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

Двусвързан списък

Полина Красимирова Кръстева, Факултетен № 61789

Съдържание

1. Въведение

В проекта е реализиран двусвързан списък. Той представлява линейна структура от свързани еднотипни компоненти. Двусвръзаният списък е реализирн чрез шаблонен клас. Компонентите му са динамични променливи от тип запис с три полета:

1. Информационно поле стойност
2. Две указателни полета – указващи предходната и следващата компонента в двусвързан списък

С цел създаване на удобство се използва и целочислена променлива за брояч на компонентите в списъка. Обработката на двусвързания списък включва добавяне и премахване на елемент в началото, в края, както и на произволна позиция в списъка, сравняване на записите в списъка, изпразване на целия списък, проверка размерността на списъка.

2. Описание на приложените алгоритми

Използваните алгоритми в задачата са за добавяне на елемент – **push_front, push_back, insert** и за премахване на елемент – **pop_front, pop_back, erase.** Методите за добавяне и премахване не елемент в началото/края на списъка работят на следния принцип:

- Проверка дали списъка е празен. Ако е празен,при добавяне на елемент, го прави първи и последен, за да е свързан списъка.
- Ако списъкът не е празен, в зависимост от посоката, в която се изтрива или добавя елемент, се прави връзка с предходния/следващия елемент, като връзката е двустранна – свързват се указателите **pPrevious и pNext.**
- Увеличава се и размера на списъка.

Insert и erase за добавяне/премахване не елемент на определена позиция, работят чрез обхождане с **iterator**. Ако се добавя в началото или края на списъка, алгоритъмът е като този при **push_back, push_front** и т.н. Особената част е при добавянето/премахването на елемент от произволна позиция **iterator**. Тогава посредством нов временен **Node** се вмъква конкретната стойност и се свързва със **Node**-овете от двете му страни.

3. Описание на програмния код

Проектът се състои от 3 класа – **Node**, **List** – списък от **Node*, Iterator** – имплементиран вътре в класа **List**. **Iterator** служи за обхождане на списъка. Класът **Node** представлява отделните елементи на списъка. Той има три член-данни: стойност на елемента, указател към предишния и указател към следващия **Node**. В този клас са имплементирани прости функции за достъп и промяна на стойностите на свързаните с текущия **Node** елементи **(getVal, getPrevious, getNext, SetPrevious, SetNext):**

```
template <typename T>
class Node {
public:
    Node(const T&, Node<T>*, Node<T>*);
    T getVal();
    Node<T>* getPrevious();
    Node<T>* getNext();
    void SetPrevious(Node<T>*);
    void SetNext(Node<T>*);
private:
    T value;
    Node<T>* pPrevious;
    Node<T>* pNext;
};
```

Шаблонният клас **List** представлява двусвързания списък от обекти от тип **Node**. Той има за член данни указател към първия и последния елемент на списъка и размер на списъка:

```
private:
    Node<T>* pFront;
    Node<T>* pBack;
    int size;
```

Класът съдържа конструктор по подразбиране-създаващ празен списък, copy конструктор-за създаване на списък от друг, вече съществуващ и деструктор. Функциите за добавяне и премахване на елемент са обяснени по-подробно в 3. Класът включва и функции **front** – достъп до елемента в началото на списъка, **end** – достъп до елемента в края на списъка, **get_size** – достъп до размера на списъка, **clear** – изтриване елементите на списъка и **empty** – проверява дали списъкът е празен. Вътре в класа **Node** е включен и клас **iterator**, имащ като член данна указател към текущ **Node**. Посредством прости методи **getCurrent, setCurrent** и **предефинирани оператори**, класът реализира обхождане на листа.

4. Използвани технологии

Проектът е реализиран чрез Microsoft Visual Studio 2013. Езикът за програмиране, който е използван, е C++.