

Programmieren 1 - WS 2022/23

Prof. Dr. Michael Rohs, Tim Dünke, M.Sc., Jan Feuchter, M.Sc.

Präsenzübung 11

Diese Aufgaben sind zur Lösung während der einstündigen Präsenzübung gedacht.

Aufgabe 1: Binärbäume (ehemalige Klausuraufgabe)

Implementieren Sie die Funktion `IntList* level_bintree(Tree* t, int level, IntList* result)`. Diese soll die Werte in Ebene `level` des Binärbaum `t` in einer Liste sammeln und zurückgeben. Die Wurzel liegt in Ebene 0, deren Kinder in Ebene 1, usw. Die Testfälle zeigen einige Beispiele. Die Liste `result` stellt das bisher ermittelte Zwischenergebnis dar. Zu diesem werden, falls gefunden, weitere Elemente der Ebene `level` hinzugefügt und die Liste zurückgegeben. Damit die Reihenfolge der Werte in der Ergebnisliste stimmt, wird beim Durchsuchen des Baums der rechte vor dem linken Unterbaum durchsucht. Der Eingabebaum darf nicht verändert werden.

Template-Datei: `aufgabe1.c`

Aufgabe 2: Strings und Listen (ehemalige Klausuraufgabe)

Implementieren Sie die Funktionen `find_node` und `total_by_item`. Die Funktion `total_by_item` bekommt eine Zeichenkette `counts` der Form

`"nuts:3,oranges:2,...,nuts:1"`

und eine weitere Zeichenkette `prices` der Form

`"apples:199,oranges:199,..."`

übergeben. In `counts` sind Häufigkeiten von Objekten und in `prices` Preise pro Objekt in Cent codiert. Die Funktion soll eine Liste der Gesamtpreise pro Objekt berechnen. Die Testfälle zeigen einige Beispiele. Die Funktion `find_node` bekommt eine Liste, die durchsucht werden soll, und einen Objektnamen, nach dem gesucht werden soll, übergeben. Falls der Name in der Liste vorkommt, soll der erste zugehörige Knoten (`Node*`) zurückgegeben werden. Sie dürfen `strlen`, (oder `s_length`) und ggf. selbst definierten Hilfsfunktionen verwenden.

Template-Datei: `aufgabe2.c`