

Team: Team 06, Corinna Klaukin und Anna Christin Steinhauer

Aufgabenaufteilung: Skizze gemeinsam

Quellenangaben: -

Begründung für Codeübernahme: -

Bearbeitungszeitraum: 01.10. Skizze 30 Minuten

Aktueller Stand: <Welche Teile der Software sind fertig inklusive Tests, welche sind fertig, aber noch nicht getestet, welche müssen noch implementiert werden>

Änderungen in der Skizze: -

Skizze:

ADT Liste:

Die Position ist eine ganze natürliche Zahl.

create: wir legen intern eine neue Liste ohne Elemente an

isEmpty: wenn die Startposition noch mit keinem Element belegt ist, ist die Liste leer und true wird zurückgegeben, ansonsten wird false zurückgegeben

length: die höchste belegte Position ist die Länge der ADT Liste

insert: an der vorgegebenen Position wird das Element in der Liste abgespeichert; das ursprünglich an dieser Position und alle folgenden Elemente werden um eine Position nach rechts verschoben

delete: das Element wird an der vorgegebenen Position aus der Liste gelöscht, alle Elemente die auf einer höheren Position liegen, werden um eine Position niedriger verschoben

find: die Suche startet an der 1. Position und vergleicht das dort gespeicherte Element mit dem vorgegebenen. Bei Gleichheit wird die Suche abgebrochen, ansonsten wird die Position um 1 erhöht und das dort befindliche Element verglichen, usw. bis das Element gefunden wurde. Wenn die aktuelle Position die Länge der Liste überschreitet, wird die Suche auch abgebrochen.

retrieve: das Element an der vorgegebenen Position wird zurückgegeben

concat: Es wird eine neue Liste angelegt in die zuerst alle Elemente aus Liste 1 und danach die Elemente der Liste 2 eingefügt werden.

ADT Stack:

createS: durch internes create von ADT Liste einen Stack anlegen

push: mit insert an erster Position das Element in der Liste einfügen

pop: mit delete das Element an erster Stelle aus der Liste löschen

top: das Element an erster Stelle ausgeben

isEmptyS: delegieren an isEmpty der Liste

ADT Queue:

createQ: es werden intern zwei neue ADT Stacks ohne Elemente angelegt. Einer ist der Eingabestack, der andere der Ausgabestack

front: das zuerst hinzugefügte Element wird ausgegeben. Ist der Ausgabestack leer, werden zuerst alle aus dem Eingabestack in den Ausgabestack umgestapelt.

enqueue: delegiert mit push wird das angegebene Element auf dem Eingabestack abgelegt

dequeue: das zuerst hinzugefügte Element (das erste Element des Ausgabestacks) wird entfernt. Ist der Ausgabestack leer, werden zuerst alle aus dem Eingabestack in den Ausgabestack umgestapelt.

isEmptyQ: delegiert mit isEmptyS geprüft, ob sich Elemente im Eingabestack befinden

ADT Array:

initA: mit create eine ADT Liste anlegen

setA: Als erstes die Position um 1 erhöhen. Dann das alte Element, sofern vorhanden von der Position löschen. Danach mit insert das neue Element in die Liste einfügen. Wenn die Position höher ist als die ursprüngliche Länge, dann werden die dazwischen liegenden Positionen mit 0 initialisiert.

getA: die Position um 1 erhöhen und mit retrieve aus der Liste auslesen. Gibt es an der Position kein Element, so wird 0 zurückgegeben.

lengthA: delegieren an laenge der ADT Liste

Betreff: Re: AD Praktikum Team 6

Von: Christoph Klauck <christoph.klauck@haw-hamburg.de>

Datum: 03.10.2014 21:33

An: "Klaukin, Corinna" <corinna.klaukin@haw-hamburg.de>

Kopie (CC): "Steinhauer, Anna Christin" <annachristin.steinhauer@haw-hamburg.de>

Hallo zusammen!

Zusammenfassung: die Skizze ist nicht ausgewogen: auf der einen Seite detaillierte (nicht begründete/motivierte) Vorgaben, die jedoch teilweise implizite detaillierte Vorgaben erwartet.

Anmerkungen im Detail:

Ihre Funktionen haben keine Signatur! Dies ist aber wichtig. Sofern Sie voraussetzen, dass man auch die Aufgabenstellung liest bzw. benötigt, dies unbedingt anmerken!

"concat: Es wird eine neue Liste angelegt in die zuerst alle Elemente aus Liste 1 und danach die Elemente der Liste 2 eingefügt werden." warum geben Sie hier einen Algorithmus vor, statt das erwartete Ergebnis zu beschreiben?

ACHTUNG: "isEmptyQ: delegiert mit isEmptyS geprüft, ob sich Elemente im Eingabestack befinden" ist falsch, da beide stacks leer sein müssen!

"Als erstes die Position um 1 erhöhen. Dann das alte Element, sofern vorhanden von der Position löschen. Danach mit insert das neue Element in die Liste einfügen. Wenn die Position höher ist als die ursprüngliche Länge, dann werden die dazwischen liegenden Positionen mit 0 initialisiert." ist ein zu ungenauer detaillierter Algorithmus: welche Position soll um 1 erhöht werden? und wozu? und wer ist das alte Element? und was bedeutet der letzte Satz? das die Elemente explizit erzeugt werden sollen?

"lengthA: delegieren an laenge der ADT Liste" funktioniert nur, wenn man eine gewisse Umsetzung voraussetzt, die Sie aber nicht explizit ansprechen, sondern scheinbar als selbstverständlich erwarten. Da aufpassen: selbst in der Informatik gibt es unterschiedliche als logisch betrachtete Selbstverständlichkeiten...

Die Skizze ist angenommen

Gruß

C Klauck

Am 02.10.2014 um 20:01 schrieb Klaukin, Corinna:

Hallo,

hier kommt unsere Skizze für die erste Aufgabe.

Gruß

Corinna Klaukin