Singly-linked list

Borys Bondos

1. Task:

The aim of the exercise was to create singly linked list which is able to store data of arbitrary type (templates). Every node in the sequence has got its own unique key and info which is not unique.

We had to write implementation of the external function int split() which is used to create nodes in two objects class Sequence using algorithm based on source sequence.

2.Design of the calass:

In private section: there is a structure which keeps data of given node and pointer to the next element of the list.

There is also a pointer node* head which is created to determine the beginning of the list.

In public section: There is set of methods of the class Sequence:

Sequence(); //constructor

Sequence(const Sequence &source); //copy constructor

~Sequence(); //destructor void add_node(key Key, info

val); //add node at the end void del_node(key delId);

//remove node by key void print_sequence(); //print

every node

Sequence& operator=(const Sequence & source); //overloaded operator =

Sequence& operator+(const Sequence & n); //overloaded operator+ void

del_by_value(info del_value); //remove node by value void del_sequence(); //delete

whole sequence void add_before(key before_id, key _id, info _value);// add node at

specified position

Methods giving ***acces to read*** necessary data by external functions: int

seq_length()const; //gives info about length of the sequence key

get_key(int start_position, int sequence_length)const; //info about key info

get_info(int start_position, int sequence_length)const; //info about info

int start_position is the index of the node from which we are reading data. Index is a place of the node in the sequence and indexing starts from 0.

<u>3.</u> <u>Concept of the function int split()</u>

Thanks to the last 3 methods of the class Sequence, external function has got acces to read necessary data to create new sequences.

Because of the fact that those functions are read only, our source data is safe and encapsulation is preserved.

To create new sequences function int split() uses public methods such as add_node() etc.

<u>4.</u> <u>Testing</u>

Testing scenarios cover proper and improper usage of every function and method.

Descripnion of every test scenario is shown during execution of the program.

Testing is devided into parts:

- Function split test
- Copy constructor test
- Add methods test
- Remove methods test
- Overloaded operators test

Despite the fact that testing is divided into parts, we do not test only one type of methods or functions in each scenario.

Every scenario contains more methods which are used during the execution of the program.