Software Modeling and Verification
Lehrstuhl für Informatik 2
RWTH Aachen University

Prof. Dr. Ir. J.-P. Katoen
C. Kern

Foundations of the UML

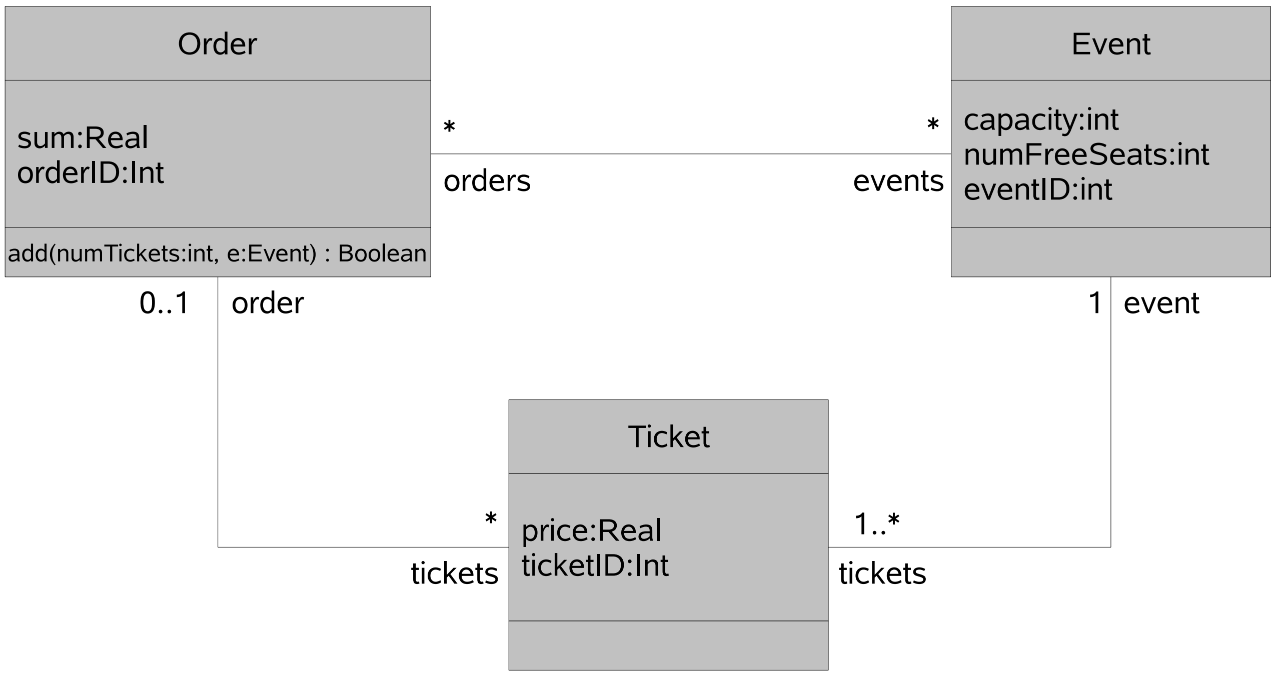
Winter Term 07/08

– Assignment 7 –

Hand in until February 6th before the exercise class.

Exercise 1 (10 points)

Let the following class diagram be given:



It is part of a ticket reservation system where a user can order tickets for events like cinema films or theater or cabaret performances.

a) Describe the following statements as OCL invariants:

   i) The attribute `sum` must not be negative.
   ii) The attribute `sum` is 0 if no tickets are ordered.
   iii) The attribute `sum` exactly describes the prize of the tickets ordered.
   iv) Different instances of `Ticket` have different order numbers.
   v) Each `Event` has at least 0 free seats.
   vi) An `Event`'s capacity (i.e., the total number of seats) is greater than the number of all ordered tickets of this event.

b) Describe the pre- and post conditions of method `add`, which adds a number of `num` tickets to the order if there are still enough free seats and in this case increases the sum correctly. In case there are not enough seats nothing is done.

Exercise 2 **(10 points)**

Given the class diagram from exercise 1, a concrete instance of the system is being created. There are two events: a film *Dumbo* and a theater performance called *The Magic Flute*. The film has got two categories: two cheap tickets for 5 Euro, each and two more expensive ones for 10 Euros, each. The theater seats are divided into 5 categories. There is a ticket for 20, 40, 60, 80 and 100 Euros. The film has got a capacity of 4 but there are already two seats occupied by an order O_0 which includes a ticket for 5 and one for 10 Euros. Moreover O_0 contains the most expensive ticket for the theater event. From now on cheaper tickets are sold first (e.g., the first ticket to be sold for the film *Dumbo* is a 5 Euro ticket).

a) Draw the system's "initial" configuration which is described above.

b) Draw the system's configuration after each of the following actions:

   (a) A new `order` O_1 is created. For this order:

      i) Buy a ticket for *The Magic Flute*.

      ii) Buy a ticket for the film *Dumbo*.

