInRentalAgreement(OENC01, 23700, Heute)
precondition `vehicleCurrentMileageGreaterThanAtRentalBegin' is true
TC_01.cmd> read RentalAgreement_checkInRentalAgreement.cmd
RentalAgreement_checkInRentalAgreement.cmd> -- checkInRentalAgreement (checkInStation : Station, vehCurMileage : Integer, curDate : CurrentDate)
RentalAgreement_checkInRentalAgreement.cmd>
RentalAgreement_checkInRentalAgreement.cmd> !set self.distanceTravelled := vehCurMileage - self.vehicleontheroad->asSequence->at(1).mileage
RentalAgreement_checkInRentalAgreement.cmd> !set self.checkInDate := curDate.today
RentalAgreement_checkInRentalAgreement.cmd> !openter self.vehicleontheroad->asSequence->at(1) checkInVehicle(vehCurMileage, checkInStation,
curDate)

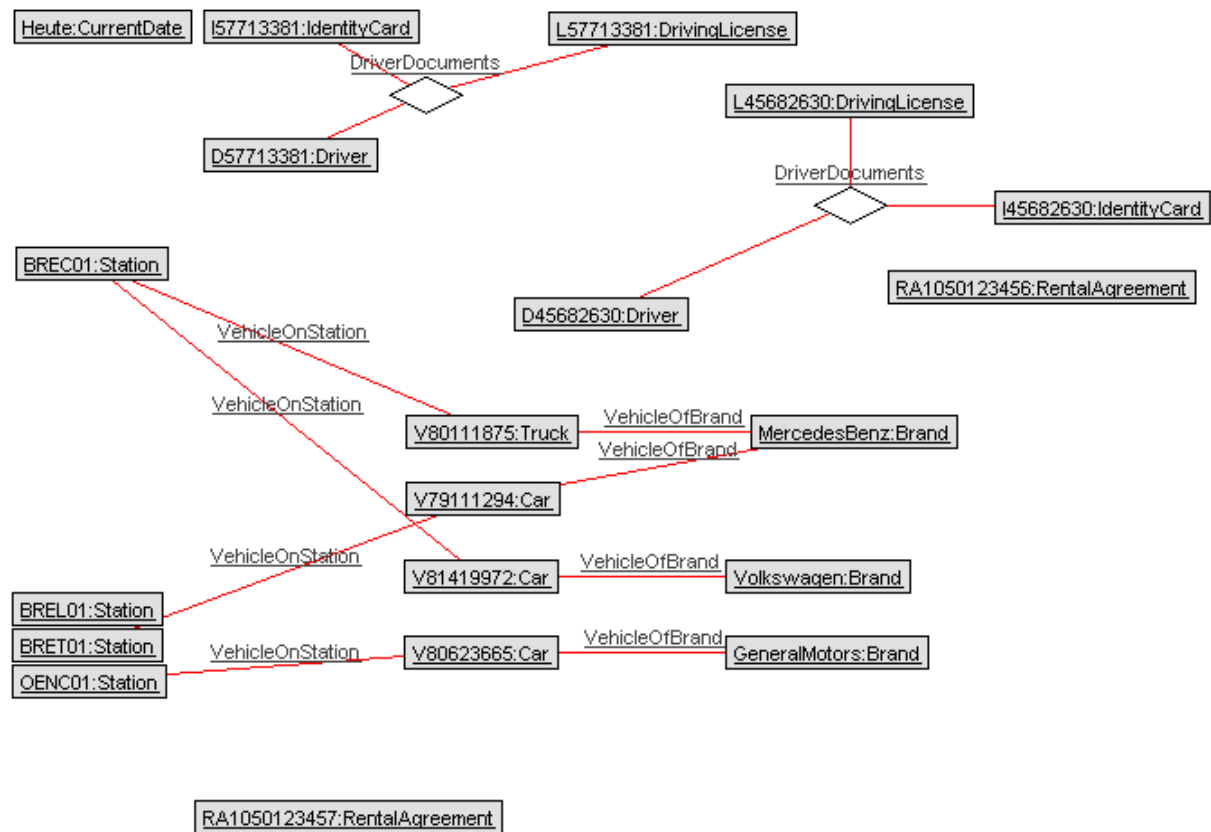
```

```
RentalAgreement_checkInRentalAgreement.cmd> read Vehicle_checkInVehicle.cmd
Vehicle_checkInVehicle.cmd> -- checkInVehicle(newMileage : Integer, checkInStation : Station, curDate : CurrentDate)
Vehicle_checkInVehicle.cmd>
Vehicle_checkInVehicle.cmd> !set self.mileage := newMileage
Vehicle_checkInVehicle.cmd> !set self.vehicleMovementType := #NOM
Vehicle_checkInVehicle.cmd> !set self.vehicleStatusType := #TC
Vehicle_checkInVehicle.cmd> !set self.vehicleLastMovementEnd := curDate.today
Vehicle_checkInVehicle.cmd> !set self.saleFlag := self.checkSaleVehicle(curDate)
Vehicle_checkInVehicle.cmd> !insert (self, checkInStation) into VehicleOnStation
Vehicle_checkInVehicle.cmd>
RentalAgreement_checkInRentalAgreement.cmd> !opexit
RentalAgreement_checkInRentalAgreement.cmd> !delete (self, self.vehicleontheroad->asSequence->at(1), self.driver->asSequence->at(1),
self.checkOutStation->asSequence->at(1), self.checkInStation->asSequence->at(1)) from OpenRentalAgreement
RentalAgreement_checkInRentalAgreement.cmd>
TC_01.cmd> !opexit
TC_01.cmd>
TC_01.cmd>
```

