ListADT

- elements : ListElement [ ]

- numItems : int

- current : int

- exists : boolean

+ ListADT(void)

Default constructor: Initializes the ListADT to an unusable state.

+ ListADT(specifiedSize : int)

Parameterized constructor: Establishes a new "list" of a specified capacity that will be in an empty and usable state.

+ Add(providedElement : ListElement) : void

Adds a copy of the provided element to the list.

+ AtEnd(void) : boolean {end}

Identifies if we are beyond the end of the list data.

+ Count(void) : int {numItems}

Gives back the number of elements in the ListADT.

+ Create(specifiedSize : int) : void

Establishes a new "list" of a specified capacity that will be in an empty and usable state.

+ Delete(searchValue : String) : boolean {deleted}

Locates the matching element and removes it if found.

+ Destroy(void) : void

Sets the list back to an unusable state.

+ Exists(void) : boolean {exists}

Identifies if the list was "created".

+ GetNext(void) : boolean {success}

Sets the "current" list location to the next element.

+ IsEmpty(void) : boolean {empty}

Identifies if the list is empty.

+ IsFull(void) : boolean {full}

Identifies if the list has no remaining capacity.

+ Reset(void) : void

Sets the "current" list location to the first element.

+ Retrieve(void) : ListElement {currentElement}

Gives back a copy of the "current" list element.

+ Search(searchValue : String) : boolean {found}

Locates an element within the ListADT whose "key" matches the given searchValue.

ListElement

- String : vehicleID

- int : year

- String : make

- String : model

- int : miles

- String : classification

- double : price

+ CreateNewVehicle(String : vehicleID, int : year, String : make, String : model, int : miles, String : class, double : price) : void

Creates a New Vehicle based on passed in information.

+ GetVehicleID(void) : String {vehicleID}

Returns the vehicleID.

+ GetVehicleYear(void) : int {year}

Returns the vehicle year.

+ GetVehicleMake(void) : String {make}

Returns the vehicle make.

+ GetVehicleModel(void) : int {miles}

Returns the vehicle miles.

+ GetVehicleClassification(void) : String {classification}

Returns the vehicle classification.

+ GetVehiclePrice(void) : double {price}

Returns the vehicle price.

+ Clone(void) : ListElement {clonedElement}

Creates a Deep Copy.

Lab03

None.

+ main(args : String[]) : void

+ GetInventoryFile(void) : String {filename}

Gets a file name from the User using a prompt.

+PerformUserChoice(ListADT : carList) : void

Performs choices determined by the user until the user decides to quit.

+GetUserChoice(void) : char {userChoice}

Gets a choice from the user.

+LoadFromUserFile(ListADT : carList) : void

GetInventoryFile. Loads the list from the user specified file.

+SaveToFile(ListADT : carList) : void

GetInventoryFile. Uploads the current carList to the user specified file.

+AddVehicle(ListADT : carList) : void

Makes a new vehicle and adds the list to the carList. Gets the vehicle attributes from the user.

+RemoveVehicle(ListADT : carList) : void

GetVehicleAttribute. Checks to see if the vehicle exists in the list. If the vehicle exists, remove from the carList.

+DisplayAllVehicles(ListADT : carList) : void

Displays all vehicles from carList.

+DisplaySingleVehicle(ListADT : carList) : void

GetVehicleAttributes. Then searches carList for matching information. If it matches, it gets displayed, if it does not exists, it lets the user know.

+DisplayVehicleSeries(ListADT : carList) : void

Gets vehicle year and class from user, then checks carList for matches and displays each vehicle with that match.

+ModifyPrice(ListADT: carList) : void

GetVehicleAttributes. Checks carList to see if the vehicle exists, if it does, then get newPrice from user and change currentPrice in carList.

+GetVehicleAttributes(void) ListElement{userVehicle}

Creates a new vehicle with user entered elements and retuns it to previous function.