

# Microsues的专栏 Stay foolish,stay hungry.

```
늘 摘要视图
                                                                                                              :■ 目录视图
                                                                                                                                              RSS 订阅
个人资料
                                       链表——用C++模板实现
                                   标签: c++ null insert string list 测试
                                                                                                     2010-12-29 20:02
                                                                                                                    8145人阅读
                                                                                                                                  评论(2) 收藏 举报
                                  ≣ 分类: 数据结构(11) ▼
                                  ■ 版权声明:本文为博主原创文章,未经博主允许不得转载。
          Microsues
                                  LinkedList.h
访问: 80522次
                                           [cpp] view plain copy print ?
积分: 1205
                                       01. //单向链表的实现
 等级: BLDG 4
                                       02. //查找、插入和移除某个元素、复制、清空单向链表的时间复杂度均为O(n)
排名: 千里之外
                                           //设计类时,应尽量减少类的成员函数之间的依赖关系
                                       03.
                                       04.
                                            #ifndef LINKEDLIST_H
 原创: 35篇 转载: 19篇
                                       05.
                                            #define LINKEDLIST H
译文: 0篇
             评论: 28条
                                            template<class DT>
                                       06.
                                       07.
                                            struct Node
                                       08.
文章搜索
                                       09.
                                             DT info;
                                       10.
                                             Node<DT> *next;
                                       11.
                                            template<class DT>
                                       12.
                                       13.
                                            class LinkedList
                                       14.
文章分类
                                       15.
                                           public:
                                       16
                                             LinkedList():
C/C++ (22)
                                       17.
                                             LinkedList(const LinkedList<DT> &aplist);
CSAPP读书笔记 (7)
                                       18.
                                             ~LinkedList():
MFC (2)
                                       19.
                                             LinkedList<DT>& operator=(const LinkedList<DT> &rlist);
winsock编程 (5)
                                       20.
                                              void insert(const DT &element); //在链表的头部之前插入
                                       21.
                                              void insert_end(const DT &element); //在链表的尾部插入
数据结构 (12)
                                             bool first(DT &listEl); //得到对头的数据
                                       22.
杂记 (2)
                                              inline bool getNext(DT &listEl); //得到当前指针所指的下一个数据
                                       23.
算法 (2)
                                       24.
                                              bool find(const DT &element); //查找一个数据
C#高级编程读书笔记 (2)
                                       25.
                                              bool retrieve(DT &element);
                                                                   //检索一个数据
C++primer复习笔记 (1)
                                       26.
                                              bool replace(const DT &newElement); //更改一个数据
                                       27.
                                              bool remove(DT &element);
                                       28.
                                              bool isEmpty() const;
                                       29.
                                             void makeEmpty();
文章存档
                                       30.
                                            private:
                                             Node<DT> *start; //指向头结点
                                       31.
2011年07月 (9)
                                       32.
                                              Node<DT> *current; //指向当前的结点
2011年06月 (9)
                                       33.
                                             inline void deepCopy(const LinkedList<DT> &original); //"深复制"
2011年05月 (2)
                                       34.
                                       35.
2011年03月 (1)
                                           #endif
2011年01月 (8)
                        展开
阅读排行
                                  LinkedList.cpp
链表——用C++模板实现
                       (8138)
                                           [cpp] view plain copy print ?
2011年.NET面试题总结---南
                       (4746)
                                            #include "LinkedList.h"
使用jrtplib写的一个简单服
                       (3924)
                                            template<class DT>
                                       02.
                                           LinkedList<DT>::LinkedList()
struct/class/union内存对齐原
```

(3617)

在VC9.0中实现C++模板类

```
阿里巴巴笔试题-求数组印
                  (3401)
优先级队列——用C++模板
                   (2820)
在vs2008 vc++ 中添加mfc中
                   (2552)
栈——用C++模板实现
                   (2358)
如何理解数组作为函数参
                  (2180)
```

```
评论排行
使用jrtplib写的一个简单服
                     (4)
在VC9.0中实现C++模板类
                     (3)
栈——用C++模板实现
                     (3)
C/C++面试颗 (二)
                     (3)
链表——用C++模板实现
                     (2)
浮点数位模式和实体模式
                     (2)
关于C/C++中的关键字dele
                     (2)
使用TCP+winsock写的一个
                     (2)
chap 13: 重载操作符与转
                     (1)
使用winsock+UDP写的一个
                     (1)
```

