

Exercise

OCL Expressions



Prof. Dr. Bernhard Rumpe
Lehrstuhl für Software Engineering
RWTH Aachen

<http://www.se-rwth.de/>

Exercise 4.1 – Solution 1

1. After the return of a vehicle, the car is available, inService or damaged and the rental contract is deleted.

```
context Administration.return(RentalContract c):  
  pre: rentalContract.contains(c)  
  post: !rentalContract.contains(c) &&  
        (forall v in c.rentedCars:  
          v.state == VehicleState.available  
            || VehicleState.inService  
            || VehicleState.damaged)
```

Exercise 4.1 – Solution 2

2. When a new rental contract is created by the administration, the customer has to exist in the car rental system and the desired vehicle has to be available. Customer and vehicles must be specified in the rental contract.

```
context RentalContract Administration.rent(Customer leaser,  
                                           Vehicle[] vehicles):  
  pre: (forall v in vehicles:  
        Administration.getAvailableVehicles().contains(v))  
        &&  
        (exists c in Administration.customers: c == leaser)  
  post: (result.getLeaser() == leaser)  
        && (result.rentedCars == vehicles)
```

Exercise 4.1 – Solution 3

3. The customer ID is unique.

Variant 1:

```
context Administration inv:  
  forall c1 in Administration.customers:  
    !exists c2 in Administration.customers:  
      (c1 != c2) && (c1.customerID == c2.customerID)
```

Variant 2:

```
context Administration inv:  
  forall c1 in customers:  
    {c2 in Customer | c1.customerID == c2.customerID}.size == 1
```

Variant 3:

```
inv:  
  forall c1 in Customer:  
    {c2 in Customer | c1.customerID == c2.customerID}.size == 1
```

Exercise 4.1 – Solution 4

4. If a vehicle is inService, it is not in the list of the available vehicles.

```
context Administration inv:  
  forall v in getVehicles():  
    v.state == VehicleState.inService implies  
      !getAvailableVehicles().contains(v)
```

Exercise 4.2 – Code Generation from OCL Expressions

- Generate Java code for the expression 2 from task 4.1.
2. When a new rental contract is created by the administration, the customer has to exist in the car rental system and the desired vehicle has to be available. Customer and vehicles must be specified in the rental contract.

```
context RentalContract Administration.rent(Customer leaser,
                                           Vehicle[] vehicles):
pre: (forall v in vehicles:
      Administration.getAvailableVehicles().contains(v))
      &&
      (exists c in Administration.customers: c == leaser)
post: (result.getLeaser() == leaser)
      && (result.rentedCars == vehicles)
```

Exercise 4.2 – Solution (1)

JAVA

```
public RentalContract rent(Customer leaser,  
                           List<Vehicle> vehicles) {  
    // stores the return-value of the method-body  
    RentalContract result;  
  
    // check of the pre-condition  
    boolean preCondition1 = true, preCondition2 = false;  
    for (Vehicle v: vehicles){  
        if ( !getAvailableVehicles().contains(v) ){  
            preCondition1 = false;  
            break;  
        }  
    }  
    for (Customer c:customers) {  
        if (c == leaser) {  
            preCondition2 = true;  
            break;  
        }  
    }  
    assert(preCondition1 && preCondition2);  
  
    ...  
}
```

Exercise 4.2 – Solution (2)

JAVA

```
...  
  
// method-body, return-value is stored in result  
result = ...  
  
// check of the post-condition  
assert((result.getLeaser()==leaser)  
        && (result.getRentedCars()==vehicles));  
  
return result;  
}
```