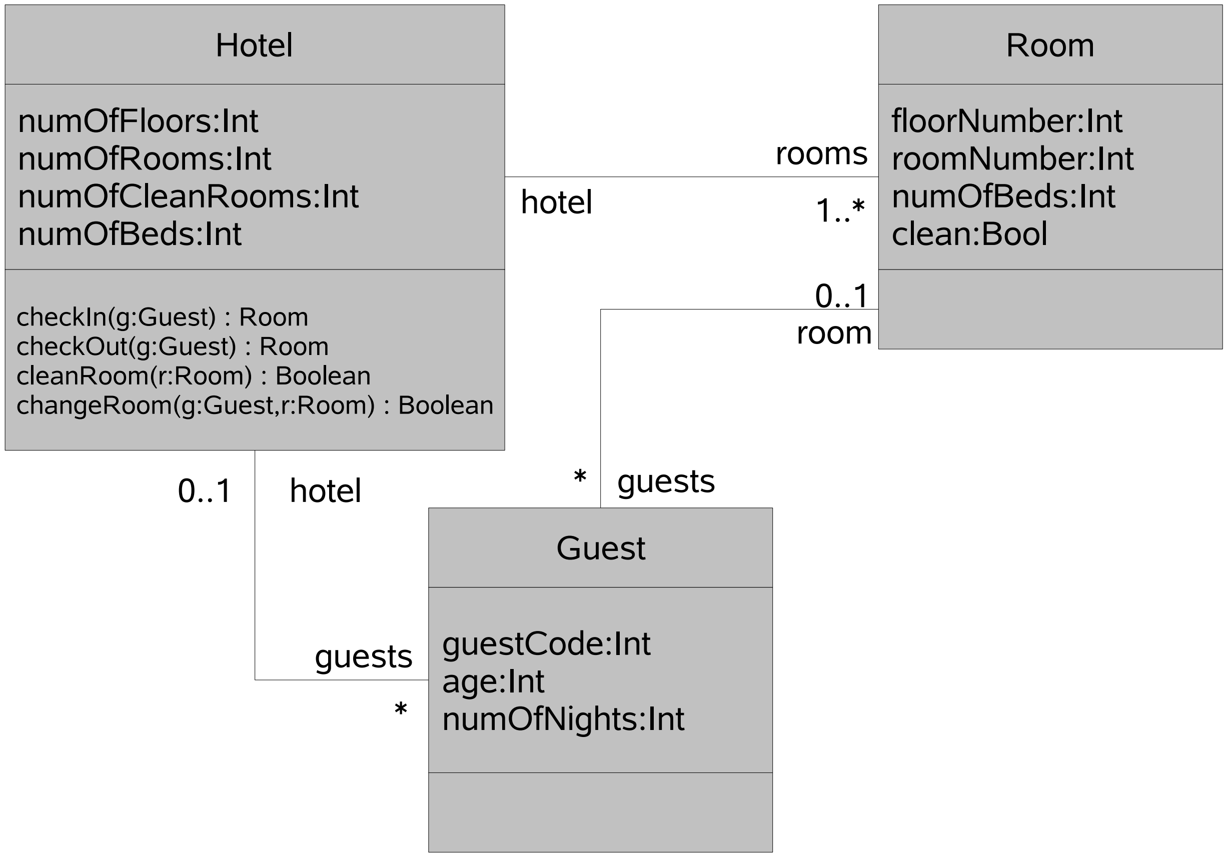
   (b) A new `order` O_2 is created. For this order:

      iii) Buy two tickets for the film *Dumbo*.

      iv) Buy three tickets for *The Magic Flute*.

Exercise 3 **(20 points)**

Given the following, slightly extended, version of the class diagram from the lecture:



Determine the formal semantics of the following OCL statements (i.e., translate the OCL statements into BOTL according to the rules from the lecture).

a) <u>invariant:</u>
```
context Hotel
inv:  guests→size ≤ numOfBeds
```

b) <u>invariant:</u>
```
context Hotel
inv:  numOfBeds = rooms→iterate(r ; x=0 | x + r.numOfBeds)
```

c) <u>pre and post condition:</u>
```
context Hotel::cleanRoom(r:Room)
pre:   (not r.clean = tt) and r.guests→size = 0
post:  r.clean = tt and
       numOfCleanRooms = numOfCleanRooms@pre + 1
```

d) <u>pre and post condition:</u>
```
context Hotel::changeRoom(g:Guest, r:Room)
pre:   guests→contains(g) and r.guests→size < r.numOfBeds
post:  g.room.guests→size = g.room.guests@pre→size - 1 and
       r.guests→size = r.guests@pre→size + 1 and
       g.room = r
```