technische universität dortmund

fakultät für informatik

Jurij Kuzmic

Dortmund, den 12. Dezember 2019

Übung zur Vorlesung EidP (WS 2019/20)

Blatt 8

Block rot

Es können 4 Punkte erreicht werden.

Abgabedatum: 19. Dezember 2019, 23:59 Uhr

Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter

  `https://ls11-www.cs.tu-dortmund.de/teaching/ep1920uebung/`

- Für die Abgabe sind die jeweils genannten Dateien zu erstellen.
- Stellen Sie sicher, dass alle von Ihnen abgegebene Dateien reine Textdateien im UTF-8-Format sind.
- Für die Kompilierung des Programms muss der C++14-Standard aktiviert sein. Dies kann im Referenzkompiler GCC 6.3 durch den Schalter `-std=c++14` sichergestellt werden. Es sollen zudem die Parameter `-pedantic` und `-Werror` genutzt werden. Der Befehl zum Kompilieren soll somit wie folgt aussehen:

  ```
  g++-6 -pedantic -Werror -std=c++14 Aufgabe.cpp -o Aufgabe
  ```

- Die Verwendung von zusätzlichen **Bibliotheken** zur Lösung der Aufgaben ist **nicht erlaubt**!
- 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 Objektkopien. (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.
In der Vorlesung haben Sie den ADT *Schlange* kennengelernt. Erweitern Sie den ADT um die folgenden zwei Methoden:

a) Die Methode `Objekt* insertAfter(Objekt *e, T const &d)` bekommt das Objekt, nach welchem das neue Objekt hinzugefügt werden soll, und den einzufügenden Wert übergeben. Als Rückgabe soll diese Methode einen Zeiger auf das neu eingefügte Objekt liefern. Hilfsschlangen und Hilfsarrays sind an dieser Stelle nicht erlaubt. (0.8 Punkte)

b) Die Methode `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. Hilfsschlangen und Hilfsarrays sind nicht erlaubt.

Beispiel: Für eine Schlange $A = (3, 1, 2, -4, 1, -5, 9)$ soll die Methode `manifold` die Schlange $A = (3, 3, 1, 1, 2, 2, 2, -4, -4, -4, 1, 1, -5, -5, 9, 9)$ erstellen. (0.7 Punkte)
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`.

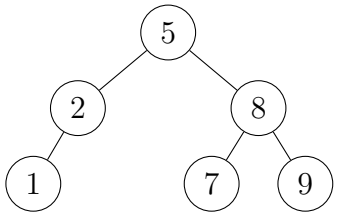
Hinweis: Beachten Sie bitte bei der Bearbeitung der Aufgabe, dass die Methode `manifold` die ursprüngliche Schlange verändert. Der Rückgabewert dieser Methode soll `void` sein.

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 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.

Beispiel:

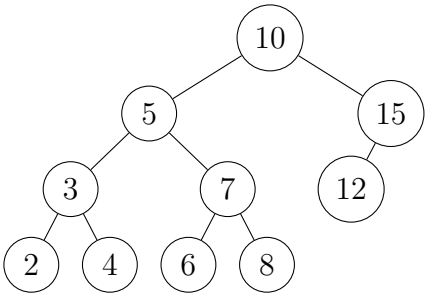


Der Aufruf `countNodes()` liefert als Rückgabewert den Wert `6`.

Kompilieren Sie Ihr Programm, überprüfen Sie Ihre Ergebnisse und kopieren Sie diese als Block-Kommentar an das Ende der cpp-Datei `Aufgabe_08_3.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.

Beispiel:



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)