

1 测试程序的设计思路

我使用统一初始化方式创造了一个链表 `List<int> lst_a{1, 2, 3, 4, 5}` 以确保 `List` 类的带参数构造函数被正确执行。

接着我使用 `for` 循环使用 `List` 的静态迭代器对其进行遍历和输出, 确保了内部类 `const_iterator` 和成员函数 `begin()`, `end()` 以及重载后的运算符 `!=`, `==`, `++`, `*` 的正确。

`lst_b` 使用无参数初始化并分别在前后按顺序 `push` 了 1 到 10, `push_back()` 和 `push_front()` 以及 `List` 的默认构造函数被确保正确执行。

通过 `List<int> lst_c = lst_b` 确保拷贝构造函数的正确, 然后测试了 `size()` 函数能正确返回链表的大小。再分别对 `lst_a`, `lst_b` 测试 `pop_back()`, `pop_front()` 函数。

最后通过一个迭代器 `iter` 遍历 `lst_a` 改变每个元素的值为 10086 确保动态迭代器能够改变链表的值。同时通过对处于初始位置时的迭代器分别进行前后自增自减验证了对 `--` 重载的正确。

2 测试的结果

测试结果一切正常。

我用 `valgrind` 进行测试, 发现没有发生内存泄露。