SHUFFLE DOCUMENTATION

Sequence is a class template, implemented as a singly-linked list. The elements of the list are Nodes that store parameters Key and Info. The first one is recognition parameter and the other is the information stored in the Node. The program is meant to shuffle two sequences with elements of given data: Key and Info.

Methods:

Sequence(const Sequence<Key, Info>& sourceSequence) -*sequence constructor*

~Sequence(); *sequence destructor*

Sequence(const Sequence<Key, Info>& sourceSequence) -*copy constructor, creates the same new element fas in the argument*

void addElementFront(const Key& nowKey, const Info& nowInfo) -*adds a new element to the Sequence at its top*

bool addAfter(const Key& nowKey, const Info& nowInfo, const Key& afterId); -*adds argument after a particular existing element, the new element is not added if the existing one can't be found*

void addElementBack(const Key& nowKey, const Info& nowInfo) -*adds new element to the bottom of the Sequence*

bool isListEmpty() -*checks if the sequence has any elements (returns 1 for no elements, 0 for existing elements in the sequence)*

void display() -*displays sequence*

void copyElements(int howMany, int startPoint, const Sequence<Key, Info>& sourceSeq) -*copies a block of elements from previously set point (startPoint) . Function howMany defines what number of elements will be copied.*

bool eraseElement(const Key& erasedId, const Info& erasedInfo); -*removes an element of defined Info and Key, if such element cannot be found, nothing is done*

bool eraseElement(const Key& erasedId, const Info& erasedInfo) -*erases element*

void eraseElements() *-erases elements using private function erasAllElements()*

void changeInfo(const Info& prevInfo, const Info& nowInfo) *-changes the value of every element which has prevInfo as info*

void changeKey(const Info& prevKey, const Info& nowKey) *-changes the Key of every element in the sequence of prevKey value to the nowKey*

operator=(const Sequence<Key, Info>& rightHS) *-assignment operator* =

operator+(const Sequence<Key, Info>& rightHS) const *-operator* +

int countElements()const *-counts number of elements in the sequence*

.

Void shuffle(...)

This funtion shuffles two desired sequences into an another sequence, it uses the length of the blocks to copy and starting points, then it copies using method copyElements for both sequences. In the end, it copies the result to the new sequence.