

Advanced Programming (I00032) 2021

Basic iTasks

Assignment 9

Goal

The goal of this exercise is to get you started with the iTask system. After making this exercise you should know how to use basic editors like `enterInformation`, `updateInformation`, and `viewInformation` for new datatypes.

Version

The iTask system is under development. To prevent version problems we will use a single system throughout the course (unless big problems pop up). This is the same version as announced previous week. Do not use the daily builds for the work you hand in. They may contain experimental features, changes in the API, as well as bugs.

In order to check whether this iTask component works correctly you can open the project `Examples/iTasks/BasicAPIExamples.prj` in the distribution and run it.

Compiling and running is a bit different than with normal `Clean` projects. This is explained briefly in the lecture slides.

1 Basic Edit Tasks

On Brightspave you will find a skeleton that defines a record type `Student` and a list of students. A function called `myTask` can be started as a single iTask with.

```
Start world = doTasks myTask world
```

You are required to make the following task using the given type `Student`. Name these functions `task n` where n corresponds to the label of the enumeration.

1. Make a task to enter a new student.
2. Make a task to enter a list of students.
3. Make a task to update the given `student`.
4. Make a task to select your favourite student form the given list `students`.
5. Make a task to select your favourite student form `students` where you only display the name of the students.
6. Construct a task to select potential partners (as many as you like) from `students`. This task should only display the name of the students and whether they are a Bachelor or Master student.

7. Display the number of students selected as well as their names after finishing the previous task.
8. Define a task to change only the name of `student`. The other fields must be displayed in one way or another, but should not be changeable. Display the resulting `Student`.
9. Define a task that allows the user to enter a student or to select `student` as a default value. The task should display the name of the resulting student.
10. Define tasks to enter the fields of a `Student` individually and compose a student record when all fields are given. Display the constructed record.
11. Define a function `updateBool :: Bool → Task Bool` that enables the user to update a Boolean value (like `updateInformation []`), but uses a drop-down box instead of the standard checkbox. Hint: use `updateChoice`.

Each of these tasks can be written in one or a few lines of code. There is a better solution when you need many lines of code in your solution.

Deadline

The deadline for this exercise is May 10, 2021, 10:30 (just before the next lecture).