## 推荐文章

- \* 致JavaScript也将征服的物联网世
- \* 从苏宁电器到卡巴斯基:难忘 的三年硕士时光
- \*作为一名基层管理者如何利用 情商管理自己和团队(一)
- \* Android CircleImageView圆形
- \* 高质量代码的命名法则

# 最新评论

链表——用C++模板实现 jizhji: 为啥没有遍历呢

# -用C++模板实现

折夜: Node \*copyPtr = top = new Node; 这样就可以把栈清空?楼主可否 解释一丁

#### 使用jrtplib写的一个简单服务器和智 烟雨\_楼台: 太不负责了

# 链表——用C++模板实现

hncdratio: 代码无法运行,链接错

# 栈——用C++模板实现

zkpingguo: operator=那个赋值函数, 应该return \*this:

# 关于C/C++中的关键字delete

SomeAody: 顶赞

C/C++面试题(二) xinghu568: 题目一作者写的是大数 下沉,怎么会输出前n个小数的呢

## 浮点数位模式和实体模式的转换 Dophi: 学习了

关于C/C++中的关键字delete jyb\_haha: 恩恩,懂了!!!

使用TCP+winsock写的一个简单客员 bxyill: 如何发送文件呢?怎么实现?谢谢

```
05.
        start = current = NULL;
06.
07.
      template<class DT>
      LinkedList<DT>::LinkedList(const LinkedList<DT> &aplist)
08.
09.
10.
        deepCopy(aplist);
11.
12.
      template<class DT>
13.
      LinkedList<DT>::~LinkedList()
14.
15.
        makeEmpty();
16.
17.
      template<class DT>
18.
      LinkedList<DT>& LinkedList<DT>::operator =(const LinkedList<DT> &rlist)
19.
20.
        if(this == &rlist)
21.
           return this:
22
        makeEmpty();
        deepCopy(rlist);
23
24.
        return this:
25.
26.
      //在链表的表头插入DT &element,插入后没有当前位置
27
      template<class DT>
28.
      void LinkedList<DT>::insert(const DT &element)
29
30.
        current = NULL:
31.
        Node<DT> *NNode = new Node<DT>;
32.
        NNode->info = element;
33.
        NNode->next = start:
34.
        start = NNode;
35.
36.
      ////在链表的尾部插入
37.
      template<class DT>
38.
      void LinkedList<DT>::insert_end(const DT &element)
39.
40.
        current = NULL;
41.
        Node<DT> *NNode = new Node<DT>;
42
        NNode->info = element:
43.
        NNode->next = NULL;
        Node<DT> *temp:
44
45.
        DT item;
46
        if(start == NULL)
47.
48
           start = NNode
49.
          temp = start;
50
          return:
51.
52
        temp = start:
53.
        while(temp->next != NULL)
54.
55.
          temp = temp->next;
56.
57.
        temp->next = NNode;
58.
59.
      //把链表中的第一个元素存放在listEI中,并且把第一个元素的位置置为current
      template<class DT>
60.
61
      bool LinkedList<DT>::first(DT &listEI)
62.
        if(start == NULL)
63
64.
           return false
        current = start;
65.
66.
        listEI = start->info;
67
        return true:
68.
      //把链表中current的下个元素放在listEl中,current指向下一个元素
69
70.
      template<class DT>
      bool LinkedList<DT>::getNext(DT &listEI)
71.
72.
73.
        if(current == NULL)
74.
          return false;
75.
        if(current->next == NULL)
76.
77.
           current = NULL;
78.
           return false:
79.
80.
        listEI = current->next->info;
81.
        current = current->next;
82.
        return true:
83.
```

```
84.
       //如果找到elment,返回true,current在getNext()中设置
       template<class DT>
85.
86.
       bool LinkedList<DT>::find(const DT &element)
87.
88.
         if(!first(item))
89.
            return false; //检查是否为空
90.
91.
92.
            if(item == element)
93.
               return true;
         }while(getNext(item));
94.
95.
          return false;
96.
97.
       template<class DT>
98.
       bool LinkedList<DT>::retrieve(DT &element)
99.
100.
         if(!find(element))
101.
            return false;
102
          element = current->info;
103.
         return true;
104
105.
       template<class DT>
106.
       bool LinkedList<DT>::replace(const DT &newElement)
107.
         if(current == NULL)
108.
109.
            return false;
110.
         current->info = newElement;
111.
         return true;
112.
113.
       template<class DT>
       bool LinkedList<DT>::remove(DT &element)
114
115.
116.
         current = NULL;
         if(start == NULL)
117.
118.
            return false;
119.
         Node<DT> *ptr = start;
120.
          if(start->info == element)
121
122
            element = start->info;
123
            start = ptr->next:
124.
            delete ptr;
125
            return true;
126
          while(ptr->next != NULL){
127
128.
            if(ptr->next->info == element)
129
130.
               Node<DT> *tempptr = ptr->next;
131
               element = tempptr->info;
132.
               ptr->next = tempptr->next;
133.
               delete tempptr;
134.
               return true;
135.
136.
            ptr = ptr->next;
137.
138.
         return false;
139.
140.
       template<class DT>
141.
       bool LinkedList<DT>::isEmpty() const
142
143.
          return start == NULL;
144.
145.
       template<class DT>
       void LinkedList<DT>::makeEmpty()
146.
147.
148
         while(start != NULL)
149.
150.
            current = start;
151.
            start = start->next;
152.
            delete current;
153.
154.
         current = NULL;
155.
156.
       template<class DT>
157.
       void LinkedList<DT>::deepCopy(const LinkedList<DT> &original)
158.
159.
         start = current = NULL;
160.
          if(original.start == NULL)
161.
            return:
         Node<DT> *copyptr = start = new Node<DT>;
```

