ADT_SingleLinkedList Specificatie datastructuren college2

Dit concept aanvullen (huiswerk voor 17 sept) tot een volwaardige specificatie

Elementen:

Elementen zijn knopen;

elke knoop bevat een data_element en een pointer naar een knoop

Type: KnoopPointer = ^Knoop;

Knoop = record

Data_Element: DataType;

Next: KnoopPointer

end;

Structuur: relatie is lineair, next_pointer wijst naar opvolger; next_pointer laatste knoop is nil.

Twee extra pointers wijzen naar voorste knoop (head_pointer) en huidige knoop (current_pointer)

Domein: het aantal knopen in de lijst is eindig

Operaties: een 10 tal; operaties vinden plaats bij current_knoop; bij elke operatie moet duidelijk zijn waar de head_pointer en current_pointer na afloop naar wijzen

Procedure **FindFirst**;

Preconditie: linked list is niet leeg

Postconditie: current_pointer wijst naar voorste knoop (net als head_pointer)

Procedure **FindNext**;

Preconditie: next_pointer van current_knoop is niet nil

Postconditie: de volgende knoop waar de next_pointer van de current_knoop naar verwijst wordt de current_pointer

Procedure **Retrieve**(var elem:DataType);

Preconditie: linked list is niet leeg (current-pointer wijst niet naar nil)

Postconditie: elem heeft als waarde die van Data_element in current Knoop gekregen

Procedure **Update**(var elem:DataType);

Preconditie: linked list is niet leeg (current_pointer wijst niet naar nil)

Postconditie: Data_element current knoop heeft waarde elem gekregen; pointers blijven ongewijzigd

Procedure **Insert**(var elem:DataType);

Preconditie: er is nog geheugen voor een nieuwe knoop

Postconditie: een nieuwe knoop met als Date_element elem en als next_pointer waarde van oude current_pointer wordt toegevoegd; oude current_pointer wijst naar nieuwe knoop; head_pointer blijft gelijk.

Procedure **Delete**;

Preconditie: linked list is niet leeg

Postconditie: current_knoop verwijderd uit de lijst; voorganger's next_pointer moet z'n next pointer overnemen; current_pointer wijst naar deze voorganger, als afwezig naar eerste, als ook die afwezig (lege lijst) dan naar nil. Als head_pointer gelijk aan current_pointer was, krijgt deze de zelfde waarde als de gewijzigde current_pointer.

Function **Empty:**boolean;

Preconditie: linked list moet bestaan

Postconditie: Y als linked list leeg (head_pointer=nil) anders N

Function **Last**:boolean; #is current_knoop laatste in lijst?

Preconditie: linked list moet bestaan

Postconditie: Y als next_pointer van current_knoop is nil en Data_element niet nil, anders N

Procedure **Create**;

Postconditie: een lege linked list wordt aangemaakt; head-pointer and current_pointer zijn nil.