## Objekt Diagramm Testfall 1 nach dem Check-Out



## Objekt Diagramm Testfall 1 nach dem Check-In



## Testfall 2 (TC\_02.cmd)

In diesem Testfall wird die Gültigkeit des Ausweises eines Fahrers der im Folgenden für einen CheckOut verwendet werden soll um ein Jahr reduziert, und ist somit zum gegenwärtigen Zeitpunkt ungültig. Daher schlägt eine Pre-Condition für den Check-Out fehl.

### Quelltext des Testfalls

```
-- TEST CASE: CHANGE DRIVERS IDENTITY CARD VALIDITY INTO INVALID DATE
!set I57713381.validTill := 1191189600

!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294,
D57713381, Heute, BRET01, 1223589600, #CreditCard)

-- 1223589600 UnixDate for 2008-10-10
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

### Kommandosequenzprotokoll

```
use> read TC_02.cmd
TC_02.cmd> -- TEST CASE: CHANGE DRIVERS IDENTITY CARD VALIDITY INTO INVALID DATE
TC_02.cmd> !set I57713381.validTill := 1191189600
TC_02.cmd>
TC_02.cmd> !create RA1050123456 : RentalAgreement
TC_02.cmd>
TC_02.cmd> -- No periodical executions possible, so this needs to be done before the
TC_02.cmd> -- actual check Out of the vehicle starts.
TC_02.cmd>
TC_02.cmd> !set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
TC_02.cmd> !openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, D57713381,
Heute, BRET01, 1223589600, #CreditCard)
precondition `driverFulfillsAgeRequirement' is true
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is true
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is false
precondition `checkInDateNotLessThanCurrentDate' is true
```

## Testfall 3 (TC\_03.cmd)

In diesem Testfall wird der SaleFlag-Status des auszucheckenden Fahrzeuges auf #BlockedForSale gesetzt. Daher schlägt eine Pre-Condition für den Check-Out fehl.

### Quelltext des Testfalls

```
-- TEST CASE: CHANGE VEHICLE SALE FLAG TO BE BLOCKED FOR SALE
!set V79111294.saleFlag := #BlockedForSale

!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294,
D57713381, Heute, BRET01, 1223589600, #CreditCard)

-- 1223589600 UnixDate for 2008-10-10
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

### Kommandosequenzprotokoll