# 链表——用C++模板实现 - Microsues的专栏 - 博客频道 - CSDN.NET

```
163.
          Node<DT> *originalptr = original.start;
164.
          copyptr->info = originalptr->info;
165.
          if(originalptr == original.current)
166.
             current = copyptr;
167.
          while(originalptr->next != NULL)
168.
169.
            copyptr->next = new Node<DT>; //这一句有难度啊!
170.
             originalptr = originalptr->next;
171.
            copyptr = copyptr->next;
172.
             copyptr->info = originalptr->info;
             if(originalptr == original.current)
173.
174.
               current = copyptr;
175.
176.
          copyptr->next = NULL;
177.
178. //#endif
```

## 测试用例:

```
[cpp] view plain copy print ?
01.
     #include<iostream>
02.
      #include "LinkedList.cpp"
03.
      #include<string>
04.
      using namespace std;
05.
      void main()
06.
     {
07.
        string str[5] = {"hello", "world", "welcome", "to", "beijing"};
08.
        //初始化一个链表
        LinkedList<string> strLL;
09.
10.
        for(int ix=0;ix<5;++ix)
                                  //调用insert(),依次插入在链表的头部
11.
     // strLL.insert(str[ix]);
12.
           strLL.insert_end(str[ix]);
13.
        //遍历链表
14.
        cout<<"遍历链表: ";
15.
        string listEl;
16.
        if(strLL.first(listEI))
           cout<<li>istEl<<""; //获得current
17.
18.
        while(strLL.getNext(listEI))
19.
           cout<<li>tEl<<" ";
20.
        cout<<endl;
        LinkedList<string> strLL2(strLL); //调用构造函数初始化
21.
22.
        LinkedList<string> strLL3 = strLL; //调用重载"="操作符
23.
        strLL2.first(listEI);
24.
        cout<<"ListEl = "<<li>istEl<<endl;
25.
        strLL3.first(listEI);
        cout<<"ListEl = "<<li>istEl<<endl;
26
27.
         if(strLL2.find(str[3]))
                                    //调用find()
           cout<<"str[3] = "<<str[3]<<endl;
28.
29.
        if(strLL3.retrieve(str[2]))
                                    //调用retrieve()
           cout<<"str[2] = "<<str[2]<<endl;
30.
31.
         if(strLL.remove(str[1]))
32
33.
           strLL.first(listEI);
           cout<<"ListEl after remove is "<<li>istEl<<endl;
34.
35.
36.
        if(strLL.isEmpty())
37.
           cout<<"List is Empty!"<<endl;
38.
        strLL.makeEmpty();
39.
        strLL2.makeEmpty();
40.
        strLL3.makeEmpty();
41.
        cout<<"hello"<<endl;
42.
```

**顶** 踩