

Управление на динамичната памет: упражнение (Rev: 1.1)

Любомир Чорбаджиев

17 април 2007 г.

Във файла `10-lab.zip` е даден е кодът на класа `class list` и примерна `main()` функция, която го използва. Класът `class list` е реализация на едносвързан списък.

Задачи

Задача 1: Да се разгледа файла `list.cc`, дефиницията на `class list` и `main()` функцията. Да се компилира програмата и да се изпълни.

Задача 2: Към `class list` да се добави метод `void append(int val)`, който добавя елемент в края на списъка.

Задача 3: Да се напише деструктор на `class list`.

Задача 4: Да се напише копиращ конструктор на `class list`.

Задача 5: Да се напише оператор за присвояване на `class list`.

Задача 6: Да се напише `class dlist`, който да реализира двойно свързан списък. Класът `class dlist` да притежава деструктор, копиращ конструктор и оператор за присвояване.

□

Приложение: файлът list.cc

```
1 #include <iostream>
2 #include <exception>
3 using namespace std;
4
5 class list {
6     struct elem{
7         int data_;
8         elem* next_;
9         elem(int val=0) : data_(val), next_(NULL) {}
10    };
11
12    elem *head_;
13
14    public:
15        list(): head_(NULL){}
16
17        bool empty() {
18            return head_==NULL;
19        }
20
21        void prepend(int val) {
22            elem* new_elem=new elem(val);
23            new_elem->next_=head_;
24            head_=new_elem;
25        }
26
27        void del() {
28            if(!empty()) {
29                elem* to_del=head_;
30                head_=head_->next_;
31                delete to_del;
32            }
33        }
34
```

```

35 void print() {
36     elem* curr_elem=head_;
37     while(curr_elem!=NULL) {
38         cout << curr_elem->data_ << ", ";
39         curr_elem=curr_elem->next_;
40     }
41     cout << endl;
42 }
43
44 int first() {
45     if(empty()) {
46         cerr << "the list is empty..." << endl;
47         throw exception();
48     }
49     return head_->data_;
50 }
51
52 void release() {
53     while(!empty())
54         del();
55 }
56 };
57
58 int main(void) {
59     list l1;
60     l1.prepend(10);
61     l1.prepend(20);
62     l1.prepend(30);
63
64     l1.print();
65
66     cout<< "first is: "
67         << l1.first() << endl;
68     l1.del();
69     cout<< "after del() first is: "
70         << l1.first() << endl;
71
72     return 0;
73 }

```