Department Mathematik/Informatik, Abteilung Informatik
Software & System Engineering Group
Software Technology Group
Lecture Object-Oriented Software Engineering, SS 2024

Alexander Korn, Adrian Bajraktari,
Prof. Dr. Michael Felderer

UNIVERSITÄT ZU KÖLN

# Exercise Sheet 1. Object-Oriented Programming

## Exercise 1. Roles in Object-Based Programming

a) Pull `https://github.com/AdrianBajraktari/oose24.git` to get a JavaScript template for this exercise (in `src/exercise1/E1_oop_template.js`). Extend the template as follows:

   1) Create a student object for Alice Wonderland, age 24. Implement a method `getName` that prints the first and last name like this: "Alice Wonderland".

   2) Implement the "functions" `makeTutor`, `makeStudent`, and `makePhD`. In each, first clone the existing role*X* object using the clone "function". Then, add the provided parameters (except for person) as properties to the cloned object (i.e. for student role: `matNr`, `studyProgram`).

   3) Set the parent object of each role*X* to the provided person.

   When done correctly, the overall program should create new role*X* objects which delegate to `alice`, i.e., `alice` takes on different roles during her time at university.

b) How does the memory layout of the object structure from (a) look like? Create a rough sketch.

**[0 points]**

## Exercise 2. Roles in Class-Based Programming

Try to mimic the functionalities realized in exercise 1 using class-based programming in Java by doing the following:

1) Create classes `Person`, `RoleStudent`, `RoleTutor`, `RolePhD`. Add all corresponding attributes and methods to them as in the JavaScript version.

2) Create an interface class `IPerson` with the method `String getName()`. Make all four classes implement the interface.

3) Java has no object inheritance. How can you realize delegation via objects at runtime? Try your own ideas. We will see "the" solution in an upcoming lecture.

**[0 points]**

$\sum$ **0.0 points**