Functional Programming
Project: Doodle (part 1)

Assistant: Noah Van Es
noah.van.es@vub.be

The final grade of the course is determined by an oral exam and a project (each counts for 50% of the total grade for the course). The project consists out of two parts. A non-empty solution for both parts has to be submitted in order to get a grade. This document describes the first part of the project.

Project description

The goal of this first programming assignment is to create a terminal-based application for scheduling meetings (similar to the popular Doodle[1] meeting application).

A "doodle" is used to agree on a timeslot for an event with many participants. Specifically, the organiser of the event can create a doodle, providing all possible timeslots for an event. Upon creating such a doodle, the organiser receives an identifier which can be shared with other participants. These participants can then use this identifier to retrieve the doodle and indicate for which timeslots they are available. Once all participants have filled in the doodle, the idea is that the organiser can select the best timeslot for the event (usually, the one that satisfies all or most participants).

Practical information

Submission The deadline for this assignment is **May 16, 2022 (23:59)**. You submit all your code in single ZIP-file through the Canvas platform.

Grading Note that your project will not only be graded on functionality, but also on the efficiency and quality of your implementation (programming style, performance considerations, ...).

[1] `https://doodle.com`

Individual work In case you have questions or need clarification to complete this project, feel free to contact the assistent by mail (`noah.van.es@vub.be`) or make an appointment to schedule a meeting.
The project has to be executed on a strictly individual basis! Automated tools are used to scan for plagiarism between projects; any suspicion of code exchange shall immediately be reported to the faculty's dean, and both parties shall be sanctioned in accordance with the examination rules regarding plagiarism.

Project assignment

Your code should support the functionalities described below.

Doodles It should be possible to:

- Create a new doodle with a given title.
- Display a doodle as an ASCII-based table containing its title and timeslots. Timeslots should be displayed by order of starting time.
- Add a timeslot to a doodle with a given start and end time.
- Remove a timeslot from a doodle (based on its position).
- Add/remove ("toggle") a name for a given timeslot (based on its position).

Timeslots should be precise to the minute and work everywhere on earth. **It is not allowed for a doodle to contain overlapping timeslots.**

To guide you in the right direction of a well-designed solution, you are required to implement the following `Doodle` type class:

```
class Doodle d where
  -- create an empty doodle with a given title
  initialize :: String -> d t
  -- add a timeslot to the doodle
  add :: (t,t) -> d t -> d t
  -- remove a timeslot from the doodle
  remove :: Int -> d t -> d t
  -- toggle a name to a timeslot
  toggle :: String -> Int -> d t -> d t
```

Pools We will use a pool to store all doodles, associating each doodle with a key. It should therefore be possible to associate a new doodle in the pool using a fresh (i.e., unused) key, allowing later access to the doodle using this key. More specifically, you should implement the following `Pool` type class.

```haskell
class Pool p where
  -- return an unused key in the pool
  freshKey :: (Ord k, Enum k) => p k (d t) -> k
  -- associate a key with a doodle
  set :: Ord k => k -> d t -> p k (d t) -> p k (d t)
  -- lookup the doodle for a given key
  get :: Ord k => k -> p k (d t) -> Maybe (d t)
```

For this programming assignment, you are allowed to use a Haskell file[2] that contains code to help you interact with the user. It also contains the aforementioned `Doodle` and `Pool` type classes. To help you creating the `main` function of your application, the file exports a function called `run`. It should be called with an initial (empty) pool of doodles, for instance: `main = run emptyDoodle`.

Below, we illustrate its intended usage with an example.

```
Create a new doodle or participate to an existing one?
> Left "Christmas party!"
Add/Remove a slot?
> Just (Left ("2022-12-25T12:00+01:00", "2022-12-25T17:00+01:00"))
+-------------------------------------------------+
| Christmas party!                                |
+-------------------------+-----------------------+
| 2022-12-25T12:00+01:00 | 2022-12-25T17:00+01:00 |
+-------------------------+-----------------------+
Add/Remove a slot?
> Just (Left ("2022-12-24T18:00+01:00", "2022-12-25T00:00+01:00"))
+-------------------------------------------------+
| Christmas party!                                |
+-------------------------+-----------------------+
| 2022-12-24T18:00+01:00 | 2022-12-25T23:00+01:00 |
+-------------------------+-----------------------+
| 2022-12-25T12:00+01:00 | 2022-12-25T17:00+01:00 |
+-------------------------+-----------------------+
Add/Remove a slot?
> Nothing
Doolde ID: 1
Create a new doodle or participate to an existing one?
> Right 1
Identify yourself
> "Alice"
+-------------------------------------------------+
| Christmas party!                                |
+-------------------------+-----------------------+
| 2022-12-24T18:00+01:00 | 2022-12-25T23:00+01:00 |
+-------------------------+-----------------------+
| 2022-12-25T12:00+01:00 | 2022-12-25T17:00+01:00 |
+-------------------------+-----------------------+
Toggle?
> Just 0
+-----------------------------------------------------------+
| Christmas party!                                          |
```

[2]Made available on Canvas.

```
+-------------------------+-----------------------+---------+
| 2022-12-24T18:00+01:00 | 2022-12-25T23:00+01:00 | Alice |
+-------------------------+-------------------------+-------+
| 2022-12-25T12:00+01:00 | 2022-12-25T17:00+01:00 |       |
+-------------------------+-------------------------+-------+
Toggle?
> Nothing
Thanks for participating!
Create a new doodle or participate to an existing one?
^C
```