```
use> read TC_03.cmd
TC_03.cmd> -- TEST CASE: CHANGE VEHICLE SALE FLAG TO BE BLOCKED FOR SALE
TC_03.cmd> !set V79111294.saleFlag := #BlockedForSale
TC_03.cmd>
TC_03.cmd> !create RA1050123456 : RentalAgreement
TC_03.cmd>
TC_03.cmd> -- No periodical executions possible, so this needs to be done before the
TC_03.cmd> -- actual check Out of the vehicle starts.
TC_03.cmd>
TC_03.cmd> !set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
TC_03.cmd> !openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, D57713381,
Heute, BRET01, 1223589600, #CreditCard)
precondition `driverFulfilAgeRequirement' is true
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is false
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is true
precondition `checkInDateNotLessThanCurrentDate' is true
```

## Testfall 4 (TC\_04.cmd)

In diesem Testfall wird versucht, einen Check-In mit einem Kilometer-Stand versucht, der niedriger als der Check-Out-Kilometerstand ist. Daher schlägt eine Pre-Condition für den Check-In fehl.

### Quelltext des Testfalls

```
-- TEST CASE: CHECK IN WITH LESSER MILEAGE THAN CHECK OUT
!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294,
D57713381, Heute, BRET01, 1223589600, #CreditCard)

-- 1223589600 UnixDate for 2008-10-10
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!openter RA1050123456 checkInRentalAgreement(BRET01, 10, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

### Kommandosequenzprotokoll

```
use> read TC_04.cmd
TC_04.cmd> -- TEST CASE: CHECK IN WITH LESSER MILEAGE THAN CHECK OUT
TC_04.cmd> !create RA1050123456 : RentalAgreement
TC_04.cmd>
TC_04.cmd> -- No periodical executions possible, so this needs to be done before the
TC_04.cmd> -- actual check Out of the vehicle starts.
TC_04.cmd>
TC_04.cmd> !set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
TC_04.cmd> !openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, D57713381,
Heute, BRET01, 1223589600, #CreditCard)
precondition `driverFulfillsAgeRequirement' is true
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is true
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is true
precondition `checkInDateNotLessThanCurrentDate' is true
TC_04.cmd>
TC_04.cmd> -- 1223589600 UnixDate for 2008-10-10
TC_04.cmd> read RentalAgreement_checkOutRentalAgreement.cmd
RentalAgreement_checkOutRentalAgreement.cmd> -- checkOutRentalAgreement (RANo : Integer, veh :
Vehicle, drv : Driver, curDate : CurrentDate, checkInStation : Station, checkInDate : Integer,
payType :
PaymentTypes)
RentalAgreement_checkOutRentalAgreement.cmd>
RentalAgreement_checkOutRentalAgreement.cmd> !set self.RANo := RANo
RentalAgreement_checkOutRentalAgreement.cmd> !set self.PaymentType := payType
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkOutDate := curDate.today
RentalAgreement_checkOutRentalAgreement.cmd> !set self.checkInDate := checkInDate
RentalAgreement_checkOutRentalAgreement.cmd> !insert (self, veh, drv, veh.currentStation,
checkInStation) into OpenRentalAgreement
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleMovementType := #ORA
RentalAgreement_checkOutRentalAgreement.cmd> !set veh.vehicleStatusType := #ON
RentalAgreement_checkOutRentalAgreement.cmd> !delete (veh, veh.currentStation) from
VehicleOnStation
RentalAgreement_checkOutRentalAgreement.cmd>
TC_04.cmd> !opexit
TC_04.cmd>
TC_04.cmd> !openter RA1050123456 checkInRentalAgreement(BRET01, 10, Heute)
precondition `vehicleCurrentMileageGreaterThanOrEqualToAtRentalBegin' is false
```

## Testfall 5 (TC\_05.cmd)

In diesem Testfall wird das SaleFlag eines Fahrzeugs auf #BlockedForSale gesetzt, nachdem das Fahrzeug ausgecheckt wurde. Dies passiert in der „realen Welt“ in dem Moment, wo das Fahrzeug während einer Miete die Nutzungsdauerbeschränkung oder eine geschätzte Kilometerbegrenzung erreicht hat. Das Fahrzeug muss dann getauscht werden. Daher fällt die Invariante in dem Moment auf false um auf dieses Problem hinzuweisen.

