Daniel Schween

07/20/2024

CS 300 SNHU

**Module 3 LinkedList**

Initially, we define the main Class ‘LinkedList’. This class serves as a hub to implement the functions and structures used to modify and manipulate bids in the linked list. The class implements several functions, such as, append, prepend, Print, Remove, Search, and Size. The Class also implements a pointer to the next node in the linked list. Additionally, the class ‘LinkedList’ supports private members head, tail, and size.

Additionally, several methods are used in the class to support and manipulate the results of bids. String to double is used to convert a CSV file data into usable values in the class. ‘Bid’ is a structure containing the data used in a vector that will be sorted. ‘displayBid’ is used to send the bid values to the console while ‘getBid’ allows a user to append/prepend a bid to the linked list. ‘loadBids’ is used to read the CSV data (bids) into the program so we can use functions to manipulate. After the functions and methods of the linked list, a main class is used as the driver and menu selection.

**Pseudocode:**

**LinkedList::Prepend(Bid)**

**Create** a new node for Bid

**If** head’s next pointer is not NULL

**Set** the new Node as the head

**Set** size to one larger

**Else**

**Call Append()**

**End**

**LinkedList::PrintList()**

**Create** a new Node

**Loop** through the list starting at the head

**Output** to console: bid, title, amount, fund

**End**

**LinkedList::Search(String)**

**Create** a new node pointer called cursor

**Set** cursor to the head

**Loop** until cursor is NULL (end of the list)

**If** the Node at cursor contains a bid equal to string

**Return** cursor

**Set** cursor equal to the next Node

**End**

**LinkedList::Remove(String)**

**Create** a new Node pointer called cursor and **Set** to head

**Create** a new Node called temp node

**Loop** cursor from head node to length minus 1 node

**If** head matches String

**Set** head equal to cursor’s next pointer

**Delete** cursor memory

**Set** size to one less

**If** tail matches String

**Set** node equal to cursor’s next pointer

**Set** tail equal to cursor

**Set** cursor’s next pointer to NULL since its now the tail

**Delete** temp node

**Set** size to one less

If the cursor Node contains a bid equal to String

**Set** temp node equal to cursor’s next pointer

**Set** cursor equal to temp node; now 2 copies of Node following cursor

**Delete** temp node

**Set** size to one less

**Set** cursor equal to the Node pointed at in next

**End**