Assignment 8

Recommended readings:

- Lecture Slides as starting literature
- https://www.typescriptlang.org/docs/home.html
- https://www.typescriptlang.org/docs/handbook/tsconfig-json.html
- https://www.typescriptlang.org/docs/handbook/compiler-options.html
- https://www.typescriptlang.org/docs/handbook/classes.html#accessors

Exercise 1 – TypeScript I

- First install TypeScript in node.js and then compile the file *ex1/Person.ts* to JavaScript. Execute the JavaScript in node.js. Then compare *Person.ts* and *Person.js* and explain the difference.
- What are the differences between the transpilation options '`--target es5`' and '`--target es6`'? How can you save `es6` as a compiler option for *Person.ts*, such that simply running 'tsc' will produce the same output on any machine?

Exercise 2 – TypeScript II

Start looking at the file *ex2-3/src/Person.ts*, and edit it and create new .ts files in *ex2-3/src* (Professor, ProfessorPosition) such that:

- There is one module per class (each in a separate .ts file), and modules are imported as needed in other classes.
- There is a *Person* class implementing the *PersonI* interface, and a new class for Professors, which extends the *Person* class.
- Persons additionally have a birthdate (type: Date) and a method *getAge()* that returns the current age as a number, as well as a city of residence and a method *getCity()* that returns it as a string. The *sayHello()* method should return a greeting which includes a person's first and last name, age, and city where he/she is from.
- Professors have additionally a property *employeeID*, which is a number.
- Professors can have a professorPosition, which can be "*full*", "*adjunct*" or "*associate*". Use an enum to represent Professor Positions. Add an *introduce()* method for Professors as a greeting, composed by the string returned by *sayHello()*, followed by *'I am a ${professorPosition} professor'*.

Exercise 3 – TypeScript III

Create new .ts files Course, StudyPlan, Student for homonym classes in *ex2-3/src*, such that:

- Courses have an id (type: number), a name, and are assigned to one professor. Courses have a method *assignTo(Professor)* to assign them to a professor. Each course belongs to one *StudyPlan*, which is another class, which has an *id* (type: number) and a *studyPlanName*. A *StudyPlan* has one or more courses, stored in an array. StudyPlans have a method *addCourse(Course)* to add courses to them. Make sure that a course is added only once to the same StudyPlan!
- *Student* is a subclass of *Person*. Students have the additional properties *matriculationNumber* (type: number), *studyPlan*, and *courses*. The property *studyPlan* is an instance of one *StudyPlan* and is set in the constructor. The property *courses* is an array of *course* instances. Students also have the additional methods: *enrol(course)*, *cancel(course), and showCourses()*. The method *showCourses()* returns a list of all enrolled *course* names as a string. Make sure that a student can enroll to a given course only once! Students have also a method *introduce()*, which returns a string composed by the one returned by *sayHello()*, followed by *'I am a student and I study ${studyPlanName}. I enrolled in the following courses: ${showCourses}'*.

To test your implementation, you can use the files "test.html" and "test.js" provided.

Be sure to use proper TypeScript type annotations, suitable visibility modifiers, and that you have implemented the required getter and setter methods for the implementation to work.

Exercise 4 – NodeJS vs. Angular

- What is Angular and how does it relate to TypeScript?
- Discuss the capabilities of Angular in relation to NodeJS, and highlight the differences between the two. Give examples of types of applications for which Angular is more suitable than NodeJS, and examples of applications for which NodeJS is more suitable instead.

Exercise 5 – Angular

With this exercise, you will get familiar with Angular development by completing a tutorial.

- First, set up the environment on your machine: to get started you can follow the instructions at the following link: https://angular.io/guide/setup-local
- Now you are ready to go to https://angular.io/tutorial and follow the Tour of heroes: implement the project following the tutorial step-by-step.