

```

1 // linklist.h : definition for the linklist ADT
2
3 #ifndef LINKLIST_H_INCLUDED
4 #define LINKLIST_H_INCLUDED
5
6 typedef struct node{
7     int aid;
8     wchar_t name[MAXN];
9     wchar_t auth[MAXN];
10    wchar_t pub[MAXN];
11    int time;
12    double price;
13    struct node * nxt;//下一个指向的节点
14 } LinkNode, * LinkList;
15
16 void swapint(int * m, int * n)//整数的交换
17 {
18     int t;
19     t = *m;
20     *m = *n;
21     *n = t;
22 }
23
24 void swapdble(double * m, double * n)//浮点数的交换
25 {
26     double t;
27     t = *m;
28     *m = *n;
29     *n = t;
30 }
31
32 void swapwcs(wchar_t * m, wchar_t * n)//字符串的交换
33 {
34     wchar_t t[MAXN];
35     int l = sizeof(wchar_t) * MAXN;
36     memcpy(t, m, l);
37     memcpy(m, n, l);
38     memcpy(n, t, l);
39 }
40
41 void swapnode(LinkList s, LinkList p)//交换链表节点
42 {
43     swapint(&((*s).aid), &((*p).aid));
44     swapwcs(s->name, p->name);
45     swapwcs(s->auth, p->auth);
46     swapwcs(s->pub, p->pub);
47     swapint(&((*s).time), &((*p).time));
48     swapdble(&((*s).price), &((*p).price));
49 }
50
51 void prt_node(LinkList p)//打印节点
52 {
53     printf("%ls%d\n", L"图书编号: ", p -> aid);
54     printf("%ls%ls\n", L"书名: ", p -> name);
55     printf("%ls%ls\n", L"作者: ", p -> auth);
56     printf("%ls%ls\n", L"出版社: ", p -> pub);
57     printf("%ls%d\n", L"出版时间: ", p -> time);
58     printf("%ls%.2f\n", L"价格: ", p -> price);
59     printf("\n");
60 }
61
62 void InsertList(LinkList t, LinkList e)//将链表尾部插入一个节点
63 {
64     LinkList p, q;
65     q = t;
66     while(q -> nxt != NULL){//将q指针一直指到链表最后一个节点
67         q = q -> nxt;
68     }
69     p = (LinkList)malloc(sizeof(LinkNode));
70     memcpy(p, e, sizeof(LinkNode));//将新添加的内容拷贝到新节点中
71     q -> nxt = p;//添加节点到链表尾部
72 }
73

```

```

74 void DispList(LinkList t)//遍历打印链表节点
75 {
76     LinkList q;
77     q = t->nxt;
78     while(q != NULL){
79         prt_node(q);//打印节点
80         q = q -> nxt;
81     }
82 }
83
84 void SortList(LinkList t)//将链表内容排序
85 {
86     LinkList p, q, s;
87     if(t->nxt == NULL) return;//如果链表为空直接退出
88     for(p = t->nxt; p->nxt != NULL; p = p->nxt){//实现简单的选择排序
89         s = p;
90         for(q = p->nxt; q != NULL; q = q->nxt){
91             if(s->aid < q->aid) s = q;
92         }
93         if(s != p){
94             swapnode(s, p);//交换两节点
95         }
96     }
97 }
98
99 void DeleteList(LinkList t, int id)//删除链表节点
100 {
101     LinkList pre, p;
102     pre = t;
103     p = pre->nxt;
104     while(p != NULL){//遍历链表
105         if(p->aid == id){//符合条件
106             pre->nxt = p->nxt;//将前一个节点指向本节点的下一个节点
107             free(p);//删除本节点
108             break;
109         }
110         pre = p;
111         p = p->nxt;
112     }
113 }
114
115 void DestroyList(LinkList t)//销毁整个链表
116 {
117     LinkList pre, p;
118     pre = t->nxt;
119     if(pre == NULL) return;
120     t->nxt = NULL;//头部节点nxt值设置为NULL
121     p = pre->nxt;
122     while(p != NULL){
123         free(pre);//删除前一个节点
124         pre = p;
125         p = p->nxt;//移动到下一个节点
126     }
127     free(pre);//删除最后一个节点
128 }
129
130 void FindListId(LinkList t, int id)//根据图书编号查找
131 {
132     LinkList p;
133     p = t->nxt;
134     while(p != NULL){
135         if(p->aid == id)//符合条件
136             prt_node(p);
137         p = p->nxt;
138     }
139 }
140
141 void FindListAuth(LinkList t, const wchar_t * au)//根据作者查找
142 {
143     LinkList p;
144     p = t->nxt;
145     while(p != NULL){
146         if(!wcscmp(p->auth, au))//符合条件

```

```

147         prt_node(p);
148         p = p->nxt;
149     }
150 }
151
152 void ModifyList(LinkList t, LinkList e) //修改节点内容
153 {
154     LinkList p;
155     p = t->nxt;
156     while(p != NULL){
157         if(p->aid == e->aid){ //符合条件
158             swapnode(p, e); //交换节点内容
159             break;
160         }
161         p = p->nxt;
162     }
163 }
164
165 void StatList(LinkList t, const wchar_t * pu) //按照出版社统计图书
166 {
167     int cnt = 0;
168     LinkList p;
169     p = t->nxt;
170     while(p != NULL){
171         if(!wcscmp(p->pub, pu)){
172             prt_node(p);
173             cnt++;
174         }
175         p = p->nxt;
176     }
177     printf("%ls%d%ls\n", L"共统计到", cnt, L"本书。");
178 }
179
180
181 #endif // LINKLIST_H_INCLUDED
182

```