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

Homework 2. Advanced Programming

Hand out: 03.05.2024, 18:00
Hand in: 17.05.2024, 14:00

Exercise 1. Generic Programming

You have been assigned to implement the seminar class. Get the Java template code from `https://github.com/AdrianBajraktari/oose24.git` (in `src/hw2/a1/`). In `Seminar`, we want to store a list of participants of an unknown type. The class is non-generic in the template, so you need to make it generic. Then, implement a method `getParticipantsFromStudyProgram(...)` with the following characteristics:

- It takes a parameter of type `String` that denotes the subject in question.
- The method shall retrieve those participants of the participants list who's subject equals the provided parameter.
- Matching participants shall be stored in a new list that is returned in the end.
- Be aware that you need to make the participants subject somehow accessible (hint: type bounds).
- Think about how you may need wildcards.

As you can see in `main`, the same method should work for `Student` and `PhD` instances.
Hint: The comments in the template will guide you to know where you need to do what.

[4 points]

Exercise 2. Memory Management

You substitute a colleague in the system level development team. Your task for the day is to implement a student storage in C. Get the C template code from `https://github.com/AdrianBajraktari/oose24.git` (in `src/hw2/a2/StudentDataBase.c`). Then, do the following:

- Create a struct type `Student` that stores a `name` and a matriculation number (`matNr`).
- Create a struct type `StudentDataBase` that stores references to the students data in a field `data` and the `size` of the data (i.e., the number of students in the data field).
- In `main`, allocate memory for the student database object, as well as enough memory for the `data` field so that all students can be stored (the number of students is read from console).
- In `deleteStudentDB`, add free(...) calls at the marked locations so that all allocated memory for the student database object, the data field and the name field of each student is deallocated.

Hint: The comments in the template will help you to know where you need to do what.

[4 points]

Exercise 3. Operator Overloading

a) Get the Python template code from `https://github.com/AdrianBajraktari/oose24.git` (in `src/hw2/a3/LearningGroup.py`). Do the following:

- In the `Student` class, override the + operator (`__add__`) so that when two student objects are added, i.e., `student1 + student2`, they become a learning group. That means, a new object of type `LearningGroup` is created, and both students are added to it.
- Implement the class `LearningGroup` so that it stores a list of all students in this group. Override `LearningGroup`'s + operator so that when a student is added to the learning group, i.e., `learningGroup1 + student3`, the student is added to the list of students.

b) Given the above, what is the outcome of `student1 + student2 + student3`? Give a step-by-step explanation of your answer.

[2 points]

$\sum$ **10.0 points**