UNIVERSITÄT BONN

Algorithmen und Programmierung
Übungsblatt 11
WS 2022/23

Dr. Felix Jonathan Boes
Benedikt Bastin, Ellen Bundschuh, Anna Höpfner, Gina Muuss, Adrian Oeyen, Felix Roth, Thore Wolf

Ausgabe: 19.12.2022
Abgabe: keine Abgabe; Präsentation in der Übung

Aufgabe 1 (Hashtable Teil 2)**.** In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_11` angelegt. Erweitern Sie die dort, unter Verwendung der Projektstruktur, Ihre Klasse `Hashtable` und demonstrieren Sie Ihre Implementierung. Ihre Implementierung der Hashtable soll wie folgt erweitert werden.

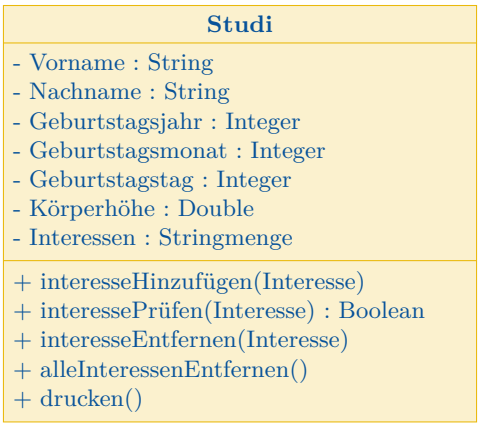
- Die Bucketanzahl soll durch die einfügenden und entfernenden Operationen dynamisch, entsprechend den Vorgaben der Vorlesung, angepasst werden.
- Beim Instanziieren soll eine Hashfunktion übergeben werden können. Implementieren Sie dazu einen entsprechenden Konstruktor.

Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden. Alternativ verwenden Sie das Buildsystem cmake.

```
clang++ -std=c++17 -I./include -I./external -fsanitize=address \
        IHRE_QUELLDATEIEN \
        examples/aufg1.cpp -o aufg1
```

Aufgabe 2 (Gerichtete Graphen als UML-Diagram)**.** In der Vorlesung haben wir die Klasse `GerichteterGraph` implementiert. Beschreiben Sie diese Klasse durch ein UML-Diagram im Sketchingdialekt.

Aufgabe 3 (Studiklasse entlang eines UML-Diagrams implementieren)**.** Das folgende UML-Diagram im Sketchingdialekt beschreibt den Typ `Studi`. Legen Sie in Ihrem Abgabe-Repository einen neuen Branch mit dem Namen `Studi` an. Verwenden Sie dort die übliche Projektstruktur um dort die Klasse `Studi` zu implementieren und demonstrieren Sie Ihre Implementierung. Ihre Implementierung soll alle getter-Methoden (zum Auslesen alle Attribute des Typs) implementieren. Ihre Implementierung soll außerdem sinnvolle setter-Methoden bereit stellen. Achten Sie bei den setter-Methoden darauf, dass diese in jedem Fall zu einem zulässigen Zustand führen.



Aufgabe 4 (Studis sortieren)**.** In der vorherigen Aufgabe haben Sie die Klasse `Studi` implementiert. Legen Sie einen Array von 16 individuellen Studis an und sortieren Sie dieses Array mithilfe der Funktion `std::sort` bezüglich

(1) Vornamen (lexikographisch),

(2) Nachnamen (lexikographisch),

(3) Alter (aufsteigend) und

(4) Körperhöhe (absteigend).

Drucken Sie jeweils das sortierte Array aus.