

# Übung zur Vorlesung EidP (WS 2018/19)

## Blatt 8

Block rot

**Es können 4 Punkte erreicht werden.**

**Abgabedatum:** 20. Dezember 2018, 23:59 Uhr

### Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter  
<https://ls11-www.cs.tu-dortmund.de/teaching/ep1819uebung/>
- Stellen Sie sicher, dass alle von Ihnen abgegebene Dateien reine Textdateien im UTF-8-Format sind.
- Für die Abgabe sind die Dateien `Aufgabe_08_1.txt`, `Aufgabe_08_2.h`, `Aufgabe_08_2.cpp`, `Aufgabe_08_3.h` und `Aufgabe_08_3.cpp` notwendig.
- Für die Kompilierung muss ebenfalls sichergestellt werden, dass sich Ihre Programme mit den Parametern `-pedantic` und `-Werror` kompilieren lassen.
- Für die Programmieraufgaben kopieren Sie immer die Ergebnisse als Block-Kommentar an das Ende der Datei, welche das jeweilige Hauptprogramm enthält.
- Dies ist das letzte Blatt im zweiten, roten Block.

### Aufgaben

#### Aufgabe 1: Grundlagen (1 Punkt)

Legen Sie für Ihre Antworten eine Text-Datei `Aufgabe_08_1.txt` an.

- a) Wie wird ein ADT formal definiert? (0.1 Punkte)
- b) Nennen Sie mindestens drei Eigenschaften von ADT. (0.2 Punkte)
- c) Erklären Sie den Unterschied zwischen einem Kopierkonstruktor und einer Zuweisung. (0.2 Punkte)
- d) Erklären Sie die Begriffe der flachen bzw. tiefen Klassenkopien. (0.2 Punkte)
- e) Wie kann man automatisch erzeugte Methoden einer Klasse erzwingen bzw. verhindern? Geben Sie jeweils einen Beispiel an. (0.2 Punkte)
- f) Was ist die minimale bzw. maximale Anzahl der Knoten eines binären Suchbaums der Höhe  $n$ ? (0.1 Punkte)

## Aufgabe 2: ADT Schlange (1.5 Punkte)

Die Implementierung des ADT *Schlange* sowie die Testeingabe sind auf der Website der Übung zu finden. Laden Sie für Ihre Abgabe die Dateien `Aufgabe_08_2.h` und `Aufgabe_08_2.cpp` von der Website der Übung herunter.

Erweitern Sie den ADT *Schlange* aus der Vorlesung um die Funktionen `insertAfter` (0.8 Punkte) und `manifold` (0.7 Punkte). Die Funktion `manifold` bearbeitet die gespeicherten Zahlen so, dass an der Stelle jeder ungeraden Zahl zwei Instanzen dieser Zahl vorkommen sollen, und an der Stelle jeder geraden Zahl drei Instanzen dieser Zahl vorkommen sollen. Die Reihenfolge der geraden und ungeraden Zahlen aus der ursprünglichen Schlange soll hierbei unberührt bleiben. Für das Einfügen der Zahlen in die Schlange soll die zu implementierende Funktion `insertAfter` verwendet werden. Diese Funktion liefert einen Zeiger auf das neu eingefügte Objekt zurück. Hilfsschlangen und Hilfsarrays sind an dieser Stelle nicht erlaubt.

**Beispiel:** Für eine Schlange  $A = (1, 2, 2, -6, 5, -3, 7)$  soll die Funktion `manifold` die Schlange  $\bar{A} = (1, 1, 2, 2, 2, 2, 2, 2, -6, -6, -6, 5, 5, -3, -3, 7, 7)$  berechnen.

Kompilieren Sie das Programm, überprüfen Sie Ihre Ergebnisse und kopieren Sie diese als Block-Kommentar an das Ende der cpp-Datei `Aufgabe_08_2.cpp`.

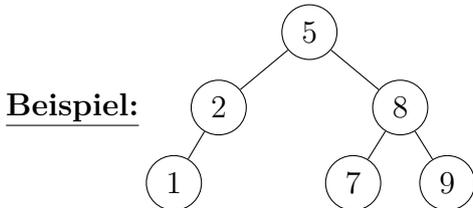
**Hinweis:** Beachten Sie bitte bei der Bearbeitung der Aufgabe, dass die Funktion `manifold` die ursprüngliche Schlange verändert. Der Rückgabewert dieser Funktion muss `void` sein.

### Aufgabe 3: ADT Binärer Suchbaum (1.5 Punkte)

Die Implementierung des ADT *binärer Suchbaum* sowie die Testeingabe sind auf der Website der Übung zu finden. Laden Sie für Ihre Abgabe die Dateien `Aufgabe_08_3.h` und `Aufgabe_08_3.cpp` von der Website der Übung herunter.

In der Vorlesung haben Sie die den ADT *binärer Suchbaum* für den effizienten Zugriff auf Elemente kennengelernt. Erweitern Sie den ADT um die folgenden zwei Methoden:

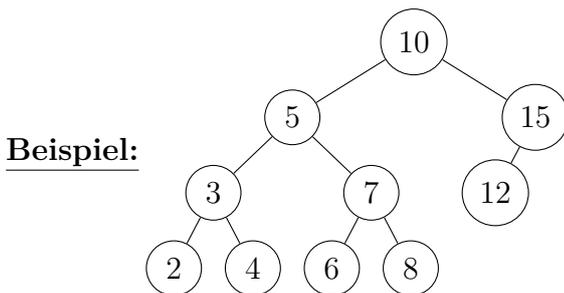
a) Die Methode `unsigned int countNodes(Node *node)` soll für den Suchbaum, für den sie aufgerufen wird, die Anzahl der Knoten des Baums **rekursiv** berechnen und zurückgeben.



Der Aufruf `countNodes()` liefert als Rückgabewert den Wert 6.

Kompilieren Sie Ihr Programm, überprüfen Sie Ihre Ergebnisse und kopieren Sie diese als Block-Kommentar an das Ende der `cpp`-Datei `Aufgabe_08_2.cpp`. (0.7 Punkte)

b) Die Methode `Node* pred(T const &x)` sucht den Knoten, der das gegebene Datum `x` enthält, und gibt dann den Zeiger auf den Knoten mit dem größten Wert im linken Unterbaum mit Wurzel `x` zurück.



Der Aufruf `pred(5)` gibt den Zeiger auf den Knoten 4 zurück.

Wenn kein linker Unterbaum existiert sowie wenn das gegebene Datum `x` im Baum nicht vorhanden ist, soll die Methode `nullptr` zurückgeben.

Testen Sie Ihre Funktion auf allen Knoten des Baums in der Testeingabedatei.

(0.8 Punkte)