

# Персистентни СД. Умни указатели

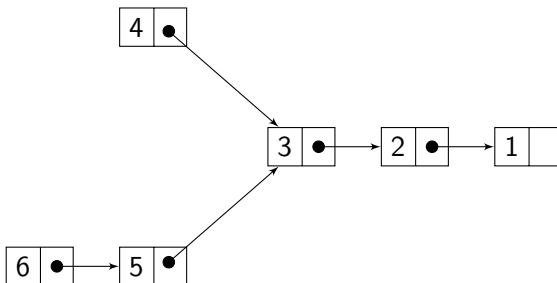
Калин Георгиев

26 октомври 2020 г.

# Персистентни СД

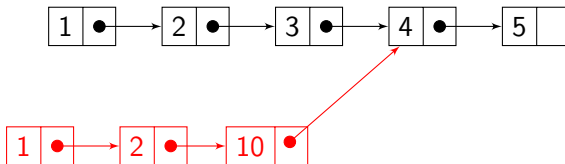
# Персистентни СД

```
let list = [3,2,1]  
list1 = 4:list  
list2 = 6:5:list
```



# Модификация на данните

```
list<int> list ({1,2,3,4,5});  
//list.set(2,10); -невъзможно  
list<int> result = list.set(2,10);
```



Потребител / собственик на памет

# Създател, потребител, собственик

```

struct node
{int data; node* next};

node* cons (int head, node* tail)
{return new node{head,tail};}

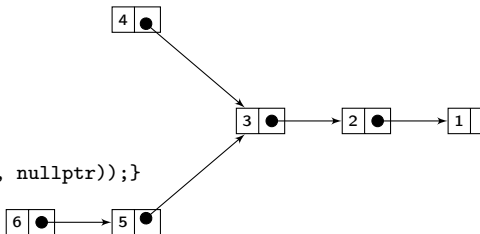
node* makelist ()
{return cons (3, cons (2, cons (1, nullptr)));}

void uselists ()
{
    node* list = makelist();
    node* list1 = cons (4, list);
    node* list2 = cons (6, cons (5, list));

    anotheruser1(list1);
    anotheruser2(list2);

    //delete list, list1, list2?
}

```

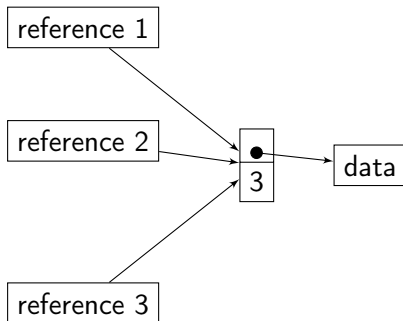


## Умни указатели

# Reference counting. `std::shared_ptr<T>`

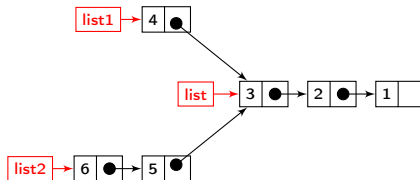
`node* → std::shared_ptr<node<T>>`

`new → std::make_shared`





# Reference counting



```

struct node
{int data; shared_ptr<node> next};

shared_ptr<node> cons (int head, shared_ptr<node> tail)
{return make_shared(head,tail);}

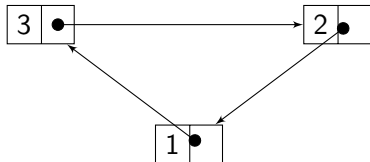
shared_ptr<node> makelist ()
{return cons (3, cons (2, cons (1, nullptr)));}

void uselists ()
{
    shared_ptr<node> list = makelist();
    shared_ptr<node> list1 = cons (4, list);
    shared_ptr<node> list2 = cons (6, cons (5, list));

    anotheruser1(list1);
    anotheruser2(list2);
} // ~list, ~list1, ~list2

```

# Някои проблеми



Благодаря за вниманието!