

Algorithmen und Programmierung

Übungsblatt 8

WS 2022/23

Dr. Felix Jonathan Boes

Benedikt Bastin, Ellen Bundschuh, Anna Höpfner, Gina Muuss, Adrian Oeyen, Felix Roth,
Thore Wolf

Ausgabe: 28.11.2022

Abgabe: keine Abgabe; Präsentation in der Übung

Aufgabe 1 (Einfach verkettete Liste). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_08` angelegt. Implementieren Sie die dort, unter Verwendung der Projektstruktur, die Klasse `List` und demonstrieren Sie Ihre Implementierung.

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 -fsanitize=address \  
    IHRE_QUELLDATEIEN \  
    examples/aufg1.cpp -o aufg1
```

In dieser und den folgenden Aufgaben wird mit dem Parameter `-fsanitize=address` der AddressSanitizer in Ihr Programm integriert. Dieser überprüft, dass einerseits keine Speicherlecks auftreten (also der gesamte Speicher auf dem Heap wieder freigegeben wird) und andererseits alle Zugriffe auf gültigen Speicher erfolgen.

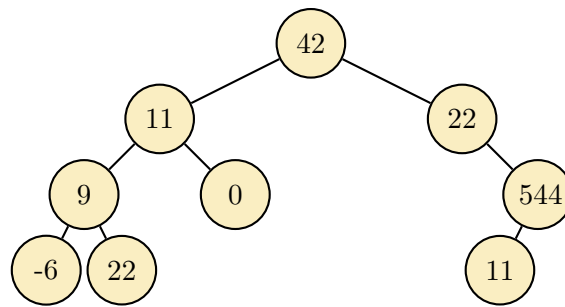
Wenn Sie beim Ausführen des Programms eine Fehlermeldung wie

```
=====
==112251==ERROR: LeakSanitizer: detected memory leaks
Direct leak of 400 byte(s) in 1 object(s) allocated from:
    #0 0x55c0c1107012 in operator new[](unsigned long) (/uebungscore_08/aufg1+0x11a012)
    #1 0x55c0c1109f2d in List::List() (/uebungscore_08/aufg1+0x11cf2d)
    #2 0x55c0c1109e3e in main (/uebungscore_08/aufg1+0x11ce3e)
    #3 0x7f0406b5428f (/usr/lib/libc.so.6+0x2328f)

SUMMARY: AddressSanitizer: 400 byte(s) leaked in 1 allocation(s).
```

sehen, bedeutet das beispielsweise, dass Sie ein Speicherleck haben, also Speicher reserviert, aber nicht wieder freigegeben haben.

Aufgabe 2 (Binärbäume mit Smartpointern). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_08` angelegt. Implementieren Sie die dort, unter Verwendung der Projektstruktur, die Klasse `BinaryTree` mithilfe von Smartpointern und demonstrieren Sie Ihre Implementierung. Erstellen Sie zu diesem Zweck mindestens den folgenden Binärbäume und drucken Sie diesen aus.



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 -fsanitize=address \  
IHRE_QUELLDATEIEN \  
examples/aufg2.cpp -o aufg2
```