

# VU Programming Languages

## Task 3

May 24, 2013

### 1 Specification

In a strongly typed functional language (ML or Haskell), develop an application for managing train seat reservations. Every reservation is made for a specific train between two named stations for some count of persons. If it is necessary to switch trains on a journey, separate reservations must be made.

The amount of available sets is not only determined by the number of available seats per train car and the number of train cars per train; additionally, each train has a certain number of seats which must remain free for passengers without reservations. Of course, no seat may be booked twice.

If a reservation is made for more than one person, each seat must be in the same train car and have consecutive seat numbers (i.e. Car 3, Seats 5 to 9).

Besides placing reservations, the following queries must be supported:

- The train network with all known stations, trains (including count and size of train cars) and stations where switching trains is possible. The train network should include several crossing lines as well as multiple trains per line.
- The minimum number of seats which must remain free as well as the maximum number of reservations per train and train car between two stations. Consider that reservations might not be made for the entire line.
- All reservations for a seat in a specific train. The result must include all station pairs for existing reservations.
- The smallest number of seats available for reservation between two stations, as well as the maximum group count (= maximum number of people per reservation) for one or more trains (if it is necessary to change trains).
- All data should be persisted between program executions.

This task is about managing data. At first glance, this seems to be a contradiction to modern functional languages. However, there are many different ways of solving this exercise. Taking a closer look at the language libraries might help finding an optimal solution.

## **2 ML**

For programming in ML, OCaml is recommended. This system extends ML by object-oriented concepts; however, limit yourself to the functional aspects of the language and do not use any object-oriented extensions.

## **3 Haskell**

General information about Haskell is available at <http://www.haskell.org/>. The recommended system is GHC: <http://www.haskell.org/ghc/>.