

10.2.1 Time tables, Routes, and Flights

Description

Time tables, routes and flights are obvious represented by objects. A timetable has an entry for each of the different routes, and in the model each entry is represented by an object of class `Route`:

```
timeTable: obj
  entries: obj OrderedList(Route)
```

Each `Route` object has attributes that represent the name of the route, the source and destination airports, the scheduled departure and arrival time, and scheduled flying time. The flights of a route is represented by a list of `Flight` objects for each `Route` object.

```
class Route(name, origin, destination: var String):
  scheduledDepartureTime: var TimeOfDay
  scheduledArrivalTime: var TimeOfDay
  setTime(dt: var TimeOfDay(0 hours),
          at: var TimeOfDay(0 hours)):|
    scheduledDepartureTime := dt
    scheduledArrivalTime := at
  scheduledFlyingTime -> sft: var
  Time.Hours:
    sft := scheduledArrivalTime - scheduledDepartureTime
  flights: obj OrderedList(Flight)
```

The `scheduledFlyingTime` is an important information when booking. `scheduledDepartureTime` and `scheduledArrivalTime` are also used for showing flight status, at airports or at the flight status website of the airline company. The reason that `scheduledDepartureTime` and `scheduledArrivalTime` are not modelled as parameters of `Route` is that these times may be changed at different setups of the time table.

Setting up the routes of the time table is done by a sequence of actions that generate `Route` objects, set the values of their attributes and insert them into the `timeTable`:

```
SK1926: obj Route("SK1926", "AAR", "OSL")
SK1926.setTimes(TimeOfDay(12.30 hours), TimeOfDay(13.50 hours))

timeTable.entries.insert(SK1926)
...
SK1927: obj Route("SK1927", "OSL", "AAR")
SK1927.setTimes(TimeOfDay(14.20 hours), TimeOfDay(15.40 hours))|

timeTable.entries.insert(SK1927)
...
```

As mentioned above, flights are represented by objects. As the notion of flight is defined in the context of route, the class `Flight` is defined in the context of the class `Route`.

As mentioned above we shall use this model of routes and flights for both booking of flights and for showing status of flights. In the next section we look at what is required for booking, and then for the status of flights.