



**СУ “Св. Климент Охридски”,
ФМИ – Софтуерно инженерство
Курсов проект по Обектно-ориентирано
програмиране**

List

Габриела Грозданова Грозева, Факултетен № 61850

Съдържание

1. Въведение.....	3
2. Описание на програмния код.....	3-4
3. Използвани технологии.....	4

1. Въведение

List е шаблонен клас, който позволява да бъде създадена променлива от тип двусвързан списък.

Двусвързаният списък представлява линейна структура от свързани еднотипни компоненти. Компонентите на листа са динамични променливи от тип запис с три полета:

- 1) Поле за съхранение на информация от даден тип
- 2) Поле съдържащо указател за предходния компонент
- 3) Поле съдържащо указател за следващия компонент

За да се означаи началото на листа е използван указател `start`, а за да се означаи края на листа – указател `_end`.

Този клас дава възможност да се трият и добавят елементи както в началото на листа, така и в края му. Също така стойността на елементите в края и началото могат да бъдат изведени с помощта на специални функции. Потребителя, може да изтрие всички елементи на списъка, да види дали той е празен и колко елемента съдържа. Достъпа до компонентите на листа се осъществява с помощта на променлива от тип `iterator`. `Iterator`-а позволява да се трият и добавят елементи, да се обхожда списъка, както и да се извежда стойността на даден елемент.

2. Описание на програмния код

List е клас който съдържа структура `Node` – типа на компонентите на двусвързания списък, както и клас `iterator`.

Член функциите на List са:

- `List ();` - конструктор
- `~List ();` - деструктор
- `void push_front (const T& value);` – добавя нов елемент в началото на списъка
- `void push_back (const T& value);`– добавя нов елемент в края на списъка
- `T& front();` - връща стойността в началото на списъка
- `T& back();` - връща стойността в края на списъка
- `void pop_back();` - изтрива елемента в края на списъка
- `void pop_front();` - изтрива елемента в началото на списъка
- `int size ();` - връща броя на елементите в списъка
- `bool empty () const ;` - проверява дали списъка е празен
- `void clear();` - изтрива всички елементи на списъка

Имплементираният клас `iterator` с член - дана указател към `Node` има конструктор по подразбиране (`iterator ();`) и следните предефинирани оператори:

- ❖ `T& operator*();` - Връща стойността на даден `Node(data)`
- ❖ `iterator& operator++();` - префиксен оператор за инкрементиране (`it = ++v.begin()`)
- ❖ `iterator& operator++(int);` - постфиксен оператор за инкрементиране (`it = v.begin()++`)
- ❖ `bool operator!=(const iterator&) const;` -проверява дали адресите на два `Node`-а са различни

В класа `List` са добавени следните функции:

- `iterator begin();` - връща `iterator`- а към началото на списъка
- `iterator end();` - връща `iterator` – а към края на списъка (един елемент след края на списъка)
- `void insert(iterator & it ,const T& value);` - вмъква елемент със стойност `value` на позиция `iterator`
- `void erase(iterator & it);` - изтрива елемент на позиция `iterator`

3.Използвани технологии

- ❖ Език за програмиране: C++
- ❖ Среда за разработка: Visual Studio 2013