

LinkedList 单链表模块

本模块实现了一个通用的单链表模板类 `LinkedList<T>`，支持插入、删除、查找、遍历等常用操作，接口风格参考工业界标准，适用于 C++ 项目。

特性

- 支持任意类型元素（模板实现）
- 动态内存分配，无需预设容量
- 提供常用链表操作接口
- 边界检查与异常安全
- 代码风格规范，接口注释详细
- 附带交互式测试程序

主要接口

- `LinkedList()`：构造函数，初始化空链表
- `~LinkedList()`：析构函数，释放所有节点
- `void clear()`：清空链表内容
- `bool empty() const`：判断链表是否为空
- `int size() const`：获取链表长度
- `T get(int index) const`：获取指定位置的元素
- `int find(const T& value) const`：查找元素首次出现的位置
- `void insert(int index, const T& value)`：在指定位置插入元素
- `void remove(int index)`：删除指定位置的元素
- `void traverse(void (*visit)(const T&)) const`：遍历链表

详细接口说明见 [./include/linkedlist.hpp](#)。

用法示例

```
#include "linkedlist.hpp"
#include <iostream>

void printInt(const int& x) { std::cout << x << " "; }

int main() {
    LinkedList<int> list;
    list.insert(0, 10);
    list.insert(1, 20);
    list.insert(1, 15);
    std::cout << "链表内容: ";
    list.traverse(printInt);
    std::cout << std::endl;
    std::cout << "第1个元素: " << list.get(1) << std::endl;
    list.remove(0);
    std::cout << "size = " << list.size() << std::endl;
    std::cout << "index of 20 = " << list.find(20) << std::endl;
    return 0;
}
```

交互式测试

本模块提供了交互式测试程序，支持命令行操作链表，便于学习和调试。**测试程序所有交互均为中文，适合中文用户体验。**

编译并运行：

```
g++ -std=c++11 test/test_linkedList.cpp -o test_linkedList
./test_linkedList
```

示例交互（中文版）：

```
===== 单链表交互测试菜单 =====
命令列表：
  insert <下标> <值>      : 在下标插入值
  remove <下标>           : 删除指定下标的元素
  get <下标>              : 获取指定下标的值
  find <值>               : 查找值，返回下标
  size                   : 当前元素个数
  empty                  : 判断链表是否为空
  print                  : 打印链表内容
  help                  : 显示菜单
  exit / 0              : 退出程序
-----

请输入命令：> insert 0 42
已在下标 0 插入 42。
> print
链表内容：42
> insert 1 99
已在下标 1 插入 99。
> get 1
下标 1 的值为：99
> exit
程序结束，再见！
```

常见问题

- **Q: 插入/删除越界怎么办?**
A: 会抛出异常并提示错误信息。
- **Q: 支持哪些类型?**
A: 支持任意可赋值类型（模板实现）。

模板使用说明

本模块为模板实现，建议直接包含 `linkedList.hpp` 头文件，无需单独编译 `cpp` 文件。例如：

```
g++ -std=c++11 your_code.cpp -o your_program
```

相关文档

- [doc/ADT.md](#)：单链表抽象数据类型说明
- [./include/linkedList.hpp](#)：接口定义与注释

