

```

#pragma once
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
//  单向循环链表的类定义
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
#include "LinkList.h"

template<class type> class CirList
{
private:
    node<type> *head;// 头结点指针

public:
    CirList(void);// 无参数的构造函数模板
    ~CirList(void);// 析构造函数模板
    CirList(const CirList<type> &copy);    // 复制构造函数模板
    CirList(type a[], int n);    //为调试程序，增加通过数组构造链表的构造函数

    void Clear();
    bool Empty() const {return head->next == head};    // 判断线性表是否为
空
    int Length() const;    // 求线性表长度
    node<type>* Locate(type &x);    //定位
    void InsertBefor(node<type> *pcurrent, const type &x); //当前结点之前插入
    void InsertAfter(node<type> *pcurrent, const type &x); //当前结点之后插入
    void PrintList();    //输出链表所有内容
    void Append(const type &x);    //追加元素到链尾
    void InsertAsFirst(const type &x);//插入元素作为表中的第一个元素
    ////////////////////////////////////////////////////////////////////补充
    void InsertAfter (const type & x, const type &y); //在 x 前插入 y
    void InsertAfter (const type & x, const int i);    // 在第 i 个元素后插入 x
    type Remove(const int i);    //删除第 i 个结点
    node<type>* GetElement(int i); //返回第 i 个元素的地址
};

template<class type> void Josephus(int n, int m);

```