### Quelltext des Testfalls

```
-- TEST CASE: VEHICLE GETS BLOCKED FOR SALE WHILE IN RENTAL AGREEMENT
!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check Out of the vehicle starts.

!set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294,
D57713381, Heute, BRET01, 1223589600, #CreditCard)

-- 1223589600 UnixDate for 2008-10-10
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!set V79111294.saleFlag := #BlockedForSale

check

!openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

### Kommandosequenzprotokoll

Zur Verkürzung des Protokolls ist anzunehmen, dass sowohl der Check-Out als auch der Check-In erfolgreich sind. Der Check-Out weil das Fahrzeug zu dem Zeitpunkt noch nicht BlockedForSale ist und der Check-In da das eingeecheckte Fahrzeug trotz seines BlockForSale Flags ja trotzdem in die Station eingeecheckt werden können muss.

```
TC_05.cmd> check
checking structure...
checking invariants...
checking invariant (1) `CurrentDate::CurrentDateGreaterThanOrEqualToZero': OK.
checking invariant (2) `Driver::driverIDUnique': OK.
checking invariant (3) `DrivingLicense::numberOfDrivingLicenseUnique': OK.
checking invariant (4) `IdentityCard::numberOfIdentityCardUnique': OK.
checking invariant (5) `RentalAgreement::numberOfRentalAgreementUnique': OK.
checking invariant (6) `Station::stationShortNameUnique': OK.
checking invariant (7) `Vehicle::mileageNotNegative': OK.
checking invariant (8) `Vehicle::vehicleBlockedForSaleImpliesNoOpenRentalAgreement': FAILED.
  -> false : Boolean
checking invariant (9) `Vehicle::vehicleOnTheRoadImpliesNotOnAStation': OK.
checking invariant (10) `Vehicle::vehicleToBeCheckedImpliesNoOpenRentalAgreement': OK.
checking invariant (11) `Vehicle::vehicleUnitNumberUnique': OK.
checked 11 invariants in 0.031s, 1 failure.
TC_05.cmd>
```

## Testfall 6 (TC\_06.cmd)

In diesem Testfall wird das Mindestalter für das Fahrzeug so hoch gesetzt, dass der gewählte Fahrer das Alter nicht erreicht. Daher schlägt eine Pre-Condition für den Check-Out fehl.

### Quelltext des Testfalls

```
-- TEST CASE: CHANGE VEHICLE MIN AGE FOR RENTAL TO BE HIGHER THAN DRIVER'S AGE
!set V79111294.minAgeForRental := 27

!create RA1050123456 : RentalAgreement

-- No periodical executions possible, so this needs to be done before the
-- actual check out of the vehicle starts.

!set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
!openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294,
D57713381, Heute, BRET01, 1223589600, #CreditCard)

-- 1223589600 UnixDate for 2008-10-10
read RentalAgreement_checkOutRentalAgreement.cmd
!opexit

!openter RA1050123456 checkInRentalAgreement(BRET01, 23700, Heute)
read RentalAgreement_checkInRentalAgreement.cmd
!opexit
```

### Kommandosequenzprotokoll

```
use> read TC_06.cmd
TC_06.cmd> -- TEST CASE: CHANGE VEHICLE MIN AGE FOR RENTAL TO BE HIGHER THAN DRIVER'S AGE
TC_06.cmd> !set V79111294.minAgeForRental := 27
TC_06.cmd>
TC_06.cmd> !create RA1050123456 : RentalAgreement
TC_06.cmd>
TC_06.cmd> -- No periodical executions possible, so this needs to be done before the
TC_06.cmd> -- actual check out of the vehicle starts.
TC_06.cmd>
TC_06.cmd> !set V79111294.vehicleStatusType :=
V79111294.automaticReadyForRentAfterFiveDays(Heute)
TC_06.cmd> !openter RA1050123456 checkOutRentalAgreement(1050123456, V79111294, D57713381,
Heute, BRET01, 1223589600, #CreditCard)
precondition `driverFulfilfsAgeRequirement' is false
precondition `vehicleIsNotOnTheRoad' is true
precondition `vehicleNotBlockedForSale' is true
precondition `vehicleMustBeReadyForRent' is true
precondition `driversDrivingLicenseStillValid' is true
precondition `driversIdCardStillValid' is true
precondition `checkInDateNotLessThanCurrentDate' is true
```