LABORATORIES EXERCISE NR 2

INDEX NUMBER: 293071

NAME: TOMASZ MIAZGA

GROUP: 103

1. FILES

The whole project consists of five files:

- DLR.h
- Produce.h
- Testing.h
- Testing.cpp
- main.cpp

First file contains DLR class with Iterator class inside. Function produce is held in *Produce.h. Testing.h* and *Testing.cpp* keeps Testing class methods (their implementation and declaration). Main.cpp is used to induce Testing class object and its methods.

2. DLR CLASS

DLR class is the most important part of the program. It is combined of the struct called Node:

```
struct Node {
	Key key;
	Info info;
	Node * next;
	Node * prev;
};

Node * door = nullptr;
```

The door pointer is the one that grants access to the doubly-linked ring. The ring, as supposed consists of Nodes which are dynamically allocated. The methods and operators of the class:

```
		MEMBER METHODS:

DLR();
DLR(int);
DLR(const DLR<int, int> &)
~DLR();
DLR<Key, Info> &operator=(const DLR<Key, Info> &);

		METHODS:

void pushNext(Key, Info);
void pushPrev(Key, Info);
void createRandomNodes(int);
void displayNode(int)
void displayRing();
void clearRing();
void deleteByKey();

bool checkKeyInfoType();
```

```
bool containsKey(Key);
bool isEmpty();

Iterator begin() const

            OVERLOADED OPERATORS:

DLR<Key, Info> &operator+=(const DLR<Key, Info> &);
bool operator==(const DLR<Key, Info> &);
bool operator!=(const DLR<Key, Info> &);
```

The most extraordinary methods:

- void createRandomNodes(int) – it is used solely in testing (also in DLR(int) constructor). Method creates given number of nodes type <int, int> with random parameters (the values of key and info are randomly generated in range from 1 to 1000)
- bool checkKeyInfoType() – its purpose is to check if Key and Info types are 'int'. It is only used in reateRandomNodes(int). If the types are different than 'int' the mentioned method wont work and will print a message.

What is worth mentioning, there is one additional class parameter called 'length'. It is of 'int' type and stores number of Nodes in the ring.

Inside this class another class is declared and implemented – Iterator class. It allows user to use iterators in operating on DLR objects and produce function (however, the methods of DLR don't use the iterators alone). It overloads standard operators: ++,--,+,-,=,==,!=. Extra methods: `Key& getKey()` and `Info& getInfo()`. They respectively return key and info from the iterator pointed on given Node.

3. PRODUCE FUNCTION

The function is contained in *Produce.h* file. Its purpose is to take two rings as parameters (additionally 3 booleans and 5 integers) and return a ring which is a combination of nodes from both passed rings using iterators.

4. TEST CLASS

Class is supposed to test methods from DLR class. The methods of Testing class are defined in *Testing.h* file and implented in *Testing.cpp*. They are the only methods used in main.cpp file. This file creates Testing object and 'uses' all methods from this class in order to test DLR class. Additionally, the produce function is tested in this class.

List of Testing class methods:

- displayTest()

It tests displayRing() and displayNode(int) methods.

- addingNodesTest()

It tests pushNext(Key,Info), pushPrev(Key,Info), createRandomNodes(int)

- constructorsTest()

It test all DLR class constructors

- operatorsTest()

It tests DLR class overloaded operators

- deletingNodesTest()

It tests deleteByKey(Key) and clearList()

- produceTest()

It tests produce function