EADS

LAB 1 Project Report

Single Linked List

Keshav Dandeva
Student ID : 302333

Introduction

This project report is about implementing single linked list in a class Sequence and class Counter with an external function for Counting words while also using templates. The Sequence Class depicts various methods and operations possible on a single linked list. The external function counts the occurrence of various words from either keyboard input by the user or by any text file.

Sequence Class

The sequence class contains various public member functions for different utilities and a private structure Node for linked list. The template used for Sequence class is *<typename val_type>.*

- struct Node
  It contains only two members i.e. value and *next. Value is the val_type data stored in the node and *next is used to create linked list. It has an object *head which is always the first element of the linked list.

- Sequence();
  It is a constructor of the class which initialises head to null value.

- ~Sequence();
  It is a destructor which deletes all the elements of the object when it goes out of scope.

- Sequence(const Sequence<Val_type> &newSequence);
  It is a copy constructor which adds the element of the object of the class to the new object.

- void pushFront(const Val_type &val);
  This function adds the element to the starting of the linked list.

- void pushBack(const Val_type &val);
  This function adds the element to the end of the linked list.

- void removeFront();
  This function removes the element present at the top of the linked list i.e. head

- void removeBack();
  This function removes the element present at the end of the linked list i.e. tail

- void deleteAll();
  This function deletes all the elements of the linked list.

- bool valExists(const Val_type &val);
  This function checks if the value asked is present in the list or not and returns true or false.

- Val_type getVal(int index);

This function return the value present at the index asked for.

- bool removeVal(const Val_type &val);
  This function deletes the element containing the value asked for.

- void print();
  This function prints the list

- int length() const;
  This function returns the length of the list.

- void insertAtIndex(const Val_type &val, int index);
  This function inserts the element at the index asked for.

- Sequence<Val_type> &operator=(const Sequence<Val_type> &sequence);
  This function is used to define assignment operator '=' that copies the data of the object of the class in another object.

- Sequence<Val_type> &operator+(const Sequence<Val_type> &sequence);
  This function is used to define addition operator '+' that adds the list of the object of the class with another list present in another object.

Counter Class

The counter class contains various public member functions for different utilities and a private structure Node for linked list. The template used for Counter class is *<typename Key>.*

- struct Node
  It contains three members i.e. *Key key; unsigned int occurence; Node* next;* key is the Key type data stored in the node, occurrence is for count of times the key appears in the text and *next is used to create linked list. It has an object *head which is always the first element of the linked list.

- Counter();
  This is a default constructor of the class and it initialises head to null.

- ~Counter();
  This is a default destructor of class which deletes all elements of the object when it goes out of scope.

- bool Keyexists(const Key &key);
  This function is used to check whether a certain key is present in the list or not. It returns true or false value.

- int Keyoccurrence (const Key &key);
  This function is used to find occurrence of a specific key in the list.

- void insert(const Key &key);
  This function is used to insert a key to the list and if the key already exists it increases the occurrence of the key by 1.

- void print();
  This function is used to print 15 words from the list and their count of occurrence.

- unsigned int length() const;
  This function returns the total length of the list i.e. total number of unique keys present in the list.

- unsigned int occurence_sum() const;
  This function returns the total number of keys in the list.

Count Words Function

- int count_words(Counter<string>& cnt, istream& source)
  This is a global function that does the main work for the counter class. It cleans the text document from punctations and converts the text to lower case and numbers to string. Then inserts the words to list using insert function of counter class.

Testing

1. Sequence Class

All the functions and their all possible behaviour is tested for this class. First simple objects of the class are created for data type int:
Sequence<int> s1;
Sequence<int> s2;
Then pushback() and pushfront() methods are used to add elements to the list at the end and at the top respectively. Then the lists are printed using print() function. Also, the removefront() and removeback() functions are tested to remove the first and last element of the list.

Then the addition operator and the assignment operator are tested by creating new object and using the previous two objects as follows:
Sequence<int> s3 = s1 + s2
s3.print();
Sequence<int> s4(s3);
s4.print();

Then the getval() function is tested for values of negative index, out of bounds index and finally a index present in the list.

Then the valExist() function is tested by entering a value present in the list and entering a value not present in the list.

Then the length function is tested which provided the accurate length of the list.

Then the insertAtIndex() function is tested for various possible scenarios like:
1. inserting an element at the head of the list
2. inserting another on the same index i.e. head
3. inserting an element at the last index
4. inserting another element on the last index
5. inserting an element at an index that is out of bounds
6. inserting an element at a negative index
7. inserting an element in the middle of the list

Finally, the destructors for all the objects are tested.

2. Counter Class & Count Words function

This class is tested with the count words function. In this class most of the members are tested while executing the 2 test methods. They are:

1.Input by Keyboard :

In this method the object of counter class is created with <string> type and the user is asked to enter strings. Then the count_words() function is tested and the list is printed with print() function. The length() function and occurrence_sum() function are also tested in this. Finally, the destructor is called and tested for the object of the class.

2.Text File

In this method the object of counter class is created with <string> type and file handling is used to read the text file provided for testing. Then the count_words() function is tested and the list is printed with print() function. Also, user can find the occurrence of a specific word by using the Keyoccurence() function. The length() function and occurrence_sum() function are also tested in this. Finally, the destructor is called and tested for the object of the class.