Imperative Programmierung

Projektaufgabe

Alle Funktionen wurden, wie schon in der letztenten Hausaufgabe in der Datei „liste.c“ bzw. „liste.h“ implementiert.

In der Datei „projekt_aufgabe.c“ wurde dann das Rahmen Programm geschrieben, dass dann die Funktion der Reihe nach ausführt.

1. Insert:
   Um ein Element an eine bestimmte Stell einzufügen, kann z.B. die Funktion „insertInt()“ bzw. „insertString()“… verwendet werden. Als drittes Argument kann dabei der Index übergeben werden. Wenn kein Index übergeben wird, wird das Element an der aktuellen Stelle des Zeigers „current“ eingefügt. Wenn ein Index übergeben wurde, wird einfach vor dem Einfügen „current“ auf das passende Element gesetzt.
   Dann beginnt der Einfüge Prozess in der Funktion „insert()“. Falls das „current“ Element dabei an letzter Stelle steht. Wird das Element mit „append()“ ans Ende der Liste gefügt. Wenn nicht wird ein neues Element erstellt und vom neuen Element das „next“ Attribut auf das „current“ Element gesetzt. Falls dann das „current“ Element nicht das erste Element ist, muss auch das „next“-Attribut vom vorherigen Element auf das neue Element gesetzt werden.
   Das vorherige Element kann mit der Funktion „getPrevious()“ gefunden werden.
2. Delete:
   Dies wurde in „removeAtIndex()“ umgesetzt. Dabei wird zuerst, dass aktuelle Element auf das Element am Index gesetzt. Dazu wird eine for-Schleife genutzt. Danach wird unterschieden, ob das Element sich an erster Stelle befindet oder nicht. Wenn dann wird einfach der „Kopf“ der Liste auf das nächste Element gesetzt. Wenn nicht dann muss das „next“ Attribut vom letzten Element auf das nächste Element gesetzt werden.
3. Swap:
   Dies wurde in der Funktion „swapAtIndex()“ ungesetzt. Dabei werden einfach die Nodes an den beiden Indizes genommen und die Referenzen auf die darin gespeicherten Daten vertauscht.