Prof. Dr. Peter Thiemann
Manuel Geffken
Matthias Keil

Summer Term 2012

Softwaretechnik

http://proglang.informatik.uni-freiburg.de/teaching/swt/2012/

Exercise Sheet 10

Exercise 1

Let *col* be a collection in OCL. Define OCL expressions that implement the following operations:

1. `hasNElements`: Returns `true` for some number n and some expression *expr*, if there exist exactly n elements in *col* that fulfill *expr*. The iteration variable in *expr* is `it`.
2. `isUnique`: Returns `true` if *col* does not contain duplicates. Do *not* use the builtin function of the same name.
3. `take`: Returns for some number n a subset of *col*. The size of the subset is the minimum of n and the size of *col*.

Exercise 2

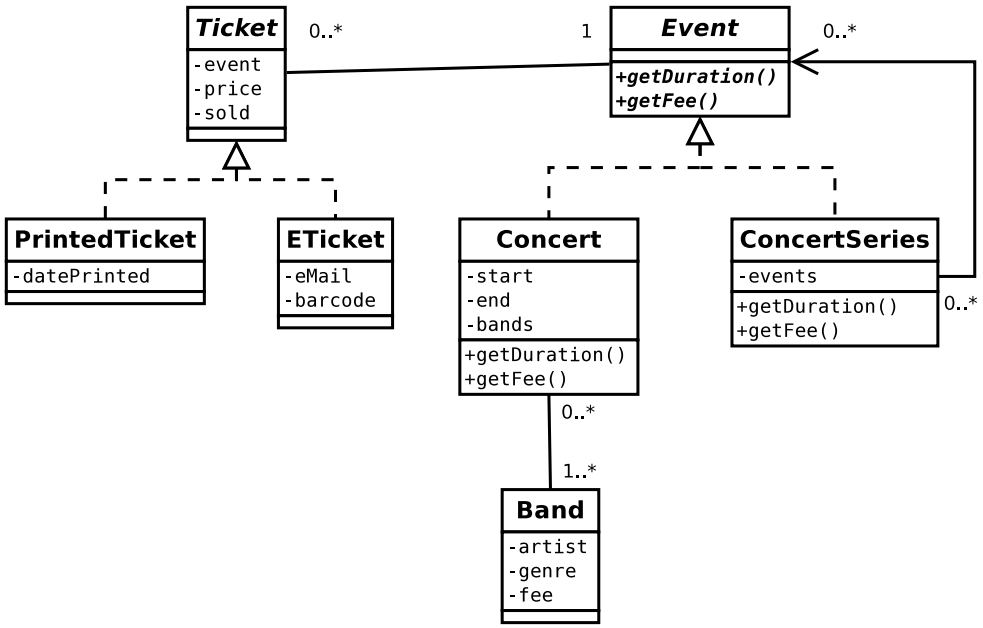
1. In the lecture, you have seen a precondition for the operation `move` of class `Meeting`. Refine this precondition so that meetings in different locations can take place at the same time. Keep in mind that each team member can only be in one location at one time.
2. The class `Meeting` from the lecture gets now extended by an operation

   ```
   relocate(newLocation : Location)
   ```

   which changes the location of a meeting. Find sensible pre- and postconditions for `relocate`.

Exercise 3

The following class diagram models parts of the Event Management System:



Implement the following constraints in OCL:

1. If an *Event* is a *Concert*, the duration is 180 minutes.
2. A *Concert* consists of 2 *Bands* (e.g. a supporting act, and a headliner) and the fee for each *Band* does not exceed 10,000 EUR.
3. Assume *PrintedTickets* are not issued anymore. Then each *Ticket* associated with an *Event* has to be kind of *ETicket*.
4. Assume all *Tickets* of an *Event* are available from the beginning, but will be sold later on. Then the fee of an *Event* must be amortized by the sum of all *Ticket* prices.