**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

**отчет**

**по лабораторной работе №1**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: «Собственный двунаправленный список с внешним итератором»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6382 |  | Черкасова Е.И. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2017

**Цель работы.**

Создать собственный двунаправленный список с внешним итератором.

**Постановка задачи.**

Реализовать возможность добавления элемента в список в начало, в конец и по номеру, удаления из начала, из конца или по номеру. Описать итератор. Перегрузить некоторые функции. Использовать шаблонные функции и классы.

**Реализация.**

Для решения задачи используется интерфейс классов Elem, List, Iter. Перегружены следующие операторы(template <typename T>):

1. Elem<T> &Iter<T>::operator\*()
2. Elem<T> \*Iter<T>::operator->()
3. Iter<T> Iter<T>::operator++()
4. Iter<T> Iter<T>::operator--()
5. Iter<T> &Iter<T>::operator=(const Iter<T> &r)
6. bool Iter<T>::operator!=(Iter<T> const &r)
7. T List<T>::operator [] (size\_t index);
8. List<T> &List<T>::operator= (const List<T> &r);

Функции (1) - (2) возвращают ссылку и указатель на объект контейнера соответственно, (3) - (4) - сам итератор (его копию), а (5) - ая - ссылку на него, функция (6) - ая возвращает true в случае, если переданный элемент контейнера и текущий не равны и false - когда равны.

Функции (1) - (4) входных данных не имеют, (5) - (6) принимают ссылку на итератор.

Функция (7) позволяет получить значение элемента контейнера, (8) - ая - сравнить 2 объекта типа Elem.

Для осуществления следующего взаимодействия с объектами контейнера

for( Elem<int> &elem : list){

elem.get()=1;

}

были описаны функции 5, 6 (описанные выше), а также Iter<T> begin() и Iter<T> end(), переставляющие интератор в начало и конец списка соотетсвенно.

**Вывод.**

В результате выполнения лабораторной работы был создан шаблонный контейнер с внешним итератором, перегружены необходимые функции, освоена работа с дружественными и шаблонными классами, а также некоторые особенности с++11. Разобран принцип написания make файлов, значение const в зависимости от положения в коде, описание конструктора.