

# 双链列表的模型与实现

李沁霞

统计学 3210300363

2022 年 10 月 5 日

## 1 项目要求

实现 `DoubleLinkedList<DT>` 的基本操作，使用 `find`, `erase`, `delete` 等函数实现双链表的基本功能，测试代码完备性，并检查代码的内存泄漏。

## 2 设计思路

1. 首先，编写 `find` 外部函数，用于找到该除的函数位置。
2. 然后，构造 `PrintList` void 函数，用于输出迭代器。
3. 最后，编写 `main` 函数进行程序测试。

## 3 实现代码解释

1. `find` 外部函数

```
DoubleLinkedList<int>::iterator find(DoubleLinkedList<int>& list, const int val)
{
    bool flag = false;
    typename DoubleLinkedList<int>::iterator b = list.begin();
    typename DoubleLinkedList<int>::iterator e = list.end();

    for (; b != e; ++b)
    {
        if (*b == val)
        {
            flag = true;
            break;
        }
    }
    if (!flag)
        b = e;
    return b;
}
```

## 2. PrintList void 函数

```
void PrintList(const DoubleLinkedList<int> &l)
{
    DoubleLinkedList<int>::const_iterator itr;
    for(itr = l.begin(); itr != l.end(); ++itr)
    {
        cout << *itr << " ";
    }
    cout << endl;
}
```

## 3. main 函数

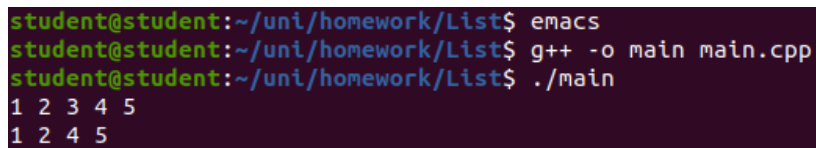
```
int main()
{
    DoubleLinkedList<int> list;
    DoubleLinkedList<int>::iterator iter;
    for(int i = 1; i <= 5; ++i)
    {
        list.push_back(i);
    }

    PrintList(list);
    iter = find(list,3);
    list.erase(iter);
    PrintList(list);
    iter = find(list,3);

    return 0;
}
```

## 4 测试结果及内存泄漏检查

测试结果：



```
student@student:~/uni/homework/List$ emacs
student@student:~/uni/homework/List$ g++ -o main main.cpp
student@student:~/uni/homework/List$ ./main
1 2 3 4 5
1 2 4 5
```

使用 `g++ -o main main.cpp` 测试代码的完备性，然后实现测试结果。显然，输入函数与输出不同。因为，在 `main` 代码中有实现 `erase` 列表的第 3 个位置。于是，输出函数剩下 4 个。

内存泄漏检查：

```
student@student:~/uni/homework/List$ valgrind ./main
==8974== Memcheck, a memory error detector
==8974== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==8974== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==8974== Command: ./main
==8974==
1 2 3 4 5
1 2 4 5
==8974==
==8974== HEAP SUMMARY:
==8974==   in use at exit: 0 bytes in 0 blocks
==8974== total heap usage: 9 allocs, 9 frees, 73,896 bytes allocated
==8974==
==8974== All heap blocks were freed -- no leaks are possible
==8974==
==8974== For lists of detected and suppressed errors, rerun with: -s
==8974== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

使用 valgrind 函数检查内存泄漏。显然，以上代码不存在内存泄漏。